TU Delft

Simulation based research on human-robot collaboration

Lukas Bannink

Chair: Dr. J. Boyle PhD
Mentor: Dr.ir. O. Siebinga
Msc. Integrated Product Design Faculty of Industrial Design Engineering

February 2026

01

Introduction to the problem

1.2 Table of contents

1.3 Introduction

The hospitality sector is experiencing persistent staff shortages, increasing interest in robotic systems as supplementary service agents. While service robots are technically capable of performing delivery and navigation tasks, their integration into dynamic human environments raises questions regarding control, responsibility, and interaction quality. Determining how autonomy and human input should be balanced remains a central challenge in service robotics.

This thesis initially set out to evaluate which robot control strategies and visualisation methods were most effective within a restaurant context. The aim was to compare predefined approaches and identify an optimal configuration based on measurable performance and user experience metrics. However, as development progressed, increasing complexity emerged within the simulation itself. Significant effort was required to construct a robust, flexible environment capable of supporting meaningful experimentation. The simulation evolved from a testing tool into a substantial design outcome in its own right.

As a result, the scope of the project shifted. Rather than focusing narrowly on identifying a single optimal control or visualisation method, the research broadened to address the design and structure of the simulation framework itself. The final system enables a spectrum of control strategies, ranging from full autonomy to full tele operation, and provides a structured foundation for evaluating these approaches. Over the ten month duration of the project, iterative refinement, pilot testing, and technical constraints shaped this transition from a focused comparison study to a more generalised and extensible experimental platform.

The outcome of this thesis is therefore twofold: an operational simulation framework for human robot interaction in service contexts, and a structured method for investigating shared control within dynamic environments.

1.4 List of abbreviations and other words

Most of of the following terms are explained in the text from the first moment they are mentioned. This list serves as a reminder for those who missed the initial explanation.

Control strategies - A combination of steering and perspective tools, such as a first person perspective paired with arrow key controls.

Visualisation methods - A set of tools designed to alert human operators of a conflict of movement.

Conflict of movement - A confrontation in which a human and a robot are in the way of one another.

NavMesh - A pre-calculated area in the simulated environment that determines which areas are accessible, and which areas are blocked by obstacles.

Robot agent - A simulated robot agent that is capable of navigating their own way around the NavMesh, while also auto-delivering orders to tables.

Human agent - A simulated robot agent that is capable of navigating their own way around the NavMesh.

Human operator - The person overviewing and operating the robots from the view of the digital twin.

Highway behaviour - A phenomenon that occurs when multiple agents calculate the same route along the same axis.

Total stops - The total count of instances where a human's path was blocked to the point of requiring them to stop.

Total stopping score - An accumulated value that scales based on the time and proximity of robots in the human's path.

Path desynchronization - A phenomenon that occurs when two identical routes deviate from their original path by unexpected interference.

Framerate (FPS) - The amount of frames per second the simulation calculates. Code is executes every time a new frame is calculated.

1.5 Trend analysis

In order to get a clear image of the future, we can try to analyse current trends that might influence the integration of robotics in socially demanding environments, such as healthcare or restaurants. This allows us to spot certain threats, but also opportunities in advance.

Medical professionals have a negative attitude towards robots with more autonomy

Main takeaway

The 111 respondents were overall negative using AI and robot technology related to caring activities. However, all groups were positive in using robots in service tasks, monitoring/alarms, telemedicine communication. Of 29 assertions, 18 were mostly positive and 13 of them were over 70 % positive. (Göransson et al., 2018)

How will this affect the thesis?

Just like restaurant staff, there seems to be a natural distrust for robots taking over certain human-centred jobs. Self preservation is usually a higher priority than a desire to improve the field in general.

Perceived empathy from robots seems to become the highest factor in device use acceptance

Main takeaway

Out of 400 hospital patients, the most based their willingness to empathy, which in turn positively affects interaction quality. it should be noted however, that the study was done in China, which can potentially mean a different cultural bias. Still these results can be quite interesting to keep in mind.(Li et al., 2024)

How will this affect the thesis?

This further enforces the importance of human intervention at key moments. Seeing as the key interaction moments in delivery-related tasks are only a small portion of the total operating time.

Employees feel less threatened by automation services with low social skills

Main takeaway

Lower level service employees tend to feel more threatened by robots with higher social skills, even though that doesn't necessarily take away their value as employees. (Parvez et al., 2022)

How will this affect the thesis?

In the end, the perception of the social skills of robots can be more important than their actual performance value for the acceptance of robots by employees. It is therefore important that the employees feel like they are in control on a social level.

Introducing robotics raises the bar of entry-level jobs in restaurant environments

Main takeaway

Introducing complex technology in a traditionally simple (yet mentally demanding) environment means that a smaller, but more skilled workforce is more efficient. This might cause certain people with more traditional skillsets to lose their value in the job market.

How will this affect the thesis?

It might be very good to look at how we can optimally use the skills that current employees poses, and how we can make it easier for them to transform into a more skilled and efficient workforce. This should be explored further back in the stakeholder analysis.

Distrust in robotics is more common among those who spend more time around robots

Main takeaway

Rising fear among workers that robots outperform humans leads to higher job insecurity, stress, and turnover intentions. (Chen & Cai, 2024)

How will this affect the thesis?

It is important to underline the robot as a potential help, instead of a potential competitor for the same position, although there is no denying it that robots improve the productivity of humans, and thus requiring less humans to do the same job.

Food delivery robots are rising at fast rate with a compound annual growth rate of 19.8% (USA)

Main takeaway

The use of delivery robots as an alternative for your standard delivery vehicle seems to be a rapidly increasing trend.

How will this affect the thesis?

While there is a lot to be said, the need for autonomous systems like these cannot be denied, and should offer plenty of opportunities for inspiration to our use case.

Summarised:

Trust in robots decreases when autonomy increases, especially in roles involving social interaction or emotional labour. People working closely with robots tend to feel more threatened, particularly when robots display high social competence. At the same time, research shows that shared control between human and machine leads to better performance, higher trust, and lower mental workload.

Perceived empathy and clear communication from robots increase acceptance, especially during moments of interaction. In environments where new technology raises the skill level required for entry-level work, existing employees may experience pressure or resistance. While the use of robots continues to grow, acceptance depends largely on how their role is framed and how transparent their behaviour is to those around them.

How will this affect feasibility and desirability?

With this in mind, it is now possible to derive the main advantages in both feasibility and desirability. This means, that in the future, certain design factors such as shared control strategies will have extra benefits in the context of these growing trends.

Design Factor	Feasibility Boost	Desirability Boost
Shared control strategies	Reduces dev complexity by using tested modes	Aligns with trust and control preferences
Modular visualisation tools	Allows custom setups for different users	Avoids info overload, supports comfort
Clear robot-human boundaries	Prevents risky overlaps in task ownership	Makes staff feel safe and respected
Communication via visuals	Easier to implement than speech or emotion	Increases perceived empathy
Flexible training potential	Helps integrate with existing workforce	Reduces fear of skill obsolescence

Conclusion

The research trends confirm the value of the current approach. Each control strategy offers a clear division of responsibility between operator and robot, and the visualisation methods support readable, non-verbal communication in shared spaces. The system is designed to reduce mental workload while keeping the human in control, which increases both practical use and social acceptance. Framing the robot as a tool for support rather than replacement will be essential to long-term integration. By focusing on shared control, transparent behaviour, and adaptable feedback, the system remains both feasible to build and desirable to use.

1.6 Stakeholder analysis

Following the trend analysis, a clearer understanding has been developed of the various stakeholders involved in the implementation of robotic systems within service environments. The next section identifies and explores the key stakeholders relevant to the context of this project. These include restaurant staff, customers, technical operators, management, and system developers. Each stakeholder group is examined based on their role, interests, potential concerns, and influence on the success of the system.

Robot operator

Motto: "Wait, robot 5 hit a child AGAIN?"

Focus: High

Short bio:

The robot operator is in charge of providing optimal service towards the customers. They do this by leaving the simple tasks to the robots as much as possible, while trying to optimally use their social skills.

Wants and needs:

- An easy interface that allows them to assess threats from a distance
- Being able to seamlessly control the robots to solve a situation
- Keep the stream of input and output both logical and digestive

Frustrations:

- Needing to multitask while multiple robots demand their attention
- Having to assess the situation as fast as possible while people are waiting

Public visitors (restaurant customers)

Motto: "I have been waiting on my drink for ages, and now that stupid robot is getting in the way"

Focus: High

Short bio:

The customers of a local restaurant, or any place really, have trouble communicating with robots. It is very difficult to asses their intentions, and this often leads conflicts of intentions.

They often show a strong preference towards working with humans, although studies show that human controlled robots tend to get more sympathy.

Wants and needs:

- Predictability from the robot
- As much human contact as possible.

Frustrations:

- Delayed response from the robot
- No clear communication
- Getting the feeling that they are not worth human attention

Traditionally operating staff (waiters)

Motto: "It's a busy day today, I hope these damned robots can take some load off.

Focus: Medium

Short bio:

The restaurant staff is already overworked at the moment, and doesn't really need the robot to intrude their business more than it needs to. They will probably be sceptical of technology, as their current work has been hard, but what they are already used to.

Wants and needs:

- Relieve work pressure
- Focus on complicated tasks while the robots take over the more simple ones
- Predictability to

Frustrations:

- Having to deal with the robot in unknown positions
- Having something that interrupts their routine
- Robot physically getting in the way
- Feeling a threat on their job security

Instance managers (restaurant manager)

Motto: "I hope these robots will make their tremendous investment costs worth it"

Focus: Medium

Short bio:

Due to a rising shortage in HORECA personnel, managers are switching to robots, as they are perceived to be more flexible.

This also means that they are taking a big risk, as taking robots can be seen as a very cheap way to do your restaurant, just like the self order machines you see at mcdonalds.

Wants and needs:

- Keeping up with the rising costs of serving the customers
- Future proofing their establisment
- Work around an unpredictable labour shortage

Frustrations:

- Abnormalities in their long term output
- A lack of flexibility in worker output
- Negative customer feedback

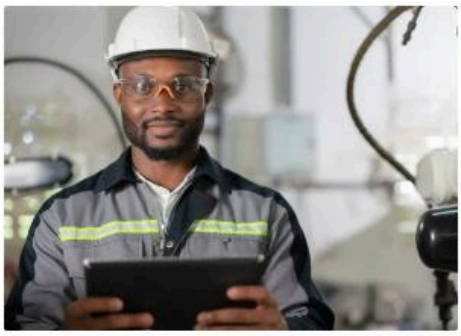
System Developers / Technical Staff

Motto: "This isn't a bug, it's a feature."

Focus: Low to medium

Short bio:

Responsible for implementing, updating, and maintaining the robot's software and hardware. They often operate behind the scenes but play a critical role in ensuring system reliability and compatibility with real-world restaurant dynamics.



Wants and needs:

- Clear feedback on system failures
- Usability insights to improve interfaces
- Modular and testable system requirements

Frustrations:

- Vague bug reports
- Misalignment between design intent and real-world usage
- Unclear performance expectations

Conclusion

The stakeholder analysis provides a comprehensive overview of the key groups affected by and involved in the integration of robotic systems within the service environment. Understanding the distinct needs, frustrations, and interactions of these stakeholders is essential for designing control strategies and visualisation methods that are both effective and accepted.

By prioritizing the robot operators, customers, traditional staff, and management, the project focuses on those with the most direct influence on system success. Including secondary stakeholders such as system developers helps ensure technical feasibility and responsiveness to real-world challenges.

This analysis underlines the importance of balancing operational efficiency with user acceptance and highlights areas where clear communication and shared control can reduce friction. Combining these findings enables the formulation of targeted demands and wishes that will guide the development of interfaces and interactions addressing stakeholder concerns and promoting smooth human-robot collaboration.

02

Why a new system is needed

2.1 Limitations of Current Robot Control Systems

While service robots are increasingly present in public and semi-public environments, their integration remains problematic due to limitations in current control systems. These limitations affect both usability for human operators and the overall acceptance of robots in shared spaces.

Lack of intuitive control

Many current robot systems rely on either fully autonomous navigation or highly technical teleoperation interfaces. These interfaces often require specialized training or rely on command-based systems that are unintuitive for non-experts. For instance, robotic platforms using ROS-based stacks commonly depend on text-based terminal commands or rigid waypoint systems that do not support flexible human input (Quigley et al., 2009). This creates a barrier for quick intervention or real-time adjustment, which is especially problematic in dynamic environments like restaurants or care facilities.

Poor adaptability to human behavior

Most navigation systems focus on obstacle avoidance and path planning without meaningful interpretation of human intent or social context. Robots often freeze or reroute in ways that appear irrational to humans, leading to frustration or avoidance (Dragan et al., 2015). In scenarios involving shared spaces, such as corridors or seating areas, robots lack the negotiation skills that humans naturally apply.

Low transparency and feedback

Another issue is the limited feedback available to users during robot operation. Many systems fail to communicate what the robot is doing or planning, leading to a "black box" effect (Zhu & Li, 2021). When human operators are uncertain about the robot's internal state, they are less likely to trust or collaborate with it effectively.

Fragmentation of interfaces

Robot control systems are often developed as isolated tools, each tailored for a specific robot or task. This leads to fragmented interfaces with inconsistent visualisation methods, camera controls, and input handling. As a result, switching between robots or control strategies becomes inefficient, limiting the scalability and general usability of such systems (Goodrich & Schultz, 2007). On top of that, the standard of these systems seems to be that every robot has their own interface that is being set by the one loading the robot.

Figure 2.1: current robot interface

It would therefore be desirable to create a system that allows operators to use the robot from a distance, instead of having to physically walk to the robot to issue a new command.

2.2 Gaps in Current Research and Practice

While there is substantial research on robot navigation and teleoperation, these studies tend to focus narrowly on the technical aspects of movement through space.

Delivery robot optimisation

The majority of this work evaluates pathfinding efficiency, obstacle avoidance, or mapping accuracy, often in static or controlled environments. What is often lacking is attention to how these systems perform when shared with human input, or how they function in complex, real-time, and socially active environments such as restaurants, hospitals, or public spaces.

Game control optimisation

Separately, the field of game design, particularly real-time strategy (RTS) interfaces — has produced many insights into optimal control schemes, camera perspectives, and multi-agent management. However, these systems are typically closed environments with no need to account for human-robot interaction, physical presence, or real-world ambiguity. While useful for interface design principles, they rarely consider social acceptability, trust, or frustration in their original design process.

The gap

This leaves a gap at the intersection of robot control, user interface design, and real-world social interaction. Specifically, there is a lack of flexible, simulation-based platforms that allow for:

- Testing how different control strategies affect human-robot interaction quality
- Studying shared-control models where humans intervene or support autonomous systems in dynamic environments
- Exploring how interface choices (e.g. perspective, feedback, control mode) influence user trust, workload, or frustration

In short, while robot navigation and control are well-explored in isolation, there is still limited understanding of how to design and evaluate control systems that are meant to be used with and around humans, especially when these systems must adapt to changing context, preferences, or operator needs.

This project aims to fill that gap by developing a modular control system and testbed that enables experimentation with interaction-driven design questions in simulated but realistic environments.

2.3 The current role of delivery robots in restaurant environments.

Now that we have done an analysis to get a clear picture on the future environments, it is also important to analyse the reasons why a modern restaurant worker would, or would not choose to purchase them.

The most common reasons for not implementing delivery robots

Initial investment price

The initial investment for a robot server is hefty, and there are ongoing maintenance, updates, and operational costs. Many robots require adapted infrastructure, such as stable flooring, enough aisle width, charging stations and integration with existing workflow. With lower end robots costing around 8,000 euro (Proven Robotics, 2024) per unit, and higher end models being around double that price, the initial investment costs are often too high to justify the risk.

Customer perception

Another major concern relates to customer perception. Delivery robots are often viewed as slow and inelegant, especially in crowded dining rooms. This often leaves users with the feeling that the restaurant is cutting corners by saving on staff wages. This perception contributes to the association of robots with informal or low cost dining environments. Research suggests that their presence in full service or fine dining settings can reduce the perceived quality of the restaurant, which makes high end venues less willing to use such systems. (Y. Li & Wang, 2021)

Robots can also limit the social and emotional aspects of service. Human servers manage spontaneous requests, offer personalised interaction, and adjust their behaviour in response to guest needs. These elements contribute to the overall dining experience in ways that current delivery robots cannot replicate.

Where delivery robots currently add value

A primary advantage is the stability that delivery robots can offer. Many restaurants experience difficulties in hiring and retaining part time staff, which affects service continuity. Robots provide a more predictable form of labour since their availability does not fluctuate in the same way as human staffing levels. This can support long term planning and reduce the impact of labour shortages.

The novelty of robotic service also has commercial value. In certain types of restaurants, the presence of a delivery robot becomes part of the dining experience. Some customers view robotic service as an entertaining or interesting feature, which can influence their perception of the venue and may help attract visitors who are curious about the technology.

2.4 Research question

With these gaps in mind, there is a clear need for shared control systems in which multiple semi autonomous robots operate independently, while a single human operator is supported in identifying situations where human intervention adds the most value. Such systems should provide both an overview of robot activity and the means to selectively intervene through remote control, allowing human and robotic strengths to be combined more effectively. In order to optimise the amount of research done in this field, the following research question will be focussed on:

"How can a simulated restaurant environment be used to research how humans and semi-autonomous robots can work together more fluently?"

The formulation of the research question emerged from an extensive analysis of stakeholder demands, system design wishes, and recent trends in human-robot interaction within dynamic service environments. Stakeholder input emphasized the necessity for a control system that balances operator usability with minimal disruption to human traffic, efficient task execution, and clear autonomy boundaries.

These priorities were further reinforced by trend analyses highlighting the growing importance of semi-autonomous systems that support human oversight while managing complex social navigation tasks. Consequently, the research question was crafted to investigate how semi-autonomous robot control systems can be designed to optimize task efficiency and human-robot collaboration, while reducing cognitive load and physical interference in shared environments.

2.5 What we can learn from already existing semi-autonomous robot control systems.

While we can take a look at the already autonomous serving robots, it is also important to look at other systems where these systems can significantly improve performance due to its ability to bridge human intent and machine action effectively.

One of such systems is the outdoor food delivery robots often seen driving around the street

Outdoor delivery robots

Public reactions to outdoor delivery robots are mixed. While the technology is often perceived as novel or convenient, several recurring concerns appear in user studies, media coverage and municipal feedback.

One common concern is obstruction of pedestrian space. Robots may block pavements, slow down foot traffic, or stop unexpectedly in narrow areas. This is particularly noticeable for wheelchair users, parents with prams, and in dense urban environments. Companies attempt to address this by limiting robot size and speed, enforcing strict geofenced routes, and programming conservative yielding behaviour. Many systems prioritise stopping or pulling aside rather than asserting right of way.
A second concern relates to safety and predictability. Pedestrians may feel uneasy sharing space with a machine whose intentions are unclear, especially near crossings or crowded areas. To mitigate this, companies design robots to move slowly, use visible lights or sounds to indicate state changes, and follow exaggeratedly cautious motion patterns. Some robots also display simple signals or screens to communicate intent.

Vandalism and misuse form another practical issue. Robots operating in public space are occasionally interfered with, damaged, or obstructed. In response, companies use robust physical designs, tamper detection, cameras, and remote monitoring. Operational strategies such as campus based deployments or supervised neighbourhood rollouts are also used to reduce exposure.

Privacy is a further concern, as robots rely heavily on cameras and sensors. People may be uncomfortable being recorded in public space. Companies typically argue that data is processed locally, anonymised, or stored only briefly. Clear communication and compliance with local data protection regulations are used to address this concern, although public trust remains uneven.
Finally, there is a broader social concern about job displacement and the perceived replacement of human labour. Companies often frame delivery robots as handling low value or repetitive tasks rather than replacing workers entirely, emphasising human roles in supervision, maintenance, and customer interaction.

Hospital service robots

Hospitals increasingly use service robots to transport medication, meals, laundry, and medical supplies. These robots typically navigate autonomously through corridors and wards while operating within clearly defined routes and schedules. Their use is motivated by efficiency, staff workload reduction, and infection control rather than patient interaction.

One major criticism concerns safety and reliability in complex environments. Hospitals are dynamic spaces with patients, visitors, beds, and emergency situations that change rapidly. Robots may block corridors, hesitate at doorways, or struggle with lifts and crowded areas. To address this, hospital robots are designed with conservative navigation behaviour, strict speed limits, and prioritisation of human right of way. Many systems also rely on infrastructure support such as automated doors, lift integration, and predefined routes to reduce uncertainty. (Messe Düsseldorf GmbH, 2024)

A second criticism relates to social and emotional discomfort. Patients and staff may find robots unsettling, intrusive, or inappropriate in sensitive contexts. This is especially relevant in wards where patients are vulnerable or distressed. Companies respond by limiting robot interaction to logistical tasks, avoiding patient facing roles, and designing robots with neutral, non expressive appearances. Clear signalling and predictable movement are prioritised over social behaviour.

Dependence on staff intervention is another concern. Robots may stop when encountering unexpected obstacles or situations, creating additional work for nurses or technicians. This is mitigated through layered autonomy, where robots attempt simple recovery behaviours first and escalate to remote assistance or staff intervention only when necessary. Training and clear protocols also reduce frustration by making robot behaviour more predictable.

Privacy and data handling present further issues, as robots rely on cameras and sensors in sensitive medical environments. Companies address this by minimising data storage, processing information locally where possible, and complying with medical data protection standards. Transparency about sensor use helps maintain institutional trust, even if public understanding remains limited. (El-Gazar et al., 2024)

Overall, hospital robots succeed when they are framed as background infrastructure rather than intelligent agents. By focusing on reliability, predictability, and clear boundaries between human and machine responsibility, current systems aim to support clinical staff without interfering in care delivery.

Example study: Moxi
Moxi is a mobile hospital robot that autonomously delivers supplies, lab samples, medications and PPE to different points in the hospital. Staff supervise its operation as part of daily workflow: nurses and technicians load tasks, monitor progress, and may intervene if the robot encounters an unexpected situation or needs to be repositioned. Because it learns from patterns of use and staff feedback, this system exemplifies human-guided learning and semi autonomous integration with clinical routines. (Moxi — Diligent Robotics, 2024)

Self driving cars

Self driving cars are developed with the aim of improving road safety, reducing driver workload, and increasing transport efficiency. In practice, most systems currently deployed operate at partial or conditional automation levels and rely on human drivers or remote oversight to manage uncertainty and failure cases.

A major criticism concerns safety and reliability in complex traffic environments. Urban roads involve unpredictable human behaviour, informal rules, and rare but critical edge cases. Self driving systems can perform well in routine conditions but struggle with unusual situations such as temporary road layouts, unclear signage, or unexpected human actions. To address this, companies restrict operation to specific environments through geofencing, limit automation to certain speeds or road types, and design systems to request human intervention when confidence drops.

Another concern is overtrust and driver disengagement. When systems appear highly capable, human drivers may become inattentive and react too slowly when manual control is required. (Nordhoff, 2024) This handover problem has been widely criticised, particularly in partially automated vehicles. Manufacturers respond by adding driver monitoring systems, clearer alerts, and stricter requirements for driver attention. Some systems enforce regular interaction or disengage automation if the driver is not responsive.

Burden of responsibility in shared control
Legal responsibility and liability also present significant challenges. It is often unclear whether responsibility lies with the driver, manufacturer, or software provider when an incident occurs. Companies attempt to mitigate this by maintaining the driver as the legally responsible party in most current deployments, while collecting extensive sensor and event data to support incident analysis. Public trust and acceptance remain uneven. Accidents involving self driving cars receive disproportionate attention and can damage confidence even when overall safety metrics improve. To counter this, companies emphasise conservative driving behaviour, extensive testing, and incremental deployment rather than full autonomy. Transparency reports and staged rollouts are used to signal caution rather than technological bravado.

Overall, current self driving systems prioritise controlled deployment and shared responsibility. Full autonomy is treated as a long term goal (Schellekens, 2022), while present designs focus on managing uncertainty through human oversight and gradual capability expansion.

Human involvement through supervision and control
Human involvement in self driving car systems takes several forms, depending on the level of automation.

The most common method is driver supervision. In partially automated vehicles, the human driver is expected to monitor the system continuously and take over control when requested. Alerts are used to signal handover, although timing and clarity remain critical design challenges.
A second method is remote supervision and assistance. In limited deployments, remote operators may monitor vehicles, approve route level decisions, or provide guidance when the system encounters ambiguity. Direct remote driving is generally avoided due to latency and safety concerns, but may be used at low speeds or during recovery.

A third method is safety fallback behaviour. When neither the driver nor a remote operator can intervene effectively, vehicles are designed to execute minimal risk manoeuvres such as slowing down or stopping safely. This approach prioritises harm reduction over task completion.
In practice, these methods form a layered control structure. Autonomy handles continuous driving, humans manage judgement and responsibility, and system design focuses on safe transitions between the two. This shared control model reflects the current state of the field and the limits of present automation technology.

2.6 Scope

Summary of opportunities so far

The earlier research phases, including the context analysis, stakeholder mapping, and trend analysis, have revealed several key opportunities. First, there is a growing demand for more socially acceptable and user-friendly service robots in public and semi-public environments. Existing systems often lack the flexibility and clarity needed for smooth human-robot interaction, which creates space for more adaptable control strategies.

Second, the absence of a widely used shared-control framework presents a clear gap: there is a need for a system that not only improves robot control but also enables structured experimentation and evaluation of different interaction methods. This opens the opportunity to develop a research platform that supports modularity, real-time switching between control modes, and integration with simulation environments. These insights form the basis for exploring a more human-aware, research-oriented robot control architecture.

Personal strengths

A key personal strength in this project is prior experience in game design, which significantly lowers the technical threshold for prototyping complex interaction environments. While not a scientific contribution in itself, this skill set allows for fast and high-quality development of simulation tools and interfaces, expanding the range of what can realistically be built and tested within the project scope.

In addition, previous experience in the field of human-machine interaction adds depth to the project. This includes 13 months working with MARCH on robotic control systems, as well as earlier work developing a robot interface during a robotics minor. These experiences provide a strong foundation in both the technical and usability aspects of robot control, directly informing the design and implementation decisions in this thesis.

Time constraints and other limitations

Due to the limited time available for this project, it will not be possible to test this new system within an actual restaurant. That means that there is much more value to be gained from creating this in a simulated environment rather than spending time coordinating with restaurants, potential test pedestrians and general connection with the robots themselves.

2.7 Design vision

Based on the background research, scope and trend analysis, it would be safe to say that finding the perfect combination of visualisation and control tools and applying them in real life would not be possible within the provided time. Therefore, the attention wil shift towards creating a system that serves as a solid base for future researchers with the ambition to research shared control in robots.

With this in mind, the greatest value of this project can be derived from overall reliability, and a comprehensible plug-and-play system that allows researchers to come up with their own ways of alerting, and solving a conflict of movement.

“The simulation must support a spectrum of control ranging from fully autonomous robot behaviour to fully teleoperated control, and provide methods to evaluate performance across intermediate control levels.”

This means that the simulation would, in itself, gather useful data on which methods affect the overall performance and perceived cognitive load on the human operator. This is useful in itself, but without testing this in a real life situation, remains partly speculative. The real value therefore comes from the adaptability to dynamically change the parameters of the experiment, allowing it to quickly transfer itself for environments such as hospitals.

2.8 Demands and Wishes

Building on the trend and stakeholder analyses, this section outlines the specific demands and wishes identified for the design and implementation of the robotic system. These requirements reflect the practical needs and expectations of the primary users and stakeholders, as well as insights gained from research.

The demands focus on essential functionalities and constraints necessary for system feasibility, while the wishes represent desired features that enhance usability, acceptance, and overall user experience. Together, they provide a foundation for the development of control strategies, visualisation methods, and interaction designs that align with real-world requirements.

Demands:

- **The system must provide an array of tools for the human operator to exert influence over the robot**
  This addresses the need for operator ease and efficient multitasking identified in the robot operator profile and is supported by shared control research showing reduced cognitive load.
- **The system must provide an array of visualisation methods to inform the human operator on where they are needed**
  This is derived from desires to avoid operator frustration and overload, as reflected in stakeholder frustrations and modular visualisation design principles.
- **The system must accurately simulate the unpredictability of a restaurant environment**
  This reflects the need for an actual way to test the aforementioned visualisation and control methods, rather than cherry picking isolated situations.
- **The system must allow operators to multitask efficiently without excessive information or interruptions.**
  The main goal is to allow researchers to find the option with the perfect effort/effect ratio, rather than the best performing features in general.
- **The system must provide transparent robot behavior to the operator through clear visual and control feedback.**
  Supported by trend data emphasizing perceived empathy and communication as key factors for acceptance.
- **The system must have a robust and modular software architecture to support maintenance and scalability.**
  Follows general best practices for system feasibility and meets technical staff needs.
- **The system must operate in real-time with minimal latency.**
  Essential for practical usability and safety, ensuring timely responses in dynamic environments.
- **The system must ensure the operator maintains ultimate control over robot decisions during all phases of operation.**
  Reinforces the importance of human oversight to reduce job insecurity and increase trust, as indicated by stakeholder analysis and trend data.

Wishes

- **The system should support the most common methods of robot control with the default settings.**
  Based on the desire for flexibility identified in stakeholder needs and shared control literature.
- **The interface should include toggleable visualisation methods tailored to user preference.**
  Responds to user comfort and cognitive load concerns, supported by modular visualisation design.
- **The system should support customizable interface options for individual operator preferences.**
  Aligns with operator usability wishes and enhances overall desirability.
- **The robot behaviour should adapt dynamically based on human proximity and environmental context.**
  Supports safe navigation and conflict avoidance, consistent with stakeholder and trend insights.
- **Training support such as tutorials should be integrated into the system to ease operator onboarding.**
  Addresses workforce skill uplift needs and reduces fear of obsolescence.
- **The system should scale smoothly to support multiple robots and complex environments.**
  Facilitates broader applicability and future-proofing as identified in managerial concerns.

Conclusion

The demands outlined here are essential requirements derived from the analysis of stakeholders, trends, and practical system needs. They establish a foundation for the development of a robotic system that is safe, efficient, and trustworthy.

Ensuring intuitive control, clear communication, and operator authority will be critical to achieving both technical feasibility and user acceptance. These demands guide the design process toward solutions that meet real-world challenges and support successful human-robot collaboration.

03

Setting up a socially complex research environment

3.1 Why a Flexible Robot Control System is Needed

As discussed in earlier sections, the integration of service robots into human environments offers clear potential but is consistently limited by the robots' lack of social intelligence. While they are competent at basic tasks such as navigation or transport, they fall short in interpreting nuanced human behaviour, responding to real-time context, or resolving complex interactions. These limitations restrict their practical usefulness and, more importantly, their desirability in public-facing roles.

The aim of this project is to explore how human input can be reintroduced in a meaningful way, precisely at the moments when robots reach their limitations. By letting robots handle the repetitive or 'stupid' tasks—such as moving from point A to point B—and allowing humans to intervene only when social or situational awareness is required, the system becomes more than just efficient. It becomes desirable. In this ideal arrangement, operators do not lose their connection to the human environment. Instead, they are freed up to engage more meaningfully with it. Ironically, by delegating the mechanical parts of the job to robots, the human operator can become more human in their role.

Although this scenario might seem overly optimistic, it serves as a useful design ambition. The system does not aim to eliminate the human role, but to restructure it in a way that prioritises high-impact interaction over low-level oversight. Robots do not need to be perfect if the system supporting them allows for timely, low-effort human correction.

Effort versus Performance

This brings us to the core trade-off that underpins the system's design: effort versus performance. In any multi-robot setting, it is not feasible for a human operator to manage every decision or movement manually. The cognitive load would quickly exceed what any individual could handle. Therefore, a basic level of autonomy is assumed. Each robot must be able to function independently under normal conditions, without requiring continuous input.
The operator's role then shifts from being a constant supervisor to a strategic enhancer. Their time is better spent improving customer experience than micromanaging robot paths. The goal is not to provide fine-grained controls such as speech-based commands, real-time joystick navigation, or adjustable acceleration curves. While these tools might seem impressive on paper, they are grossly inefficient in practice. They risk turning every minor navigation issue into a mentally exhausting task for the operator.

Instead, the system should make it easy for operators to make impactful interventions with minimal effort. A good system does not just allow humans to take control. It allows them to take smart control. This means clear visualisations, simple override tools, and an interface that highlights when and where attention is needed most. It is not about offering maximum control, but about offering meaningful control where it counts.

This principle—low effort, high impact—guides the development of the control strategies and visualisation methods discussed in the next section. The question is not just how the operator should interact with the robots, but when and why. The design must support those decisions at every level.

3.2 Specialised Movement Strategies

A top-down view offers an excellent overview of the environment, making it easy to spot when a robot requires intervention. It allows the operator to monitor multiple agents at once and quickly identify potential conflicts or inefficiencies. However, this perspective becomes less effective the moment direct control is needed. Navigating a robot from a top-down view involves too many layers of abstraction, requiring the operator to interpret both the robot's orientation and its surroundings through multiple visual overlays and indirect inputs.

This can be compared by parking a car in a tight spot. The driver is forced to first use their mirrors, then while looking through the mirrors, look for visual cues that might indicate the distance left based on limited perspective and finally moving the car. On an unconscious level, the driver has to go through all these “layers”, as opposed to just standing behind the car and seeing how much space is left behind the car.

To resolve this, the system allows the operator to take a “shortcut”, and override the camer angle and control scheme for whichever situation that requires it. For example, a first-person view might be more suited for fine navigation in crowded spaces, while a third-person trailing view could help with spatial awareness during path correction.

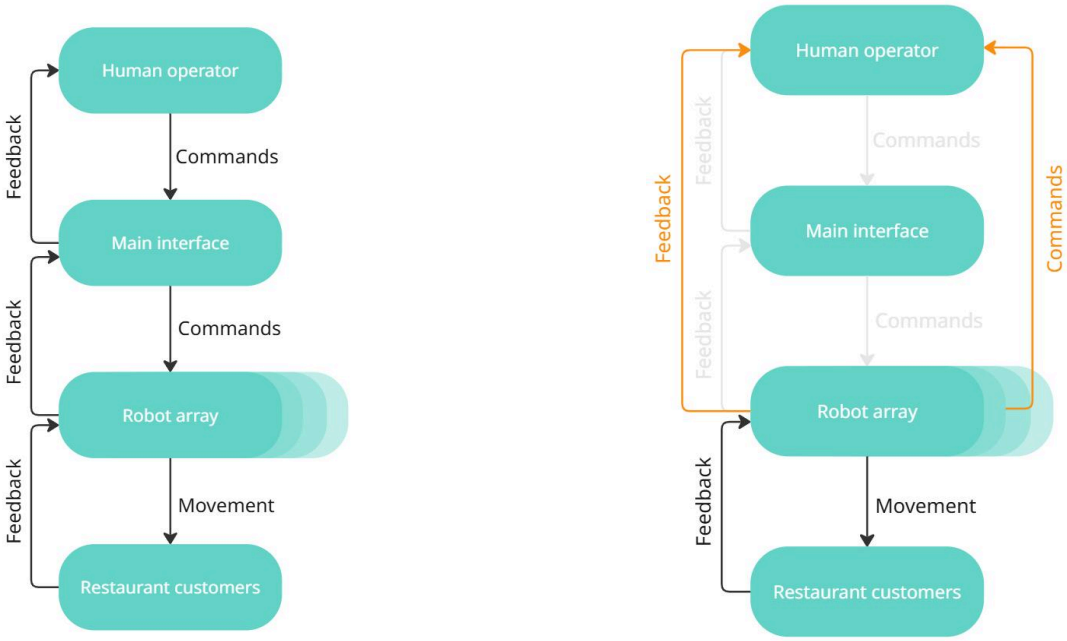


Figure 3.1: layers of information processing between customers and operators

Therefore we can conclude that finding one or several movement strategies that allow the user to consistently take over the robot with minimal cognitive load is needed to create a system that allows one robot operator to increase the desirability of several service robots.

3.3 Utilising a simulation instead of real-life

The simulation is the core of this research environment. Instead of relying on physical robots in a real-world setting, we use a fully digital system to test how operators interact with robots, how control strategies perform, and how different visualisation methods affect clarity. This choice is not just about convenience. It allows for rapid testing, flexibility, and scalability that real-world setups cannot match.

Running the experiment in an actual location, such as a restaurant, would give access to the complex social dynamics that naturally occur in those spaces. Humans behave unpredictably. They change direction without warning, hesitate, or signal intent through body language. These are valuable aspects to study, but using a real robot also means dealing with hardware, networking, physical constraints, and live sensor feedback. On top of that, the operator would still need a digital interface to manage the robot, which would require its own development pipeline.

Flexibility

By keeping everything in simulation, we can create new robot swarms instantly, adjust the layout of the environment in seconds, and test different scenarios at scale. The simulation gives full control over variables while speeding up iteration time. It allows us to focus on the interaction itself rather than spending weeks setting up and debugging a hardware system.

Digital twin

One limitation of a fully simulated environment is the lack of real human behaviour. While we can simulate people moving through the space, they do not carry the same nuance or unpredictability as real humans. A possible solution is to add multiplayer functionality. This would allow real human participants to control virtual characters and perform randomly assigned tasks in the environment. Although this does not fully replicate subtle communication cues like posture or gaze, it introduces the element of unpredictable intent and creates more realistic interaction patterns between humans and robots.

There is also the option of working with a digital twin. In practice, it would be relatively straightforward to stream camera feeds from an existing space, such as a restaurant, to give operators a visual overview. However, the true benefit of a digital twin lies in its ability to scale. In large, complex spaces like hospitals or care homes, it is almost impossible to create a complete overview using physical tools alone. A digital twin can offer a real-time, interactive visual model of the environment, giving operators an efficient way to supervise and make decisions.

This leads to the idea that there are actually two parallel worlds in this system. One is the physical world where the robots operate. The other is the digital world that serves as the operator interface. These two worlds are always synchronised. The operator sees the environment through virtual cameras that are physically attached to the robots. This creates a continuous feedback loop between action and perception. It allows for a form of control that feels direct, but still benefits from the visual enhancements and clarity of the digital world.

Conclusion

In this setup, simulation is not really a compromise, but rather an isolation of certain specific problems. It enables controlled exploration, faster development, and ultimately a more scalable approach to human-robot interaction.

3.4 Software architecture

In order to determine the form of system we shall be using, we shall need to map out all of the components, and how they interact with each other. As the complexity of the project increases, so will the total amount of connections. It is therefore essential that we map out the individual scripts, and to which objects they are attached to.

It should be noted that the scripts running on the robot are copied for the total amount of robots present in the scene.

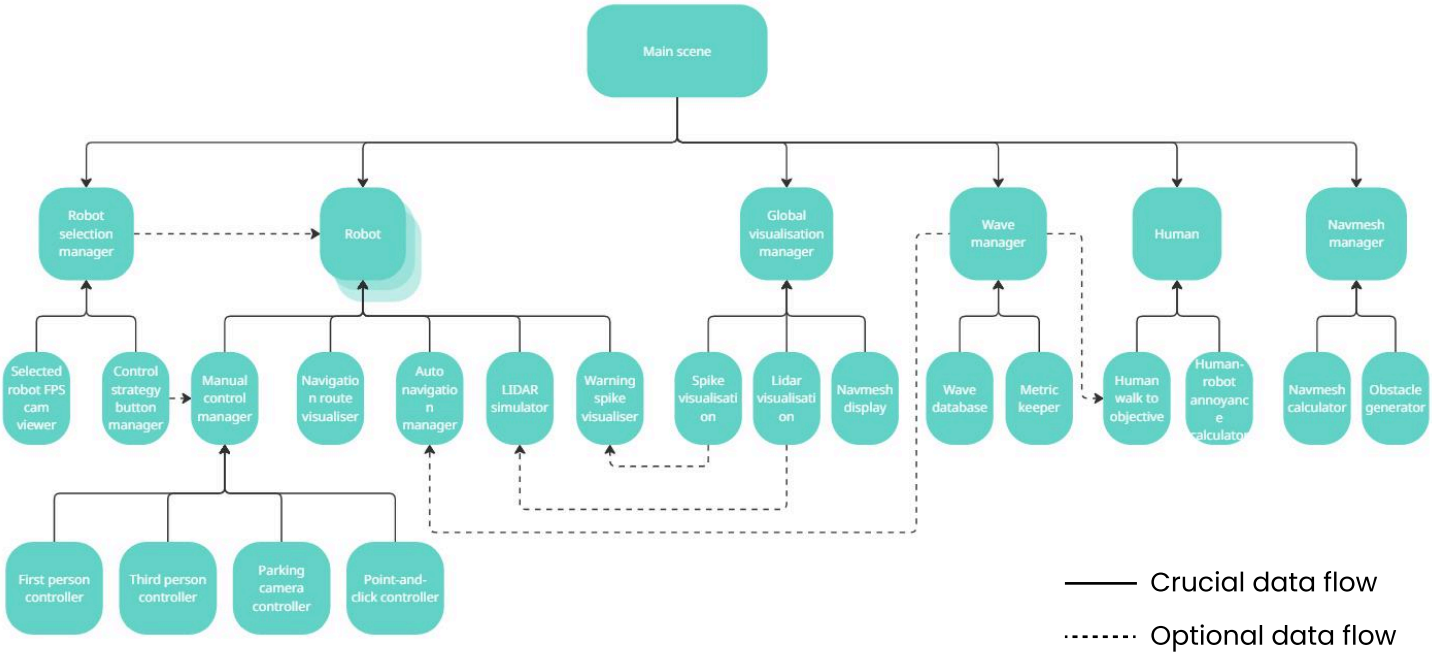


Figure 3.2: A full overview of the software architecture of the whole system

Despite a lot of work being put into creating all these scripts, no major design choices were made during this process that have not been discussed before. A larger version of this tree can be found in the appendix.

3.5 Early inspiration

While developing a system from scratch would be possible, there is still much to learn from already existing systems. Therefore, the decision was made to draw inspiration from already existing systems, such as robot control systems, or video games.

UI design in strategy video games

While technically not the same as a robot control software, video games with a focus on strategy and managing multiple entities simultaneously often struggle with keeping that information accessible and digestible for the user. It was quickly noted that older games showed a clear favour to high-contrast bright colours on static elements, usually owing to the fact that these were made for low resolution displays and thus had to be visible despite low visibility.

As games got more modernised, so did the complexity of the UI. It seems that generally, high contrast icons were replaced with more elaborate designs that seemed to prioritize style over function. It should however noted, that communication such as alerts, is increasingly done with animations and general movement, instead of iconography and colours. It was also noticed that newer games often have much more use cues on which buttons are “clickable”, and which ones are mere windows for displaying information.

Figure 3.3: four examples of videogames displaying dynamic information

It should also be noted that over the years, the average consumer of such games has more experience in the genre on average. This is mostly due to popularisation of the genre, and standardisation of certain iconography and colours. This means that aesthetic value usually has a higher priority than readability, which is a pitfall that must be avoided for this specific project. Another pitfall would be to make the assumption that every user is familiar with the controls and aforementioned standardised visual cues of these video games.

Air traffic, sea traffic and emergency call interfaces

On the opposite side of the spectrum, professionals around the world use user interfaces that maximise function over form. These systems are not made to be appealing for a consumer in any sense, and are prioritised with safety as the number one priority. This also means a near global standardisation of visualisation that has barely changed for decades. During a personal conversation with a ship navigator, they expressed a general dislike for the system due to their unpleasant design.

If these systems were to be completely redesigned from scratch, they would have looked a lot different. This means that while it is important to take over the iconography of these systems, and follow their main design priorities in the interface that will be used for the project, an exact imitation of these systems should be avoided.

Certain sounds were also considered to be taken over from these environments, but due to the noisy nature of the restaurant environments in general, the decision was made to leave these parts of the research out of the report.

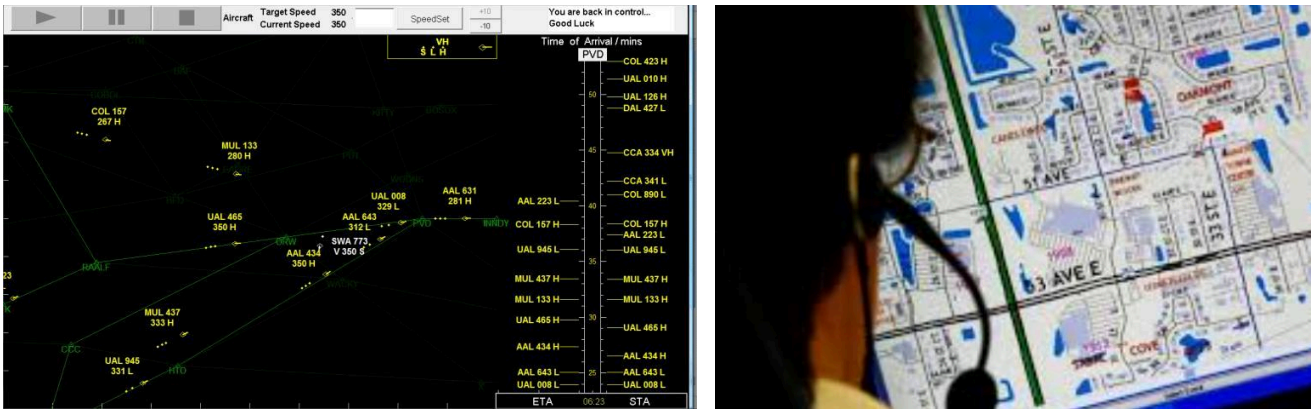


Figure 3.4: an air traffic control interface next to a U.S 911 dispatcher interface

3.6 Pilot experiment setup

The pilot experiment is designed to evaluate the effectiveness of the proposed control strategies and visualisation methods within a simulated environment. The primary objective is to identify which combinations support safe, efficient, and user-friendly operation of the robotic system. Additionally, the experiment aims to gather qualitative feedback from participants to uncover usability issues and areas for improvement.

This setup provides a controlled setting to test system functionality and operator interaction before larger-scale trials. The results will inform refinements to the design, ensuring alignment with user needs and practical constraints.

Methods used in the experiment

Participants are asked to supervise multiple waves of robot deliveries within the simulation. Each wave presents the same set of delivery tasks, ensuring comparability across runs. During these waves, participants can select from different control strategies to operate the robots and are able to toggle various visualization methods to assist in threat assessment and navigation.

After completing all waves, participants fill out a survey to provide subjective feedback on their experience, including perceived workload, trust in autonomy, ease of control, and frustration levels. This setup allows for both objective performance measurement and insight into user preferences and perceptions.

Design Value of the Pilot Experiment

- It identifies where users experience friction with different control modes, highlighting areas for interface and interaction improvement.
- It provides insight into how users perceive and trust semi-autonomous systems, informing the design of clearer autonomy cues and control handovers.
- It reveals patterns in how users manage attention and control under dynamic, time-pressured conditions, which can guide the development of more intuitive oversight tools.
- It offers concrete data on when and why users choose to intervene, supporting the design of modular control systems that adapt to varying levels of user engagement.
- It enables a human-centered evaluation of control strategies beyond efficiency, focusing on perceived control, ease of use, and frustration—critical factors in real-world adoption.

04

Design choices for phase 1

4.1 Finding the right control strategies

In order to control the robots, a set of control strategies is required. Based on the current goals and requirements, it is important to not only explore new approaches but also consider existing control methods from related domains. Examples include systems used in automotive technology or interaction techniques found in digital interfaces.

In the end, it was decided that the four distinct control strategies are:

1. First-person control
2. Third-person control
3. Point-and-click control
4. Parking camera control

Clarification on the term “Control Strategies”

The term control strategies is used because these are not simply input methods. Each strategy represents a specific combination of camera perspective, interaction method, and level of automation. This allows the system to adapt to different user needs and situational demands within the environment.

Selection of control strategies

Only four control strategies were chosen to avoid giving the operator too many options. While it would be possible to mix different perspectives and control types into many combinations, these four were based on real-life systems. They are inspired by ways people already control machines or systems in everyday situations, so there was no strong reason to invent completely new ones.

This approach also fits with the double diamond model we are using. At this stage, the main goal is to build something that can be tested. A smaller set of clear strategies is more useful for a pilot test than a large number of unclear options. Feedback from participants can then be used to improve or adjust the strategies later on. If people react strongly to certain parts—positive or negative—that will help guide the next steps.

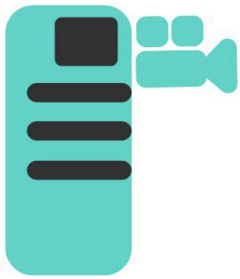
The strategies were tested in advance by the researcher and a small number of individuals. Based on that early feedback, they were considered clear, usable, and ready for testing, even though this was not a formal evaluation.

First-person control

The user controls the robot directly from its own point of view, as if seeing through its eyes. Movement is handled with standard directional input, allowing for detailed, hands-on control.

Real-life inspiration:

- First-person video games
- VR training simulators
- Telepresence robots



Pros:

- High precision in narrow spaces
- Natural feel for users familiar with first-person interfaces

Cons:

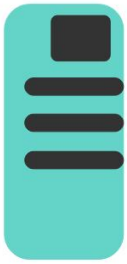
- Limited situational awareness
- Can be disorienting, especially with camera shake or poor lighting

Third-person control

The camera follows behind or around the robot, allowing the user to control it while still seeing the environment from an external point of view.

Real-life inspiration:

- Third-person games
- Drone control systems
- Remote inspection tools



Pros:

- Better overview of robot's surroundings
- Easier to correct for pathing errors

Cons:

- Less spatial precision compared to first-person
- Can cause occlusion or camera confusion in tight spaces

Parking camera control

The robot is shown from fixed external camera angles, such as a top-down or rear-view. The user steers the robot manually from these views, often switching between them for better positioning.

Real-life inspiration:

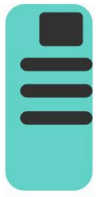
- Automotive parking assist systems
- Rear-view and surround camera interfaces in vehicles
- Forklift or warehouse robot camera setups

Pros:

- High spatial control for tight maneuvers
- Multiple views help avoid collisions

Cons:

- Slower than other methods
- Requires frequent camera switching
- Not suited for longer-distance navigation



Point-and-click control

The user clicks on a location, and the robot automatically navigates to it using pathfinding. There is no need for continuous input once the destination is set. Multiple waypoints can be placed in series.

Real-life inspiration:

- Real-time strategy (RTS) games
- Warehouse robotics interfaces
- Human-robot collaboration tools in logistics

Pros:

- Very low cognitive load
- Efficient for basic navigation tasks

Cons:

- Relies heavily on pathfinding quality
- User has little control once movement starts

Interface Design

To support quick recognition and reduce cognitive load, each control strategy is linked to a visual button with an intuitive icon. The first-person mode uses an eye icon, third-person uses a security camera icon angled to match the actual camera perspective, and the parking view uses a top-down image of a car.

The buttons are placed in order of increasing camera distance, from first-person to parking view, helping users quickly associate each option with its level of visual detachment.

First person Third person Parking

(The point and click is available at all times)

Summary and Testing Approach

The four selected control strategies offer a varied set of interaction methods, each balancing different levels of manual control, automation, and situational awareness. By choosing strategies based on existing, proven systems, the aim is to provide participants with intuitive controls while still allowing for clear differences in user experience and performance.

These strategies will be tested in a wave-based pilot setup, where participants control the robots in a simulated restaurant environment. Each wave presents a new set of delivery tasks and human obstacles, allowing the strategies to be tested under dynamic conditions. Participants will either switch between strategies or use one fixed method per wave, depending on the scenario.

Throughout the test, both objective and subjective data will be collected, including task completion time, robot-human conflicts, manual versus autonomous control usage, and user-reported frustration levels. This will help identify strengths and weaknesses in each strategy, and reveal user preferences when dealing with real-time robot interaction.

The goal is not to find a single “best” method, but to understand how different strategies influence performance, control preference, and mental workload in a shared environment.

4.2 Determining the research environment

In order to research a restaurant environment, A series of typical conflict scenarios between humans and robots have been identified and used as the foundation for modeling the experimental environment. These scenarios represent frequent points of interaction where human and robot paths intersect or compete, such as narrow corridors, shared doorways, and service areas with high pedestrian traffic.
By incorporating these conflict zones into the simulation, the environment reflects realistic challenges that operators are likely to encounter. This approach allows the pilot experiment to assess how well different control strategies and visualisation methods help operators manage these interactions, minimizing delays, frustration, and safety risks.

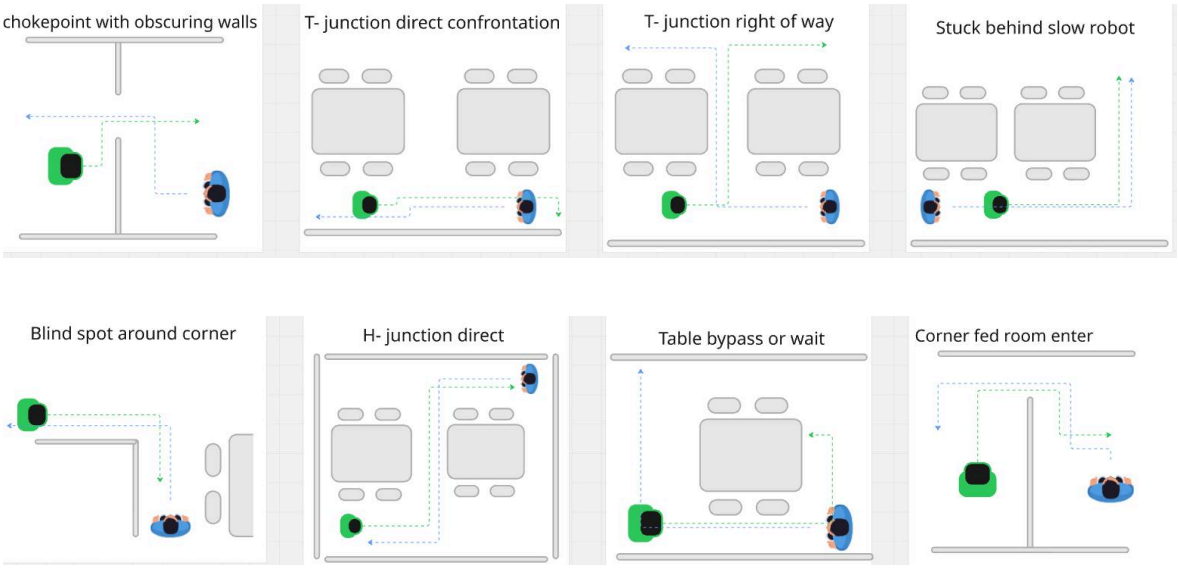


Figure 4.1: Different types of navigation archetypes

Modeling these conflicts into the main simulation explicitly supports a focused evaluation of system performance under conditions that closely mirror real-world operational demands. The end result of this analysis was a general map that has all of the situations listed above as possible patterns.

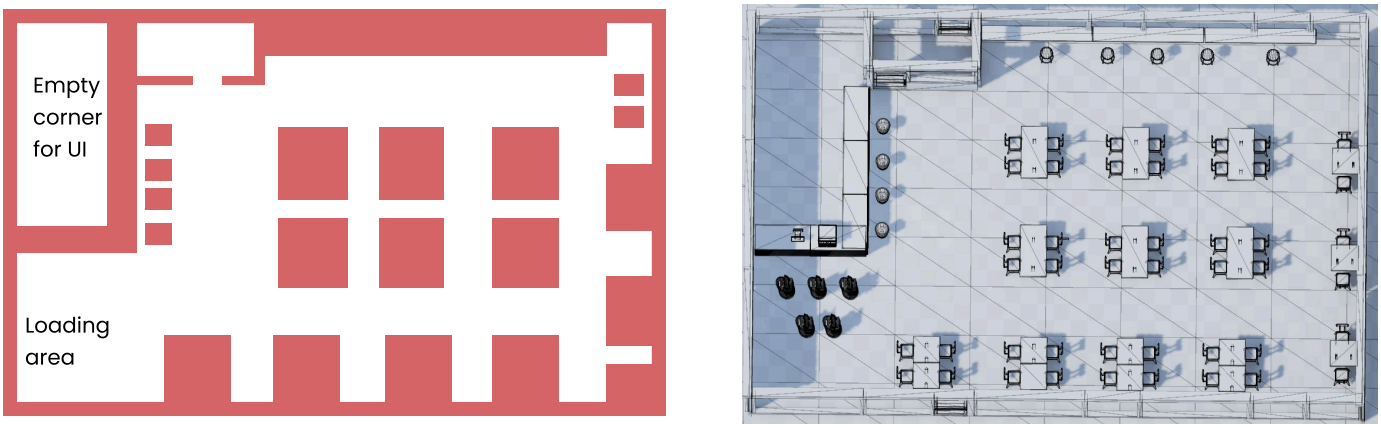


Figure 4.2: A schematic overview of the restaurant

4.3 Building a custom rendering engine

For the actual visualisation of this interface, we use Unity's Universal Render Pipeline (URP) to build the visual layer of the simulation. URP gives us fine control over how things are rendered while keeping performance overhead low, which is essential for running smooth real-time simulations without compromising on clarity.

Basic diffuse

We start from a completely blank engine. The goal is to build up a visual system that's optimised for readability, not realism. The first step is adding diffuse shaders. Without them, the scene is just a flat white void. with some silhouettes in it if objects happen to hav a different colour. The diffuse shader works by simply changing the lightness value of a surface based off the amount of light it hits.

Figure 4.3: No diffuse compared to diffuse

Highlighting using shadows

After getting some basic light interaction in place, we turn to contrast. At first we tried using outlines to make objects stand out. It did the job, but also made the scene noisy and overloaded. Every object having a thick cartoonish edge directly conflicted with our goal of keeping the interface as clean as possible.

The alternative was to strategically place shadows behind the objects. Shadows help ground the objects and give the operator a sense of depth without screaming for attention. This does create a problem where the engine has to calculate and render these shadows every single frame. This will be solved later.

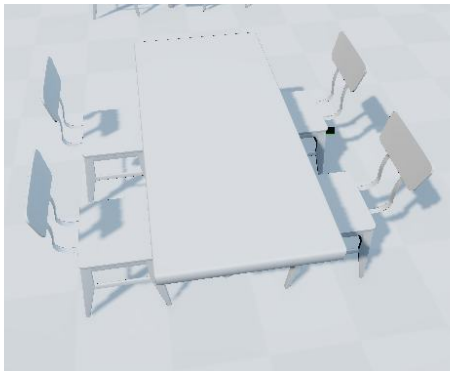


Figure 4.4: The same table with shadows

Adding ambient occlusion and global illumination

To further the amount of contrast without creating too much visual clutter, the choice was made to integrate ambient occlusion and global illumination. In real-life, light rays tend to bounce around from their initial source. This means that light has a lower chance to end up in corners or creases, which means that these are generally darker. Lucky for us, we can simulate this effect as well using ambient occlusion.

In the same sense, global illumination essentially fakes real lighting by applying extra light to objects that tend to "stick out" a bit more.

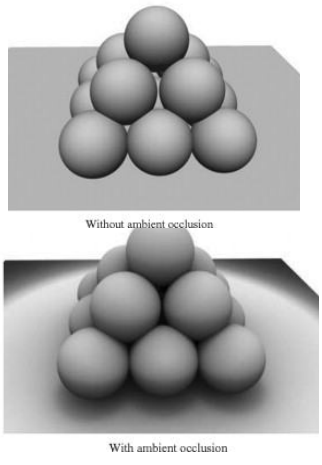
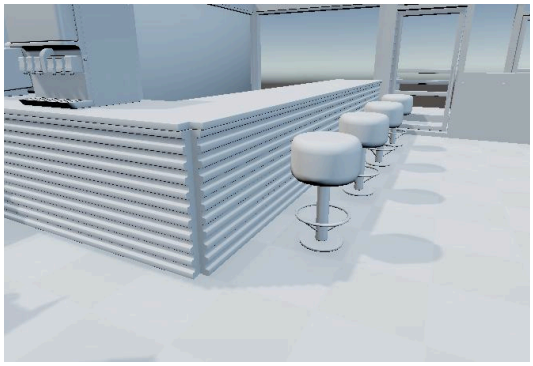
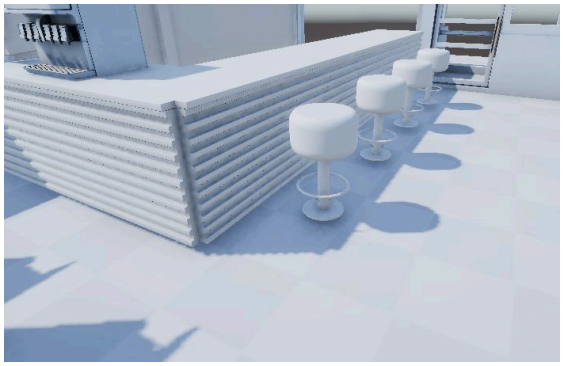


Figure 4.5: Ambient occlusion

Figure 4.6: Global illumination comparison



Before global illumination and ambient occlusion



After global illumination and ambient occlusion

Perspective vs orthographic camera

In terms of perspective, the choice had to be made between an orthographic vs perspective camera. The former is usually used for a top down view perspective in software, and the latter is more akin to a camera lens or an eye. Despite the fact that the perspective camera slightly warps the edges of the screen, the first tests showed that this felt more natural and allows the user to see more depth.

Perspective

Orthographic

Optimising the rendering engine

After some first tests, it became apparent that the constant calculation of the diffuse lighting, ambient occlusion, shadows and global illumination took about 70 percent of our frame time. While the resulting framerate could be considered stable enough, it was important to optimise this, as running this software on limited hardware would impose major limitations, as well as consuming more power than needed during prolonged use.

Lightmapping

To solve this, the choice was made to make use of lightmapping. This works by creating a seperate layer of textures that acts as a mask to display shadows. This means that we only have to perform one big calculation to create all the shadows from a certain perspective, and export that data to a separate texture map. To do this, it was necessary to go through every single object, and flag them as being “static”. This means that dynamic moving objects, such as robots, will not have any associated lighting data on them.

While this was a big undertaking on itself, it does give us the performance boost to get our renders to around 40 percent of the frame time. This is nearly half of the rendering that we had before.

4.4 Colour domains

In the design of the simulation environment, a consistent colour domain is used to help robot operators quickly interpret what they see. Visual clarity is essential when managing multiple control strategies and monitoring robot behaviour, especially under pressure. For this reason, distinct colours are assigned to different functional categories.

Green: navigation

Green is exclusively reserved for navigation-related elements. This includes the robot routes, the waypoints themselves, and any button or UI element related to directing movement. Selecting the robots also turns them green as well. By keeping all navigation cues within the same colour domain, the operator can immediately recognise which elements are about pathfinding or spatial orientation without needing to interpret each element individually.

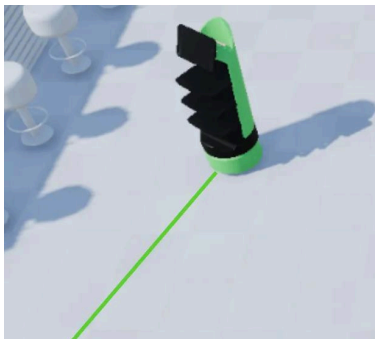


Figure 4.7: Navigation colours

Red: threat assessement

Threat detection is presented using a completely different palette. Red and orange are used for anything that signals potential danger or friction. This includes the visualisation spikes that indicate high levels of perceived threat or conflict, as well as the lidar-based visualisation dots that track dynamic objects or people. These colours signal urgency or caution, which makes it easier for the operator to differentiate them from benign elements in the scene.

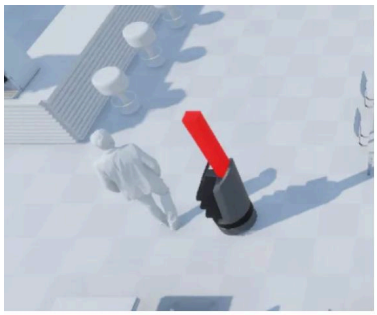


Figure 4.8: warning colours

White/blue neutral environment

The environment itself is styled using mostly white with a slightly bluish tint. This choice is deliberate. It offers a neutral background that prevents overstimulation while still allowing enough contrast for both the green and red elements to stand out. The blue tint cools the overall tone of the world, giving the interface a cleaner and more focused look. This also helps avoid the visual fatigue that could come from staring at a pure white or overly saturated background for extended periods of time.

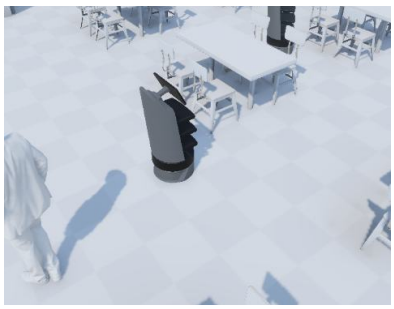


Figure 4.9: Neutral colours

Grey/black interactive elements

Interactive elements such as robots or buttons are made grey to reflect this better as well. As buttons do not move, and idle robots require no attention, this colour demands appropriate attention.

Together, these colour domains support faster recognition, reduce cognitive load, and help the operator make better decisions under pressure.

4.5 Visualisation methods

Introduction

In addition to the control strategies, a set of visualisation methods was developed to support the operator during robot navigation. These visualisations aim to improve situational awareness, highlight potential threats, and help the user make quicker and more informed decisions in shared environments.

Based on the demands and wishes defined earlier, the visualisation system needed to remain flexible, informative, and non-intrusive. Operators should be able to select the type of feedback they prefer without being overloaded with unnecessary detail. For this reason, the visualisations were designed to be modular and can be toggled on or off individually using UI buttons.

Most of the visualisations are focused on helping the user assess potential risks or robot-human conflicts, particularly in areas where navigation is complex. In addition to these, an extra visualisation method is included which displays the robot's active NavMesh. While this is not directly helpful for threat assessment, it can offer insight into how the robot perceives and plans its movement through the environment.

Each method is based on real-world systems or interface concepts and has been tested internally for usability and clarity.

Risk Visualisation

Visual spikes appear around the robot and grow in size and brightness when a threat is nearby. Spikes become more intense in areas where robot-human conflicts are likely.
Real-life inspiration:

- Proximity warning systems
- Stylised hazard indicators in VR and simulation tools

Pros:

- Clear and immediate feedback on nearby threats
- Easy to interpret at a glance
- Visually striking, works well with all control perspectives

Cons:

- Can become distracting in crowded scenes
- May block part of the camera view in close quarters

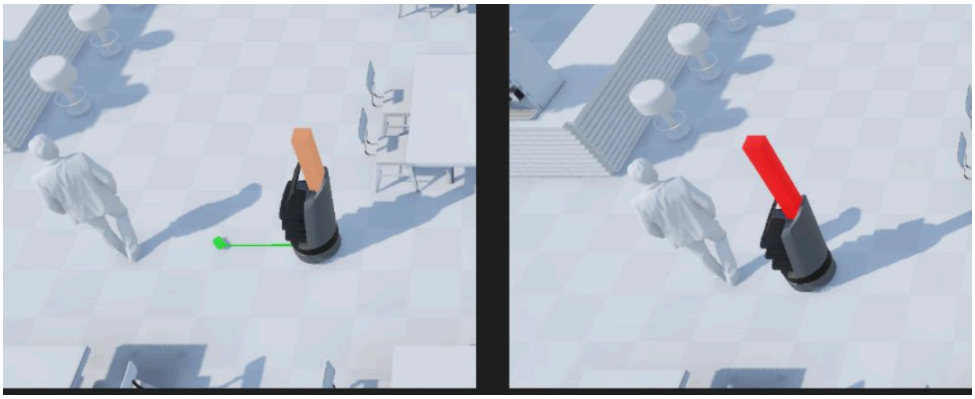


Figure 4.10: Rising severity of the risk visualisation

Lidar Dot Scaling

Lidar points around the robot scale up or increase in intensity based on proximity to humans or obstacles. Gives a more “sensor-based” feel to threat detection. This method would allow for more precision on the location of the threat, but would also create a lot of noise in the real world. To combat this, some artificial noise is added to simulate the chaotic nature that often comes with point clouds such as these.

Real-life inspiration:

- Actual Lidar data visualisations
- Robotics development tools (e.g., ROS visualisation)

Pros:

- Gives a continuous, passive sense of space
- Feels grounded in robotic logic and sensor data
- Subtle but informative

Cons:

- Less noticeable under time pressure
- Not as intuitive for users unfamiliar with lidar output
- Works from a central projecting cone, meaning that a singular sensor would often suffer from blind spots to the sides of the robot
- Is often noisy and slightly less predictable

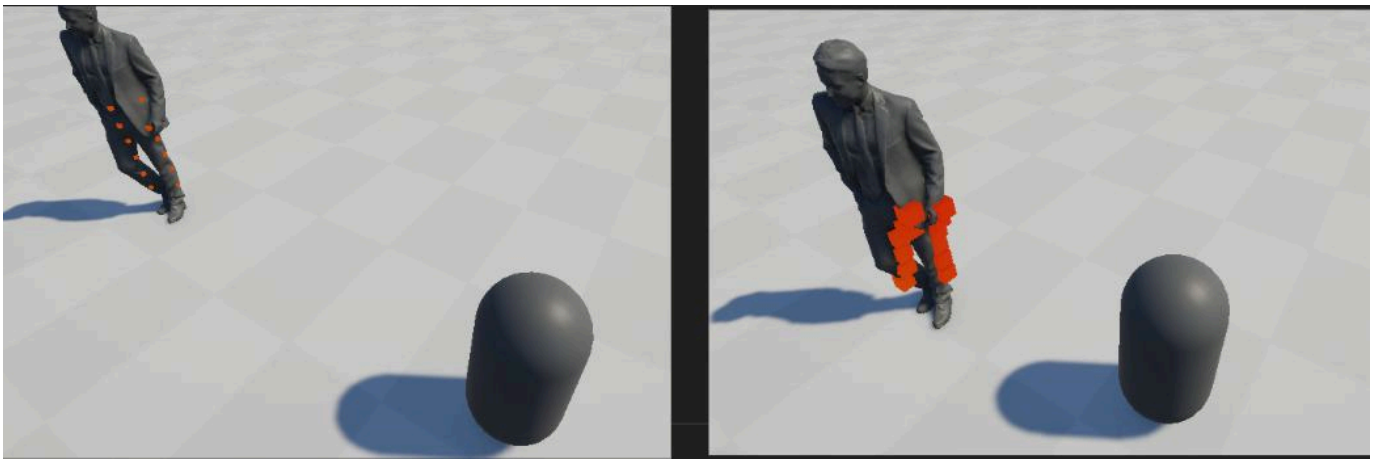


Figure 4.11: Comparison in the density and spread of LIDAR dots

NavMesh Display

Displays the walkable area used by the robot's navigation system. Shows how the robot sees the environment, including which areas are blocked or unreachable.

Real-life inspiration:

- AI navigation debugging tools
- Game development visualisation tools (e.g., Unity NavMesh)

Pros:

- Useful for understanding robot decision-making
- Helps diagnose navigation issues
- Visually interesting to observe during autonomous movement

Cons:

- Not directly useful for threat assessment
- Adds visual clutter if combined with other overlays

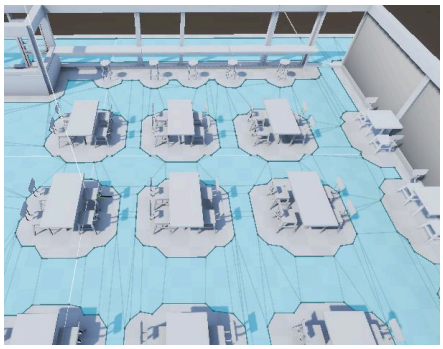


Figure 4.12: The calculated NavMesh

Summary of Visualisation Methods

The selected visualisation methods provide operators with multiple ways to perceive potential threats and navigate complex environments. Each method offers a different perspective:

- The spike indicator uses visual intensity and size to highlight danger zones, allowing rapid threat recognition without overwhelming the operator.
- Lidar points display raw sensor data, giving detailed environmental awareness useful for advanced users.
- The navmesh display reveals the robot's navigable areas, aiding in understanding path planning but less directly relevant for threat assessment.

Together, these methods offer a modular and flexible visual toolbox designed to balance situational awareness with cognitive load

Testing Approach

To evaluate the effectiveness of the visualisation methods, a pilot study will be conducted with participants representing the intended user groups. The study will measure:

- Threat detection accuracy and response time under each visualisation condition.
- Operator workload and cognitive load, using subjective questionnaires and performance metrics.
- User preferences and perceived usability, gathered through post-task interviews and surveys.
- Impact on task success rate and robot navigation efficiency.

Data collected will inform which visualisation techniques best support safe and efficient operation, and how they can be combined or adapted for diverse user needs.

Live camera view for closer inspection

Complete reliance on a digital twin would not always be preferable. The main goal of the digital twin is to have clear control over the amount of visual stimuli the human operator receives. As the digital twin is already based on a real virtual world, we can therefore also offer a live feed in order for the user to assess the situation with more detail.

Corner camera

The idea of the corner camera is to first have the user assess a potential problem from a distance. Seeing as most potential conflicts of movement are frontal in nature, simply selecting the robot that the user finds most interesting thus shows a direct camera feed of that robot. As nothing notable is to be seen in the top-left corner, a big visualisation of the camera can be shown without blocking too much of the digital twin.

This puts more emphasis on the own responsibility of the user, which could potentially have more effect, but would also cost more effort. That could take cognitive capacity away that could otherwise be used for other matters.

Figure 4.13: the corner camera

Popup camera

On the other end of the spectrum, we have the popup camera. This camera activates automatically when the self assessed risk of the robot becomes too high. This comes with the benefit of not putting that responsibility on the user, but could also cause an over reliance on the robot's autonomy. This is a problem that often also presents itself with semi-autonomous cars.

It also allows for multiple camera feeds to be active at the same time, as one pops up at the physical location of the robot. This could however, also clutter up the screen.

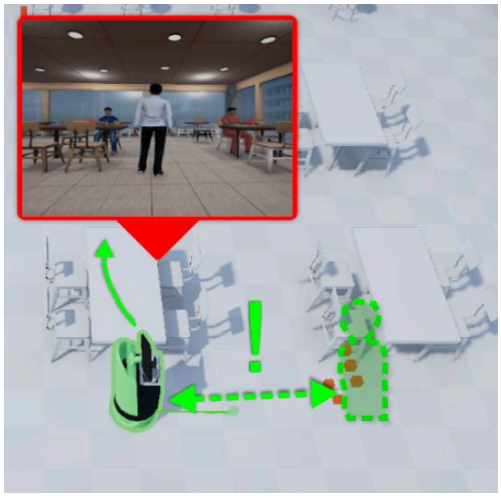


Figure 4.14: the popup camera

4.6 Creating a comprehensive stopping system

The main phenomenon we are trying to avoid with this system, is to stop the conflict of movement that occurs when a human and a robot bump into each other. It is therefore crucial for the experiment to design a system that simplifies and simulates in a way that we can control.

How do we define a conflict of movement?

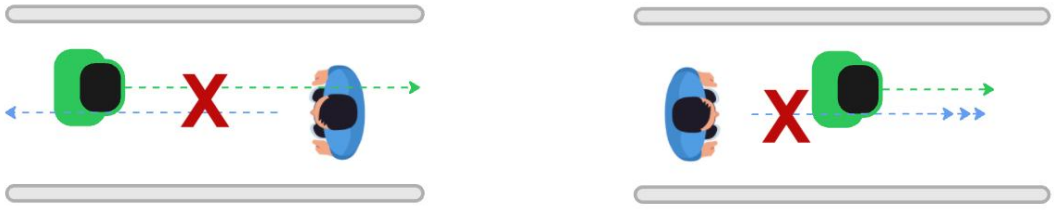
To first set boundaries on what we define to be a conflict of movement. For the sake of this experiment we will therefore make several assumptions:

- Humans are capable of judging the trajectory of other humans, and will therefore not really stop for other humans.
- Robots are aware of their own presence, as well as the location of other robots. We can therefore assume that robots will never make an active “emergency” stop for other robots.
- Humans vs humans, or robots vs robots will not stop, but instead just slide past each other within close proximity of one another.

It is therefore our main concern to focus on the “collision” between humans and robots only.

Types of conflicts

After taking the average of about 6 delivery robots, it was found that the average speed of a robot is about 0.8m/s. Humans on the other hand, have an average speed of 1.4 m/s. (CK-12 Foundation, 2025). That means that beside the naturally opposing directions problem, we also have a situation where the human tries to walk against the back of the robot because of this difference in speed.



1. Direct confrontation conflict

2. Overtake conflict

Figure 4.15: The first set of conflicts

The final two types of conflict were only spotted during testing, which are the T shaped collisions where the robot runs in the side of the human with the human not being blocked by the robot, or the other way around. (This phenomenon is also known as a broadside T-bone collision when dealing with cars.)

This would give us the next two types of conflicts, putting the total type of conflicts on a total of 4.

3. T-bone on human

4. T-bone on robot

Figure 4.16: The second set of conflicts

Simulating the logic of stopping

Now that we have defined which types of conflicts will occur, it is now a matter of programming the actual halting of movement of both agents. To do that, we must first make some assumptions:

- Stopping is only important if there is something in front of you.
- Once something is to your sides it no longer matters for your stop
- Objects directly in front of an agent are more of a threat than ones diagonally on the front.

If a shape was to be made using these parameters, the result would have a shape that is depicted in this image. While a 2D shape is useless in itself, it is possible to trace this into a 3D mesh. This mesh can then be used as a collider object in unity, allowing us to program our own logic on contact.

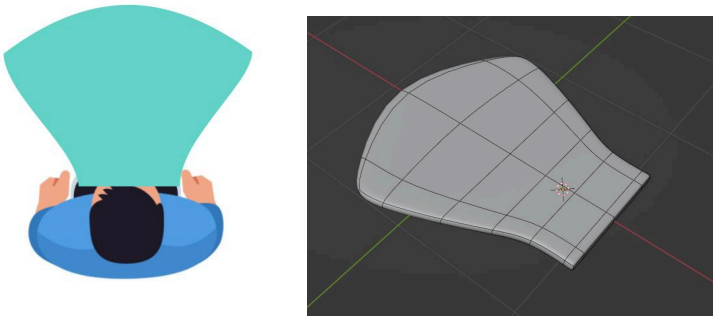


Figure 4.17: The stopping mesh

Actual stopping distance

For the sake of simplicity, we can just assume that the shape of both of these areas are roughly the same for both the human and the robot, but that it is mainly the distance that will vary. For the stopping distance of the serving robot, the Industry Safety Standards (ANSI/ISO, 2014) dictate that this must be at least 0.5 meters in this context. Just to be sure however, the robot distance is set to stop at 1 meter based on own experiences.

The humans however are more tricky to find sources for, but based off observations and general intuition, the stopping distance was set to 0.8 meters, as humans are generally more confident in their ability to move around obstacles.

Programming the collision

In order to define an actual collision in the simulation, we can say that any part of the collision mesh coming into contact at any part of the human or robot, depending on the agent's identity, will result in a full stop until the target it out of the way. To identify which meshes we have to look out for, we create a script that scans all humans/robots in the scene, and saves their meshes in a private list.

Condition to stop:
Mesh makes contact with the stopping mesh and is in the list of collision object.

Condition to resume:
Perform a check if this mesh has left the area since the last check.

After the initial round of testing, things went well. However, after repeated tests, some problems quickly arose:

- When two of the agents enter the stopping zone, it only checks if the last entered mesh has entered the stopping zone. This was quite rare, but had a big impact when it happened.
- Conflict 1 (direct confrontation) got both of the agents in an infinite loop, as both agents are waiting for the other agent to leave.

To solve the first problem, a second private list was tracked for every mesh that entered the stopping zone, which the agent can iterate over every check. The second problem required it's own research.

Resolving the infinite waiting dilemma (A.K.A the sidewalk tango)

As both parties cannot stand there for an infinite amount of time, one of them has to move. If we look at our priorities, we have to make a choice between making the robot move first, which would mean less time taken to deliver an order, or letting the human go first, and being less of an obstacle on their route.

As the humans interests in their pathfinding is more important than our delivery times, it is therefore important that the robot stays on standby until the human moves. As humans get irritated over the time of X seconds, we can assume that the human loses their faith in the robot's ability to solve this conflict, and enters a state where they take their own initiative and ignore any robots in their way for X seconds. After careful tweaking, the final waiting time seemed most realistic at 3 seconds, and the robot ignore time at 5 seconds.

Condition to stop:
Mesh makes contact with the stopping mesh and is in the list of collision object.

Condition to resume:
Perform a check if all meshes in the "entered" list have left the area after the last check
when human:
If the waiting timer reaches 3 seconds, resume walking and ignore all robots for 5 seconds.

4.7 Avoiding “highway behaviour” by implementing a steering assistant.

Avoiding “highways” and perfect alignment
After the initial testing was done, it was found that the humans were too simple and kept running into the robots despite having all the room in the world to adjust their course. This was because the pathfinding for both agents sends them on the shortest path possible. This means that turning a corner aligns them to a certain axis.

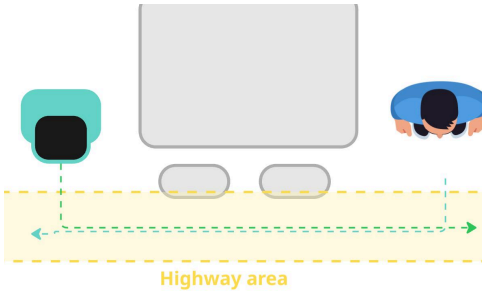
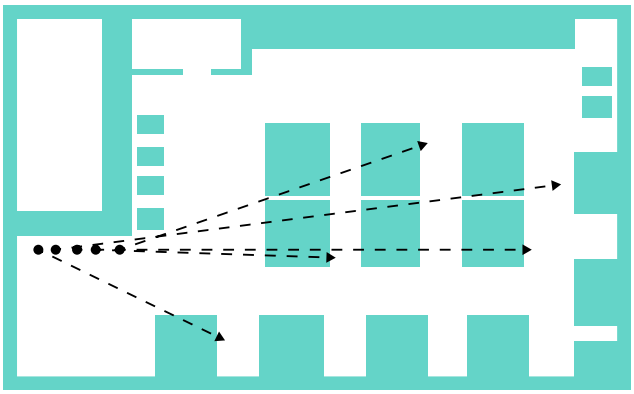


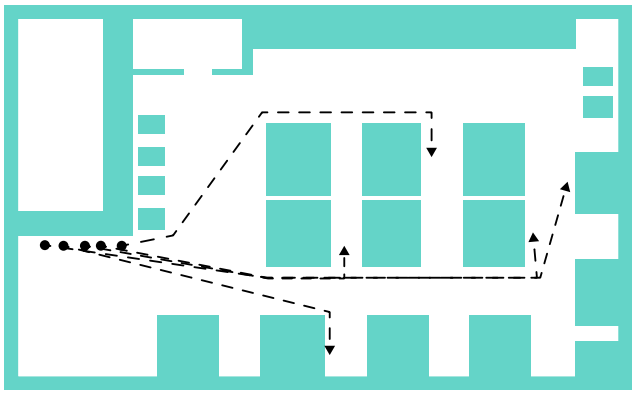
Figure 4.18: highway areas

This created essentially “highways”, which were horizontal or vertical lines that all agents seemed to follow. The original plan already involved the idea of creating a high traffic area in the lower main corridor. However, the amount of collisions that seemed unrealistic and easily avoidable inspired the change for a simple fix. On top of that, the frequency in such a small area essentially forced the user to focus on that spot, defeating the point of testing different visualisation methods.

Figure 4.19: pathfinding calculation



Initially assigned orders



Areas after pathfinding

Therefore, a system was needed to create a simple avoidance system for the humans that would eliminate the simplest collisions, while also preventing the hard collisions to be stopped as well. To create such a system, the decision was made to program a lazy steering assist, as is also seen in a lot of modern cars. It was therefore decided that the system should have the following requirements:

- The system must detect robots within x amount of meters in front of them
- The user must turn a near hit into a near miss
- A full on collision should not be prevented using the steer assist, otherwise it makes the simulation to easy.

Calculating the steer assist angle

In order to steer the human in a certain direction, we first have to calculate the amount of deflection, and which direction they are supposed to navigate to. It is possible to draw a plane directly from the front of the human. We can then calculate the amount of avoidance in degrees that is scaled as distance A and B get smaller.

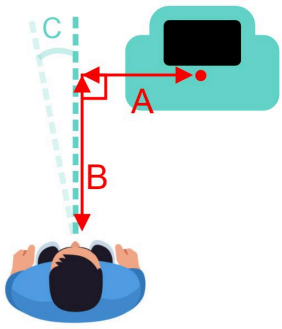


Figure 4.19: Factors in steering angle

This causes the human to deflect more and more as they get closer to the robot, and allow us to tweak these values to get the desired result. After some testing however, it was discovered that multiple robots threw off the steering system, especially whenever a human tried to walk between two robots. To fix this, a second layer was added that calculated:

$$\textbf{Ctotal} = \frac{\textbf{C1 + C2 + C3 ...}}{\textbf{Total amount of robots within range B}}$$

This resulted in a feedback system that steered the human between robots, until it found itself in an equilibrium. This meant that in these specific situations, the human would slide in between the two robots, and adjust their course as they get closer. It should be noted however, that this logic could cause two robots in close proximity at the same side would create slightly less Ctotal, thus resulting in the human steering away a bit less than usual. Quick tests proved this difference to be mostly invisible.

Avoiding steering humans into a corner

As the human and robot face each other head on with near mathematical perfection, there was a 50% chance of the human steering into the wall of the highway. After increasing the pathfinding radius of the human however, the distance to the highway defining wall was always greater than the robot, thus ensuring always steering away from the robot when the space to do so is available.

The old radius

The new radius (exaggerated)

Figure 4.20: Pathfinding clearing radius

4.8 Handling robot and human navigation

In order to simulate our restaurant environment, it is very important that the robots behave in a way that is appropriate for the context. For this project, we are assuming that **the environment is mapped out already**. (Intersys Limited, n.d.)There are plenty of papers out there discussing how this can be done, but for now, we must assume that we are using the same system that is already being used in restaurants.

Switching from custom logic to pathfinding

Earlier iterations of the simulation included scripts that had the robot traveling directly to the objective, and adjusting course whenever it bumps in to something (roomba logic). This however proved to be very unreliable, so the choice was made to switch to the A* search algoritm, which is already partially included in unity.

This system basically does the same as the roomba logic, but in a hypothetical tree pattern. After some testing, it was concluded that this sufficiently realistic for our scenario.

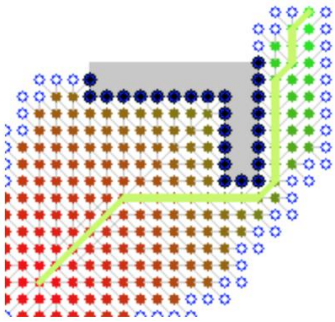


Figure 4.21: A* search algoritm

Calculating the NavMesh

One benefit of using a simulation, is that we already have a digital twin mapped out by simply copying the real world. It is now a matter of calculating the areas that are considered “accessible”. As we were already struggling with performance before, it was very important to keep this lightweight. By laying a grid over the restaurant, we can see where these points intersect with potential obstacles, thus eliminating them from the grid. The following resolution allowed for logical pathfinding while still keeping the amount of needed calculations to a minimum.

To automate this process for future researchers that might be working on this project in the future, the NavMesh calculates all obstacles that aren’t marked as a “non obstacle”. As all existing objects in the scene are already marked as such, any 3D mesh imported into the scene by researchers is therefore taken into consideration when the NavMesh is calculated again with the press of a button.

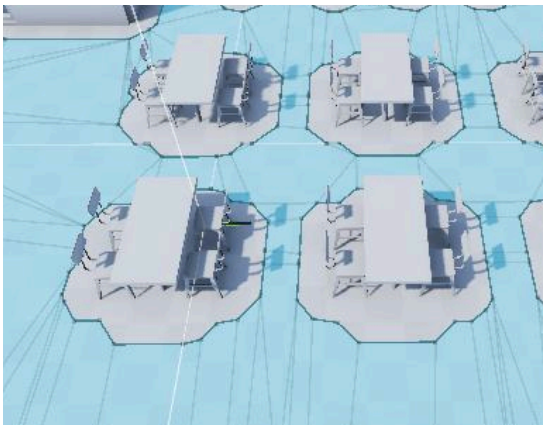


Figure 4.22: Pathfinding clearing radius

Finally, we could now give commands to both humans and robots to calculate routes around the navmesh. All that was left is to determine the “buffer zone” in which the navmesh agents take their corners. First iterations took the perfect most efficient route, but this brought them way too close to tables. The solution for this was to add a seperate safe zone around humans and robots so they keep their appropriate distance.

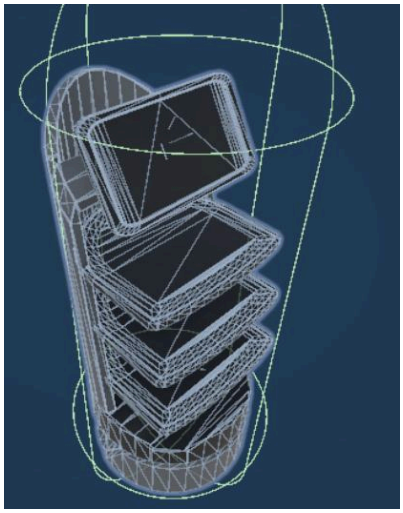


Figure 4.23: The actual collider of robot agents

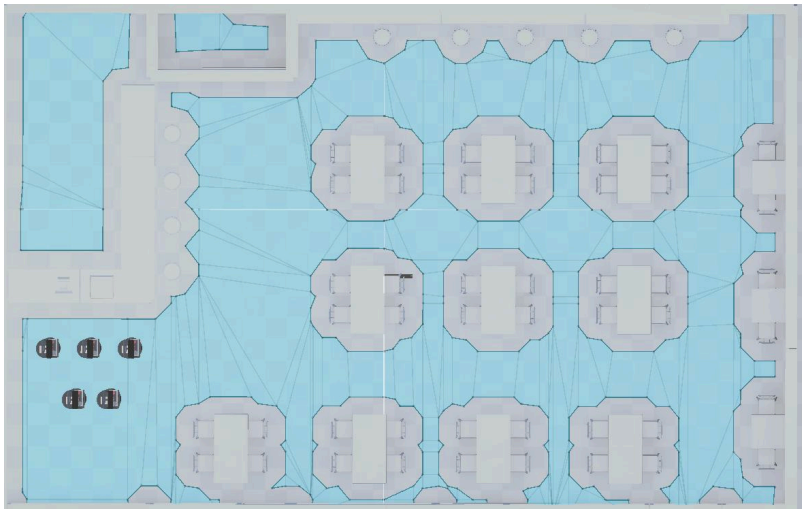


Figure 4.24: The default calculated NavMesh

4.9 Aesthetic design of the robot

General colour scheme
In order for the user to be able to distinguish the interactable robot over the background, it was possible to go for a distinction in both **form** and **colour**. In terms of colour, it was already decided that black was the colour channel that was reserved for the robot. However, in terms of form language, there is a clear conflict with the interactive element of the robot, and the selection part for the navigation.

Therefore, the choice was made to split the design up into a grey and completely black colour scheme. To make sure that this two part colour scheme was visible from all angles, a round base was added to the robot, which means that even if the darker trays and screen were pointing away from the camera, there would still be a clear visual distinction between the two colours. Another motivation for this separation in colour was the fact that it would make it easier for food to be visible on the robot due to the increase in contrast.

Communicating the order status
In order to show the user the current status of the delivery, some form of notification was needed to make a distinction between a delivering robot and a returning robot. While this was initially done via a UI marker, it was later changed to a dinner tray with neutral colours. This allowed the user to recognise the status from almost every side, while still minimising the amount of visual stimuli.

Figure 4.25: Dinner tray model

Animating a face
In the development process, the option to display the robot's status with an animated face was also used. This would allow the robot to relay certain statuses such as pathfinding trouble or perceived risk. However, due to the distracting nature of the face, the use of blue colour to visualise this, and the fact that the display cannot be seen from every angle, this approach was eventually dropped in favour of the dedicated visualisation overlays.

Figure 4.26: Earlier iterations of the faceplate

Result and evaluation

After the inital design was done, it was assessed that the current black and grey design allowed enough contrast seperation from the background to make it easy for the user to spot the robot from a distance. It was also verified that the black colour does indeed improve the contrast and readability of the robot's order status.

Lastly, the twin colour scheme means that there is still some semblance of black left, even when the robot turns away from the camera. The yellow/orange visualisation method elements are also visible on both colours, as can be seen in figure 4.27.

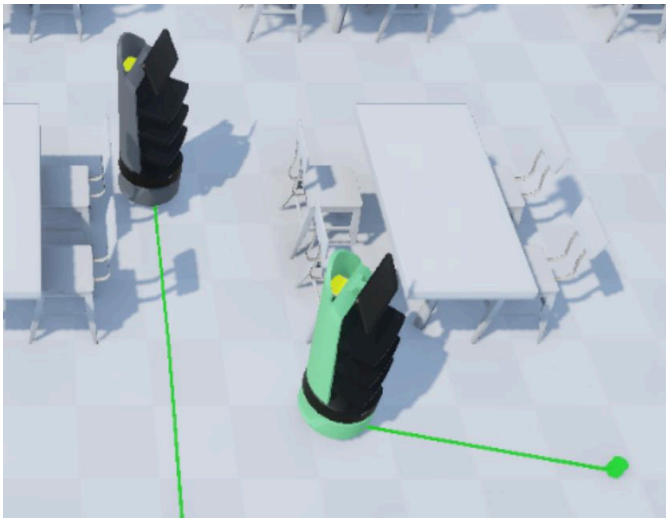


Figure 4.27: green and grey colour scheme comparison

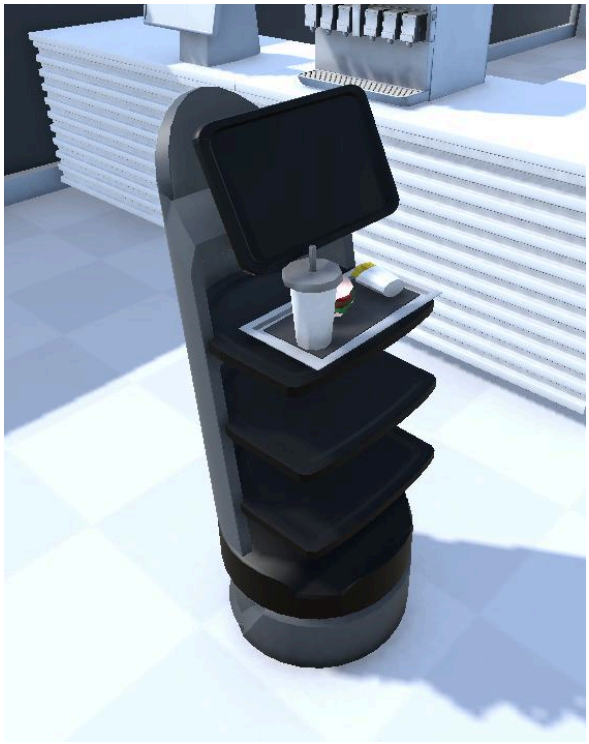


Figure 4.28: final robot design

4.10 Point and click reroute system

As the original goal of the movement methods involved a wide array of tools with a different effort to effect ratio, the point and click system would naturally fall within the spot of costing the least amount of effort.

Main concept
As the robots are almost always moving to a predetermined destination, a newly placed waypoint would merely be a reroute. If the user wants the robot to go around the table on the other side, they would simply have to place a waypoint on the opposite side, and let the pathfinding adjust the course. An example of this behaviour can be seen in figure 4.29.

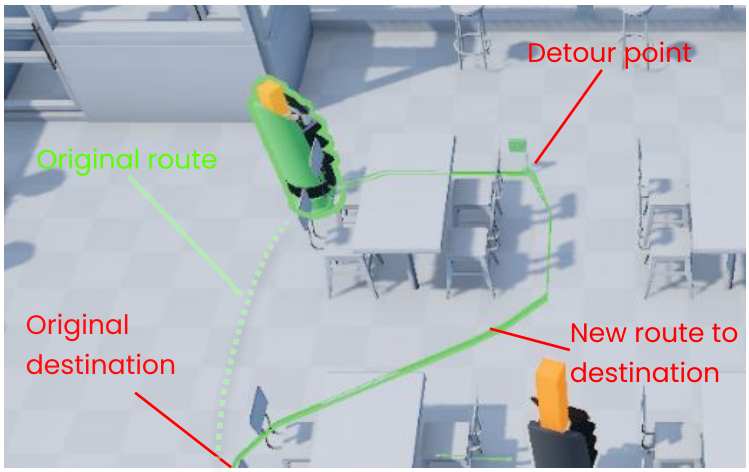


Figure 4.29: Point and click visualisation

Controls
The main goal was to develop this system to use no more than two mouse clicks, with one of them selecting the robot, and the other one for setting a waypoint. During the initial testing, it was quickly found that clicking a robot slightly outside of the hitbox would result in a direct movement order for the previous robot. To avoid the robot going to the wrong areas to to a mis click, the choice was made to bind the button to set a new waypoint to the right mouse button, which is not currently bound to anything.

While this took some time for new users to get used to, it was ultimately much more reliable than the original system. Most notably, some users who had experience with playing real-time strategy video games, required almost no explanation, as this method of control is already the standard in that genre.

Optimising the system based on feedback
On the original version, the way to the new waypoint was marked in a dotted line. After some tests however, this was found too distracting, as making a distinction between the original route and a detour was not really valuable information.

The final feedback was the wish to be able to place more waypoints. This meant that the user would now be able to place multiple waypoints when holding the shift key, which is a general method used when selecting or placing multiple objects in games or design software. Placing one without shift will remove all old waypoints. Most users found this system to be satisfactory.

4.11 Designing an automated experiment and measurement system

Generating reference data
While a simulation was now created that generated a steady stream of infinite orders, routes, obstacles and collisions, the initial plan was to let the simulation run for a set amount of time without any human interference. While the random assigned orders could mean that every simulation was somewhat different, the hypothesis was that over a period of about 20 minutes, the average amount of stops (which is the main metric of success) would be roughly the same.

To measure this reference data, a program was made that let the simulation run for 20 minutes. In order to speed up the process, the simulation was set to 10x speed. It would simply count the total amount of stops by the simulated human agents, save that value, and restart the simulation. This was done 10 times.

Run	1	2	3	4	5	6	7	8	9	10
Stops	23	12	34	28	18	36	2	18	25	26

Figure 4.30: results from the random orders pilot test

This resulted in a **mean of 24.8 stops**, with a **standard deviation of 7.39 stops**.

While it could technically still be attempted to run tests with this amount of noise in the data, it was decided that exploring alternatives would be a much more viable option to get proper results.

Creating consistency in the experiment rounds

After the failure of the former system, it was decided that a new system should virtually be the same across every experiment. Therefore, it was decided that the user has to complete a predetermined sequence of orders, while the humans walk in their predetermined paths.

This meant that it was necessary to take some autonomity away from the human and robot agent programming, in favour of a hive mind system. To automate and streamline the process, the system functioned as a central nervous system that sent and received signals between itself, the robots agents and the human agents. This system was called the WaveManager.

Switching to a general wave manager

In practice, it would read through an excel file line by line. Every line has a specific order for a table. The WaveManager then assigns an unassigned robot to that table, and starts a timer. After that robot returns, it simply logs the time taken for that order back in, and assigns that specific robot the next order. If a robot runs into a human along the way, it sends a signal to the robot, which then sends a signal to the WaveManager. This marks one human stop for that specific order. The WaveManager then continues to distribute orders, until it has no more orders left. It then writes the last line and closes the simulation automatically.

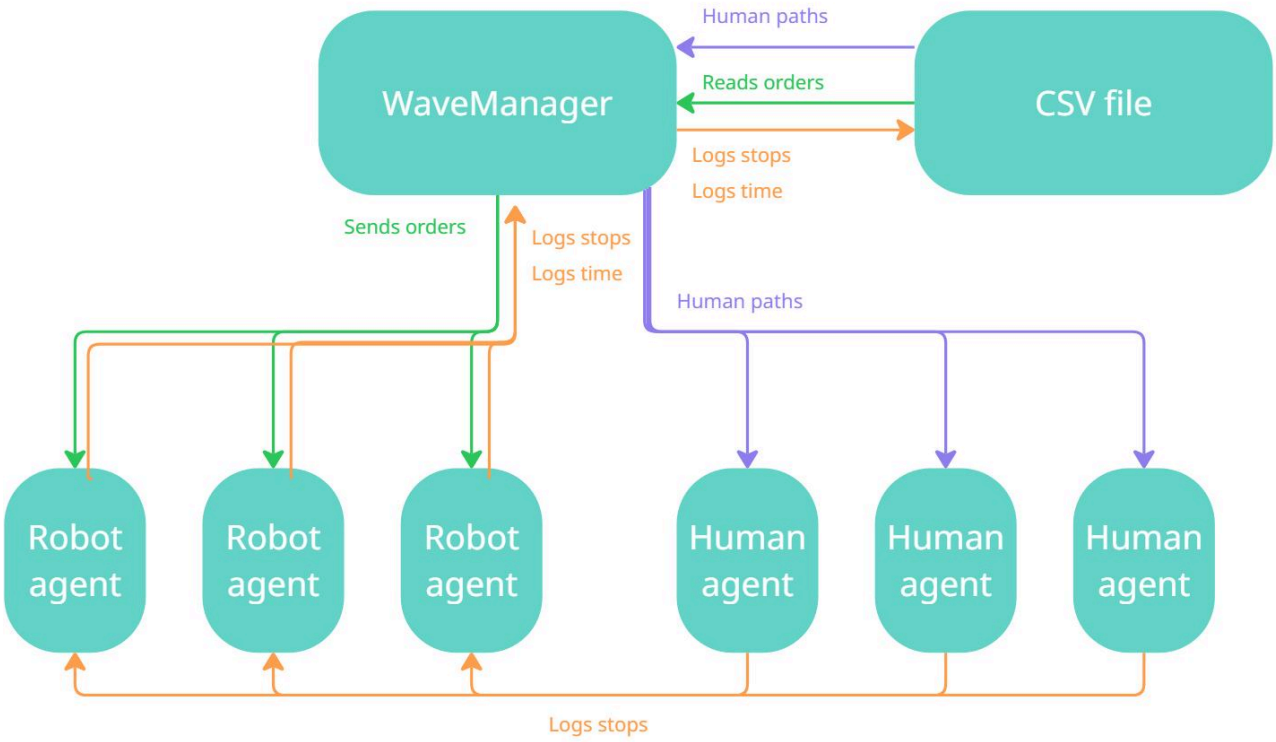


Figure 4.31: data flow between the WaveManager and the agents

At the start of every simulation, the WaveManager also takes three points for every human in the excel file. These three points are then stored inside the human agent itself. This means that the amount of data traffic between the WaveManager and human agents is minimised in order to reduce the amount of data traffic between the WaveManager and Humans in order to keep future development as simple as possible.

Finally the WaveManager also counts which visualisation method was active while a stop happens. This could later be used to cross reference if a user might feel more in control with one specific method, but still performs worse.

Adding a human predictability factor

Seeing as the main goal is to allow the operator to add value using their human intuition, it is therefore important to create a system that makes humans walk in a matter that is predictable for human operators. For example, a robot would simply see a human walking in a random direction, while a human would be able to figure out that the person walking towards a specific corner is probably aiming to go to the bathroom, or their trajectory makes it clear they are heading towards the bar to get another drink.

The previous page mentioned three points. These three points are the following:

1. A home table for the human to return to
2. A logical human destination, which is either a bar, bathroom or exit
3. A detour point in the middle that allows for more control on the path they will take.

The human is already capable of calculating their own paths towards the destination. However, as their chosen path is locked to be the shortest one possible, choosing a middle point allows for more control on the exact path taken, which is used to prevent highway behaviour, and allows situations to be created based on the situations discussed in chapter 4.2.

Figure 4.32 shows the available detour points, destination points, and home table points. After a human reaches their destination or their home table, it waits for 4 seconds before moving on in order to prevent any sudden movements that the operator might perceive as unexpected.

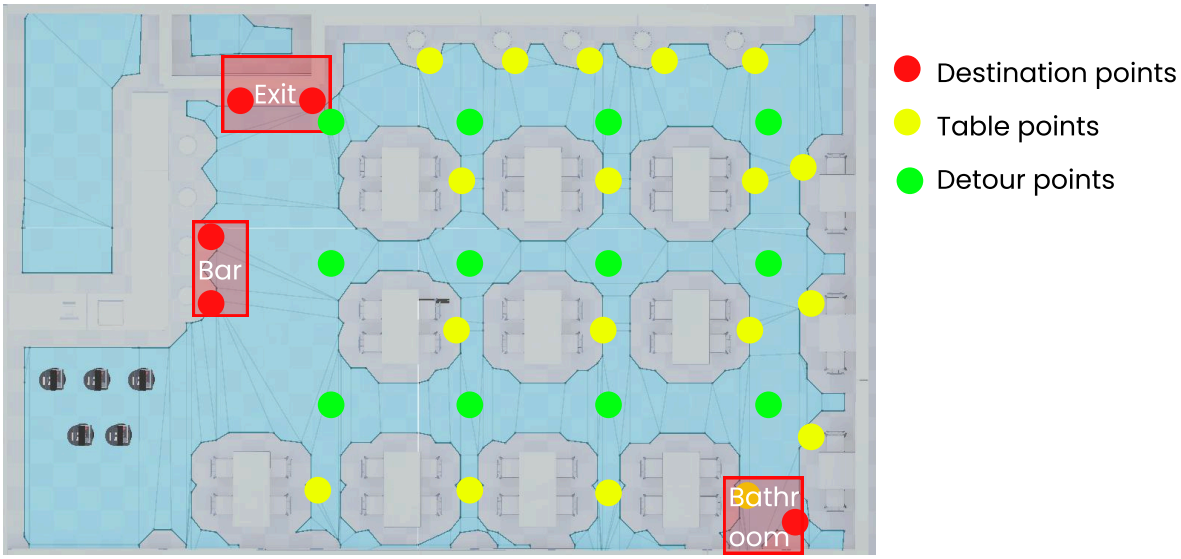


Figure 4.32: a map of all human home table, destination and detour points

05

Pilot experiment and assessment for phase 1

5.1 Goal of the pilot experiment

In order to verify the effectiveness of the human assisted control, it is important to first test the whole system and simulation with actual participants. This would bring the following benefits:

- It identifies where users experience friction with different control strategies and visualisation methods, already eliminating the ineffective options at an earlier stage

- It reveals patterns in how users manage attention and control under dynamic, time-pressured conditions, which can guide the development of more intuitive oversight tools.

- It offers concrete data on when and why users choose to intervene

- It allows us to see a direct comparison between the original automatic pathfinding of the serving robots, versus the human assisted, semi-autonomous system that this thesis tries to demonstrate.

Ideally, three or more participants would be able to influence the simulation to the best of their abilities, which would then be followed by a qualitative survey. Doing such a survey can help isolate which specific parts contribute the most towards the increased performance caused by that added human intuition.

From those findings, it would then be possible to pick a specific research focus for the rest of the project, such as the visualisation methods, or the control strategies.

Using a direct comparison using the total time and stops would therefore give a quantifiable score. This score can then be used as a reference for future experiments to deduct their effectiveness, with the goal to avoid as many conflicts of movement as possible.

5.2 Experiment setup

As it has yet to be determined if this system actually adds value to a socially complex restaurant environment, the best way to test it is by comparing two specific situations. The current setup would essentially simulate the way serving robots already work, stopping when they run into humans, and humans stopping when they run into a robot.

Three users will place themselves behind the computer, and are first introduced to an isolated test environment, where they will be taught the basics of robot controls. This means that they will be instructed on how to access the various control methods, and how to use said control methods. They will also be instructed on the different visualisation methods, and are tasked to turn them on or off using the on-screen buttons at their own leisure. The users are then actively encouraged to try all movement and control methods in order to prevent as many human-robot collisions as possible.

Main metric: time and collisions
Collisions between humans and robots causes them to delay their route for a set amount of time. Therefore, the user is essentially graded based on their total time. This often takes about 4 to 8 seconds, depending on the type of conflict of movement.

Seeing as a slight delay in delivery times caused by a detour leads to a lot less customer dissatisfaction than a conflict of movement would cause, the users are tasked to take their extra time if they need it. In the case that the user's detours or manual robot navigation takes a lot more time, but still prevents a lot of movement conflicts, the raw amount of stops is also counted, to make sure that there is always a measurable metric to work from.

Procedure
With the new wave system in place, each wave takes about 2 minutes to complete. Therefore, the user is tasked to complete 10 waves of orders in total. This ensures that the user will eventually get tired and resorts to methods that might not perform as well, as others, but provide the highest effort to effect ratio.

During the test, users are also encouraged to think out loud, so that any remarks can be noted down for future feedback.

Finally, the user is interviewed on their opinions of every control strategy and visualisation method, and will be asked about their perceived mental load, learning curve, and general effectiveness.

5.3 Hypothesis for the pilot test

In order to measure the success of this specific test, a scale will be set up in advance. Now that the test setup uses a predetermined order wave system, the simulation is first done without any humans at all, which would be considered the “perfect” execution, as all robots calculate the fastest route possible.

The opposing side of the spectrum would feature the current situation, which is the waves being run in the simulation with the robots on autopilot without any human interference. This means that robots would essentially continue their path until they collide with humans. This would essentially be the “worst” scenario.

Figure 5.1: the scale used for success measurement

Running the test with three participants would then place them somewhere on this scale, so the final results would look something like this:

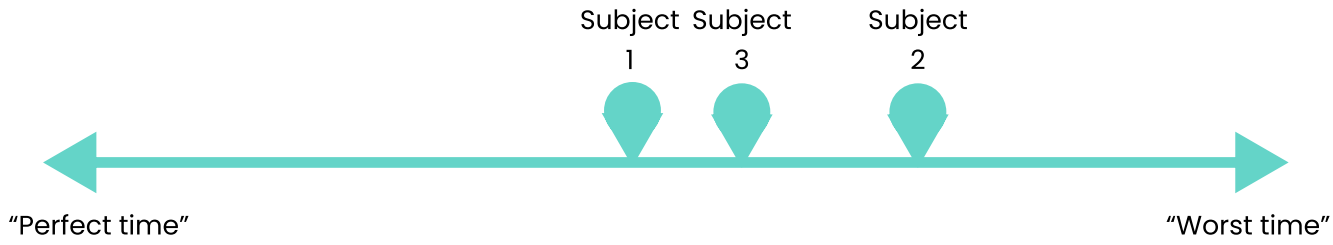


Figure 5.2: hypothesised results

This way, future feedback can be used to both group these results closer together, while also working to get results to the left side as much as possible with every new iteration of the system.

5.4 Results of the first pilot experiment

With all these systems in place, it would now be possible to create one set of reference data, and compare that to the amount of stops and time taken from the user data. It would therefore be possible to see how much better the human performs compared to the automatic pathfinding system that robots would traditionally take.

As the new wave system was consistent across all runs, the computer only needed to generate one instance of reference data. This gave a reference data of 24 stops in total. With this data, it would be possible to grade the human performance on the percentage of stops compared to the total.

Test results
After the first batch of test results, it became apparent that this method of measurement was inherently flawed, as results stayed mostly the same, and in one case became even worse. While humans performing worse than machines could be considered a logical explanation, manual observation during the tests clearly showed the users preventing situations that otherwise would have resulted in a stop.

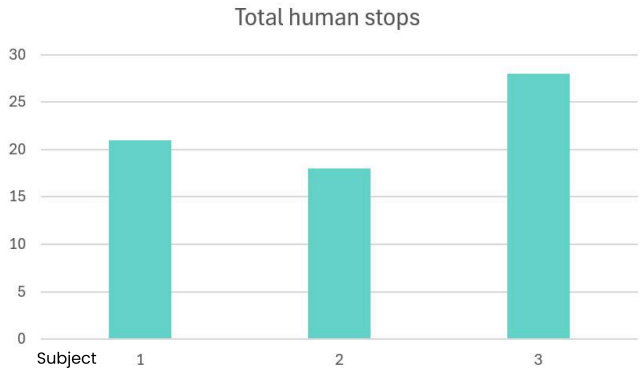


Figure 5.3: the results of the first pilot experiment with three participants

Analysing anomalous data

While the sample size is rather low for a real experiment, the clear disconnection between the observed performance and actual results means that either the aforementioned metric does not represent success, or that the way the metric is measured is somehow blurred.

In order to test the latter theory, the reference data was generated again, which got a standard deviation of 4.22 from a mean of 23.4. While this could be considered a somewhat decent level of noise in a traditional experiment, such a high variability in data that should theoretically be 100% consistent, shows that something went wrong.

The main reason for this divergence in results will be discussed at a later stage in this report.

Other feedback from the pilot experiment

While a formal questionnaire was not prepared for a pilot test of this scale, a qualitative interview was conducted for the three people participating in the pilot test.

General user experience

- The combination of control strategies and visualisation methods was considered to be a bit overwhelming
- The low visibility range of robots gave users the idea that the human agents were created, and that they disappeared when not observed
- The overall experience was considered to be too dull whenever robots were not near humans, and too chaotic when a conflict of movement was imminent.
- The users sometimes had trouble seeing robots when they were partially obscured by an object such as a chair or a table

Control strategies

- The parking camera was considered to be basically useless
- The point and click system sometimes interfered when an object such as a chair was in front of the ground the user intended to click on.
- The third person camera was described to be the most natural by the two participants that had experience with such controls being used in video games. The one without game experience preferred the first person camera.

Visualisation methods

- Most users felt like the LIDAR dots were more reliable.
- The factors that cause the risk bar to rise were not properly understood by most users.
- All three users were seen enabling every single visualisation method at the start, and had not bothered to change it at all.

Conclusions based on the interview

Based on the fact that all the users were overwhelmed to some degree, it was decided that having both the control strategies and visualisation methods freely is simply too much for a single experiment. Because users turned on all the visualisation methods at once, allowing them to freely toggle these methods would be a bad idea for future experiments.

In order to increase user friendliness, it would also be preferable to make the robots stand out more when partially obscured, preferably with some kind of outline system.

Finally, creating a system that allows users to spot potential conflicts further in advance would allow for an experiment environment where the difference in human intuition would affect the final results better. This could easily be achieved by increasing the maximum distance of every visualisation method available, while putting more focus on the scaling intensity of the visual stimuli of the warnings that come with it.

5.5 Observations of phase 1

Stopping the randomisation in data

It has now become clear that the current system in place does not work like expected, seeing as the reference data that should theoretically be completely consistent, shows a wild degree of variability. The main suspected reason for this could be the so called butterfly effect, which causes parallel situations to desynchronise after a small event earlier in the timeline. This causes the unintended behaviour of punishing the user by successfully avoiding a situation because it puts them in a more complicated conflict they otherwise would have missed.

The main solution would be to set the amount of orders that are distributed to one per robot, and wait for all orders to be completed. This comes with the drawback of creating a period of relative downtime caused by a part of the robots having already completed their order, and the other ones still being on their way back through an area that has little to no human traffic. However, the constant re-synchronisation of human and robot paths does mean that the butterfly effect can be avoided to some degree.

The binary stopping metric does not reward optimisation

One problem that was also being observed was the fact that users often did their best to prevent a frontal confrontation or T-bone on robot conflict by about 95%, just for the robot to still touch the outer edge of the stopping area, thus rendering all their effort in vain. Realistically, the human would be far more comfortable if a robot entered their outer stopping zone rather than standing right in front of their path.

Likewise, sometimes the robot and human were having a near miss originally, yet the operator still successfully redirected the robot to take a path that was about just as quick, but avoided the near miss confrontation completely. This would even increase the total time taken for that order, which actually negatively influences the final score. This especially often happened because of the steering assist feature that was implemented.

Finding a way to make a more linear system over a binary one would more accurately measure the rate of success at which the human operator redirects the robot paths. Removing the auto steering feature would also put all responsibility of avoiding a confrontation to the human operator, instead of creating uncertainty on that specific front.

Ideally, a unit that combines both the **distance** and the **time** as a combined metric would represent the total success rate with a higher rate of accuracy than a mere binary stopping system would.

Shifting the research direction to focus on visualisation methods

As the original pilot experiment made apparent, test subjects were found to be unable to properly divide their attention between identifying a potential conflict of movement with the visualisation options, and solving one using the control strategies. It is therefore important to pick only one of the two to focus the research on.

The goal of this research is to essentially see how human assessment can add value in ways that a robot cannot. Robots are already quite good at navigating around obstacles, which comes down to simple mathematical optimisation. While a human can add some value in that obstacle navigation by being able to better understand the human's movement whilst doing so, the real value lies in the way humans can fill in the blanks in very limited sensory data. Therefore, focussing solely on the visualisation methods is considered the best research direction.

The current setup fails to find the option with the right effort versus effect ratio

As was previously discussed, it would be a common pitfall to focus on the most effective solution without considering the actual user using said option for extended periods of time. With the current test setup, the main expectation was that as time goes on, test subjects would eventually become lazy, and stop working at their highest capacity. It would be after this point that the user would eventually settle for options that allows them to remain as much functionality as possible with the least amount of mental effort.

While it is true that eventually, test subjects grew mentally exhausted by doing the test, their tiredness often just resulted in a direct decline of performance. Despite that, performance also seemed to be much higher than expected as well, which does not feel representative for a worker that would realistically be distracted by all kinds of environmental stimuli and has been doing this for hours at the time. In order to combat the user always giving their maximum effort in the test, they must therefore be distracted by some kind of external task. It is important that this task is to be a "cognitive effort sink", meaning that the task should never truly be finished, like tasking them to build something physically next to the test, but rather something complicated with no end, like solving as many math questions as possible, or having the test subject count backwards from 1000 as fast as their are capable of.

This would also come with the added benefit of being quantifiable in itself. This means that even if the test subject decides to shift their priority towards the side task more than the main task (or vice versa), the total sum of both results should still be different for every visualisation method.

The butterfly effect makes data recording unreliable

As was observed with the generation of reference data, even with the new system that should theoretically give the exact same results every time, now suddenly gives seemingly random outcomes. This generates a volume of noise that effectively renders all test results almost useless. This could be caused by a variety of factors, which include, but are not limited to:

- Circular colliders with a perfect head-to-head collision may deflect in different directions, much like firing two pointy arrows directly at each other would practically always deflect in a random direction.
- Robots may be taking longer routes based on their location when the orders are distributed.
- Slower framerates might cause some collisions to be timed just in between two frames, thus sometimes registering a collision and sometimes not. This effect would be aplified by running the simulation at an increased speed.
- Both human and robot agents can eventually get stuck, which seems to have a semi randomised time it takes for them to “glitch” their way out of their stuck state.

One alternative could be to revert back to the old random orders system, and just increasing the sample size by an incredible amount. Based on the data from the pre-pilot test, that would take roughly an hour of continues simulation control, which unfortunately does not seem doable for the current participants, given their exhaustion after a 20 minute session with smaller breaks as well. Furthermore, adding the extra assigned distraction tasks would likely decrease the time untill total exhaustion as well.

06

Design choices for phase 2

6.1 Switching to a non-binary measurement metric

Initial feedback

After this round of testing, it became apparent that the stopping system already in place was too harsh in it's binary nature. The main goal of the simulation was to push the user to control the chaos to the best of their abilities. During the observation process, it was found that luck seemed to be a very big factor because of this. The user was observed to make very good calls and redirected the robot to better directions than the original course of the robot, yet often got caught within the edge of a stopping zone. Likewise, whenever the human and the robot were about to have a near miss, which was avoided altogether, the test results would still indicate a miss regardless, thus not recording any evidence of the user's contribution to the robot paths.

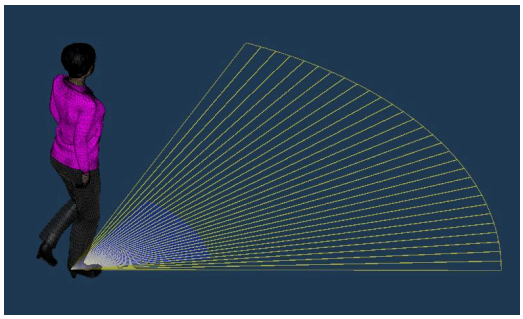
Demands for a new system

In order to create a system where optimisation and human avoidance in general is rewarded, we must make a system that meets the following demands:

- The system must make a proportionally scaling counter that makes a distinction based on human performance. No score should be perfect, as the goal is to manage as good as possible.
- The system must use a combination of time and distance to determine the total obstruction score.

Replacing the instant stop system

As a significant amount of time and resources were already invested in the LIDAR simulator for the robot visualisation system, it was possible to use this system yet again. In this specific iteration, the choice was made to go for a simplified cone form, as the exact points of contact were mainly focussed on the outer ends, away from the human. As the calculations for this method are relatively simple, the choice was made to put 32 raycasts away from the human.



Using this system, we can now use these raycasts to see which ones are hitting a robot's mesh. We can then group all the raycasts that "hit" per robot, after which it is just a matter of finding the shortest distance. The benefit of this method is that we can easily keep track of multiple robots in the field, while also keeping the amount of calculations to a minimum.

It could be argued that measuring the distance to the robot's centre of mass could be more accurate and consistent with the logic of other scripts, but using the raycasts that only detect the robot's mesh reduces the amount calculations by a significant amount.

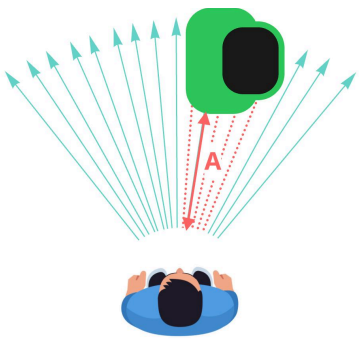


Figure 6.1: the shortest raycast

Adding up the score

Now that the distance A is available for our calculations, it is still important to create a general formula to calculate the total obstruction score. As performance was gradually becoming more and more of an issue, the decision was made to perform these calculations in "ticks" which are events that are triggered ever X amount of time, instead of running every frame, like a typical script would.

Weighted lines

If we review the assumptions that we made during the creation of the original binary stopping system, we can still assume that objects to the sides are considered to be less obstructive than those that are right in front of them.

With that logic in mind, it is possible to assign every line with a specific weight. The lines on the centre have a weight of 1.0, which gradually decreases to 0.5 for the outer lines.

This ensures that tiny amounts of obstruction can add up over time, just like they would in real life. With both of these values combined,

Obstruction score = distanceA * lineWeight

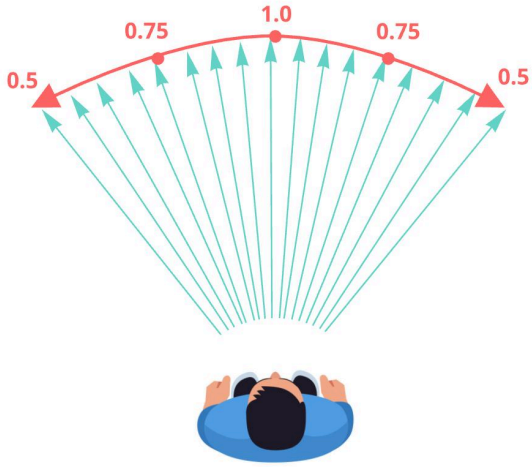


Figure 6.2: weight of all the raycasts

Evaluation

During the testing process, the system was evaluated based on expected outcome for the four different conflict of movement archetypes that were discussed on chapter 4.6. After careful testing, it was found that the direct confrontation (1) and T-bone on Robot (4) gave more or less the same with a score of about 4.5 to 6. T-bone on human, returned an expected score of 0.

After that, the overtake conflict (2) score was based on the amount of time spent behind the robot, but eventually ended up with a score between 2.5 and 5, depending on the alignment between the robot and the human.

Having to slow down behind a robot for 6 seconds could be considered as annoying as having to make a full stop, so it was decided that the current implementation of the scoring system was considered adequate and in line with the expected amount of frustration by these conflicts of movement.

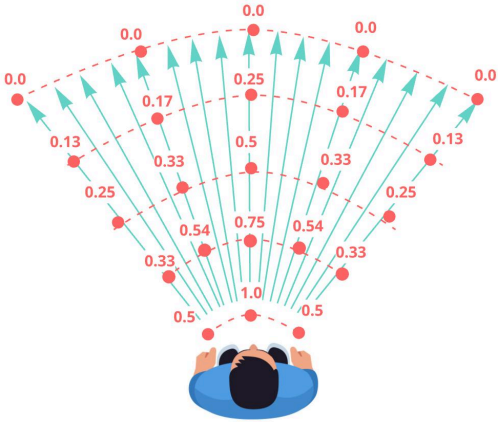


Figure 6.3: amount of obstruction score per second

6.2 Creating a new tutorial

As the offset in the results was also because people still had to take some time learning the basics of the simulation and the way they are supposed to control it. Hence the decision was made to expand the initial tutorial from a simple in-person explanation to a streamlined process. The main goal of the tutorial was to:

- Provide an environment in which people can see the different visualisation methods without any external pressure.
- Provide the exact same explanation and practice time for everyone to offset any potential differences caused by poor explanation of the mechanics.
- Provide as much exercise as needed so that the test subject will be minimising the amount of mental effort on translating their intentions via the control mechanism.

The end result of this design process was a series of isolated scenarios that highlight the different visualisations with a text and image description that the test subject can read at their own pace, followed by a live demonstration that takes about 1 minute.

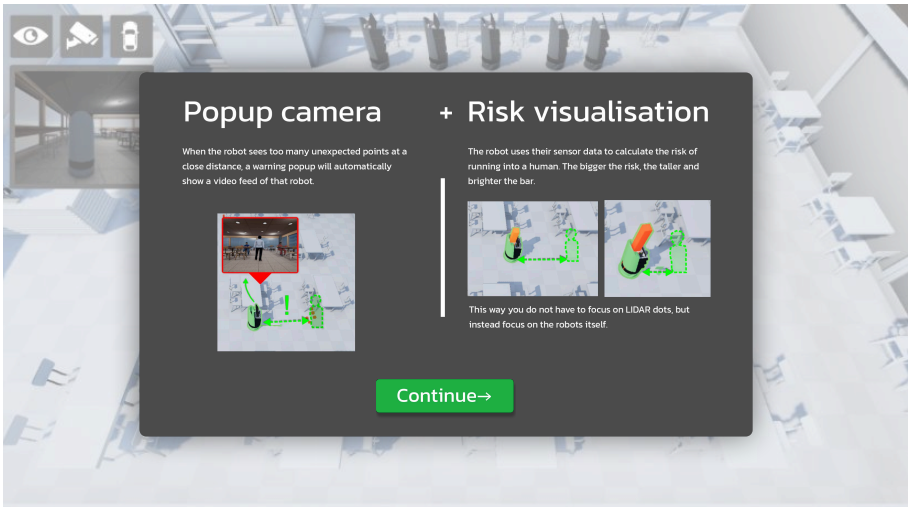


Figure 6.4: example of one of the tutorial screens

The first test subject expressed a desire to be able to shorten the demonstration time of one minute. While this was changed initially, the next person immediately started to skip the live demo after about 10 seconds. As the slight increase in pace was not worth the risk of potentially tampering with the results, it was decided to remove the feature to skip the demonstration alltogether

6.3 Tackling the butterfly effect problem

One possible solution for the larger difference in results relates to the so called “butterfly effect”, which is an effect noted by a small impact in the past having a large impact on the bigger picture. One might stop to tie their shoelace, which causes them to just miss their train, which would affect their whole day in return.

Because this experiment tries to control all variables by essentially resetting the state of the simulation to the absolute beginning every time, we already know the outcome. The drawback that comes from this is that this works under the assumption that there is one single event that the user is influencing.

In reality however, it was found that every scenario has at least a few encounters that happen in series. This is even the case after making sure that all the robots only take one order and travel back to their origin point. In some cases, this led for the results to be significantly worse. Preventing a single conflict of movement could sometimes mean that the robot arrives much earlier than they normally would, but this would then get the robot into a much trickier situation that otherwise would not have happened at all.

If we connect this to the analogy of missing your train because you got delayed, it would be that the original train you were visiting was going to get stuck in a defect, which your later train just went around using a detour.

Cherry picking the types of conflicts
This means that in general, the larger a timeline is, the more prone it is to the butterfly effect. As all the robots are already limited to one specific order per wave, and that the humans are resetting their path after every wave, we cannot make the timeline itself smaller. What we can do however, is reduce the amount of variables that could possibly affect each other to a minimum.

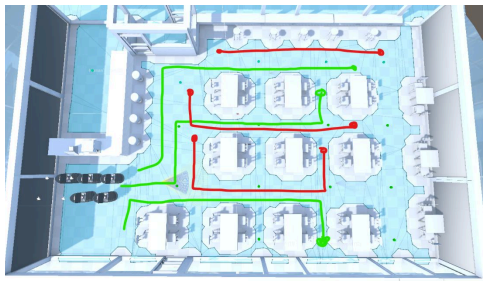


Figure 6.5: Hand written scenarios that do not interact with each other

The problem however, is that there is a very limited amount of choices available for this specific set of parameters. Keeping all the possible collision areas separated also means that redirecting the robot towards a certain path could also mean that they might find themselves interfering in another area. This means that we have to be really careful in the placement of the conflicts, which in turn, leads to too much control and the simulation to be too easy for the user to solve.

To find middle ground, most new scenarios were hand written to spread the conflicts out as much as possible, and also reduce the amount of robots from 5 to 4, which reduced the density of robots vs humans. This proved to be a decent middle ground, although this also let to the tests being perceived as significantly less interesting, as the amount of rerouting and user required input was significantly reduced.

Solving the perfect collision

One other possibility of this occurring could also be caused by presenting unity with a mathematical dilemma. In this case, that would be two circular colliders having a direct confrontation in a highway area. In a normal context, the circular shape would deflect them into the opposing direction, much like two balls in a game of pool would. However, when expressing highway behaviour, the circles are both facing a perfect collision. Theoretically, this would result in a full stop, but in unity, this results in a deflection to either side that seems to be completely random. After several failed solutions, the most consistent workaround was to create a custom script attached to each collider, that scales all dimensions to be between 99% and 101% of the original size. This functions the same way as the workaround mentioned on page 48. While this does not affect the performance of the simulation a lot, it does make the amount of perfect collisions almost 0.

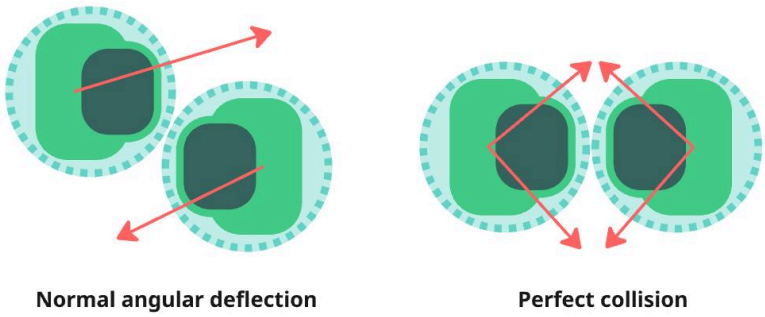


Figure 6.7: A comparison between two different deflections

Final assessment and conclusion

Unfortunately, one final collision that cannot be stopped is the diagonal collision, which happens when two actors meet each other on a corner. This is usually a T-bone collision on humans or robots that do not quite trigger a stop within the first two seconds. It was noted that these collisions often have different outcomes despite the same parameters.

The exact reason for this remains unclear, but is currently hypothesised that even small differences caused by user interference or even slight differences in framerate could affect the angle at which the circles are deflected away from each other, much like firing two arrows directly at each other will cause them to deflect in a completely randomised direction.

Solving inconsistencies caused by PC performance.

With the way simulations in engines such as Unity work, laws of physics are only executed when actually observed. As the simulation is only witnessed by the user in separated, refreshed frames on the monitor, it also makes sense to calculate physics between those frames. While this usually does not pose a problem, it does mean that the sample size at which physics and other movement calculations are done are tied to the framerate. This means that external processes, such as a different program updating in the background, could have an impact on the final results, especially if amplified using the butterfly effect.

While it is rare, it does mean that certain paths can get rounded down as the framerate slows down. Figure 6.8 shows an entity trying to move around a corner using physics calculations. In this situation, the entity tries to move along the projected path, and each frame it checks whether or not the entity is still blocked by that specific corner, but gets pushed back because it cannot intersect with the square, until it calculates a frame where it can.

Figure 6.8: the difference that a lower framerate can make

isolated speed test
In order to validate this, a series of isolated tests were conducted. One where the framerate was set at 60 fps, and one at 15 fps. It was observed that a robot moving a full route through the restaurant at a higher framerate was somewhat faster at the lower framerate, although this was a time difference of about 3-4%, with a standard deviation of about 1.5%. While this was a measurable difference, it was considered **unlikely that this would have had a major influence on the results.**

Impact of low FPS on the overall simulation
However, after running the simulation normally at an artificially lowered framerate, it was observed that the steering assist feature that was implemented for human agents seems to deflect itself further away from the robot agents on the higher framerates. While this should not be possible, as the script that calculates the deflection uses an internal deltaTime, it was decided that the added simulation aspect was not worth the risk of potentially disrupting the results, and was removed completely.

Other remarkable changes at a lower FPS also occurred not in the form of consistent changes, but the frequency at which agents could get stuck in a direct confrontation or overtaking conflict. This usually happened when two or more entities are stuck in a tight space. A lower framerate means that sometimes, agents can get pushed into a wall further between frames, which pushes them back with more force, resulting in what can turn into a classic case of "unstoppable object meets an unmovable obstacle" which is commonly described in video game physics engines as "glitching". This was actually observed to cause delays that can last seconds, which can very well impact the overall score if combined with the butterfly effect.

Unfortunately, even high-end video game physics engines are still often demonstrating this phenomenon, which means that this could be considered a hard technical limitation. While we cannot avoid these glitches entirely, increasing the frames per second can at least minimise the frequency of these situations.

Final optimisation
As a final test, the simulation was run once again with to actual limit on the FPS. In the resulting graph of FPS over time, it was observed that there were significant dips in performance, that sometimes shifted it from 70 to about 5 for a second or two. After enabling and disabling all the scripts, it was eventually discovered that a lot of scripts used a set interval to perform certain calculations. This was originally done to save as much performance as possible, but the pathfinding updater and WaveManager script did this at the exact same interval, thus creating a sudden spike in computer workload. This was then fixed by editing most of the major scripts to perform their calculations at a spread interval. The total FPS was then also limited at 25, as this seemed to be the framerate that the computer was able to keep at a stable pace without having certain calculations queue up.

Conclusion
With the steering assist for human agents removed, the scripts optimised to spread their workload, and the total FPS set to cap at 25, the simulation should now be a lot less influenced by varying performance. Glitching, while less frequent, is still something that occurs from time to time, and cannot be reduced more than this at this point.

6.4 Using automated math questions as a secondary research metric

During the pilot test, it was also discovered that user performance was surprisingly high for the duration of this session. The main leading reason for this was the relatively short length of this session. Due to which, the user puts in the maximum amount of mental effort available. In a real situation however, the user would eventually have trouble focussing their attention on the application after hours of intense use. As we are looking for the visualisation option with the right effort to effect ratio, it is therefore desirable to test the user's performance with limited cognitive capacity available.

To simulate the user not paying all their attention, a secondary task was given in the form of randomly generated math exercises. To do this, a separate application was made on a tablet. The main reason for using a tablet was to focus the user's attention physically away from the screen. The tablet also came with the benefit not having to pick up a pen every time the user thinks of their answer, which took a surprisingly higher amount of time than expected. To further ensure that the tablet draws as much attention away from the screen as possible, the case holding the tablet was physically taped to be around 50 cm from the main interface,

It could be argued that the randomness of the three digits could sometimes provide much easier questions to solve than otherwise, but seeing as the average score was about 14 questions per wave, the effect of any outliers could therefore be ignored.

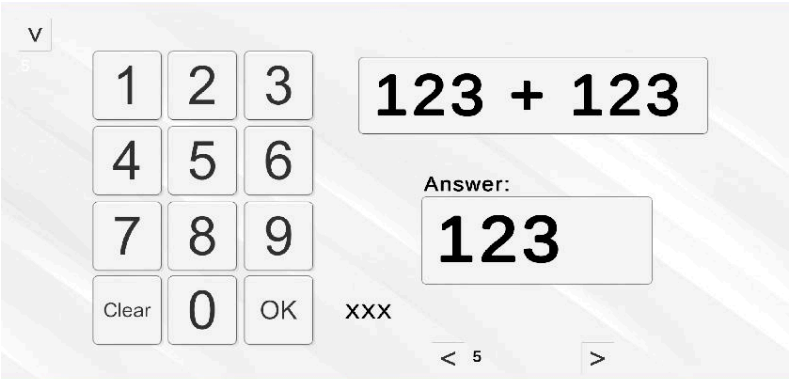


Figure 6.9: the UI of the math question setup

To automate the process even more, a simple data transfer was set to transfer all the data over a USB COM connection. This allowed the data to be sorted automatically, while also saving the timestamps of every entry. Saving the timestamps would not really be used straight away, but would instead be used to find out any potential outliers.

Testing and evaluation
After putting the system to the test, it was eventually found out that even slight movements in the USB cable could interrupt the connection, causing the system to fai. As an alternative, the user would receive prompts to manually change the round on the tablet between visualisation methods. This stored the data on the tablet locally and allowed for manual data entry after the experiment.

07

Main experiment and conclusion

7.1 Setting up the main experiment

As the conclusion from the first pilot test was that it would be best to focus on measuring the difference in performance of every visualisation method, it would therefore be desirable to isolate every one of those methods, and have users navigate their way through several waves of orders before switching them automatically. This means that the new experiment setup most fulfil the following criteria:

- The user must go through every visualisation method in roughly the same amount of time.
- All waves should have the exact same difficulty to prevent biased data.
- Each test subject must work through the visualisation methods in a different order to prevent the learning or fatigue affect disrupting the results.

While this would work for interchangeable visualisation methods, the main shortcoming of this method is that visualisation methods are meant to be combined with others, as some work better for longer distances, while others are more suitable to inspect a situation with more detail.

Switching to pre-sets
To still keep a clear separation of the variables that will be changed during this experiment, the users will be cycling between 4 different pre-set combinations that make the most sense from a usability perspective. These four presets include:

- LIDAR dots + popup camera
- LIDAR dots + corner camera
- Risk visualisation + popup camera
- Risk visualisation + corner camera

Adding a camera feed as a fifth preset
The goal of this experiment is to compare which preset of visualisation methods are considered to be the most informative, and offer the highest effort vs effect ratio. It should also be noted that it is important to prove that this alternative version of real world data is actually more readable than the real world itself. In order to validate this, a fifth preset is added that serves to simulate a direct camera feed that was placed in the room.

It should be acknowledged that while it would make sense to have direct camera access in a small restaurant, a bigger instance such as a hospital would have more trouble implementing such a system.

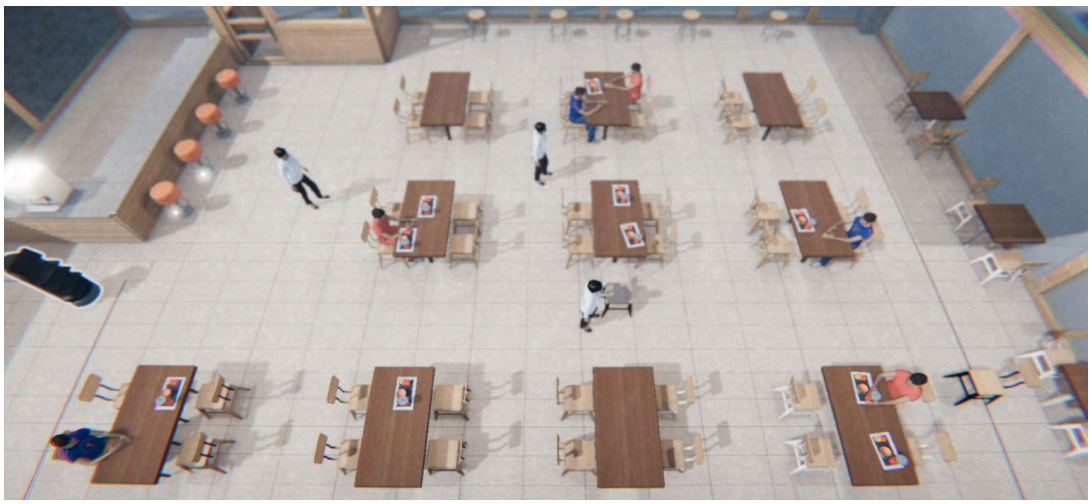


Figure 7.1: the direct camera feed

In order to prevent the direct camera feed from performing too well, some artificial filters are added to simulate the actual camera feed. This comes in the form of a fisheye effect, as the lens would need to be very wide in order to capture the whole room, chromatic aberration, to distort the colours based on cheap camera lenses, and finally some camera grain, which simulates the noise caused by low lighting and a relatively small bitrate. The camera feed also features more visual stimuli to further provide more contrast between the controlled and colour coded digital twin environment.

Automatic wave system

Finally, the UI buttons for switching between visualisation options were removed, in favour of a system that sets every preset for every wave. To make sure that every method is properly tested, the user will be given 5 waves of orders per visualisation method, which will result in a total of 25 waves. This takes around 7 minutes per wave, so the total time ends up at 35 minutes. This is admittedly rather long, but also needed in order to generate proper results. At the end of every wave, the user will be notified of their progress with a full screen message. This also funtions as a “palette cleanser” between the many rounds.

Figure 7.2: the sub-wave popup

Preventing the learning and fatigue effect using the latin square

As previously mentioned, test subjects are hypothesised to either perform worse or better due to the training and fatigue effect respectively, it is important to switch the results around to combat this. While this could traditionally be done by just randomising the order, the choice was made to shuffle the results around with the latin square method. This would ensure that all methods are tested in the same frequency. The WaveManager was therefore expanded to automatically shuffle the methods around based on the manually entered position of the latin square.

Measuring qualitative results using the NASA TLX

To get a clear picture of the user's opinion of the visualisation methods, a qualitative research is also done after the main experiment. This is done according to the NASA Task Load Index. (NASA, 1988) In order to prevent asking irrelevant questions, some parts that were removed from the form.

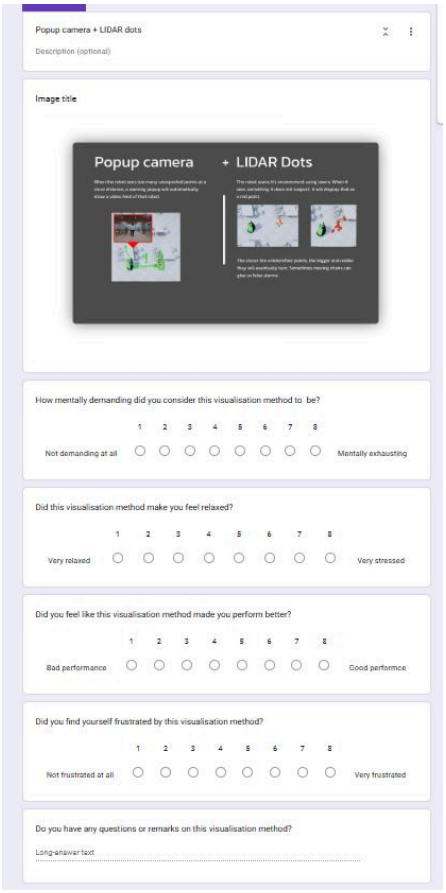


Figure 7.3: the format of the NASA TLX form

7.2 Procedure

For the final experiment, the user will be tasked to first take a quick tutorial on the different visualisation strategies, which provides them with hands-on experience, as well as a clear and standardised explanation on the pros and cons of every method. This can be done at their own pace.

After which, they will automatically start with the first wave. This means that the user is tasked to prevent as many conflicts of movement as possible, while also performing as many math questions as possible using the tablet at their side. Seeing as the connection sometimes gets interrupted, despite the fact that the tablet was taped to the table, the subject was also tasked to manually switch the rounds on the tablet when prompted.

This means that the entire project was now fully automated.

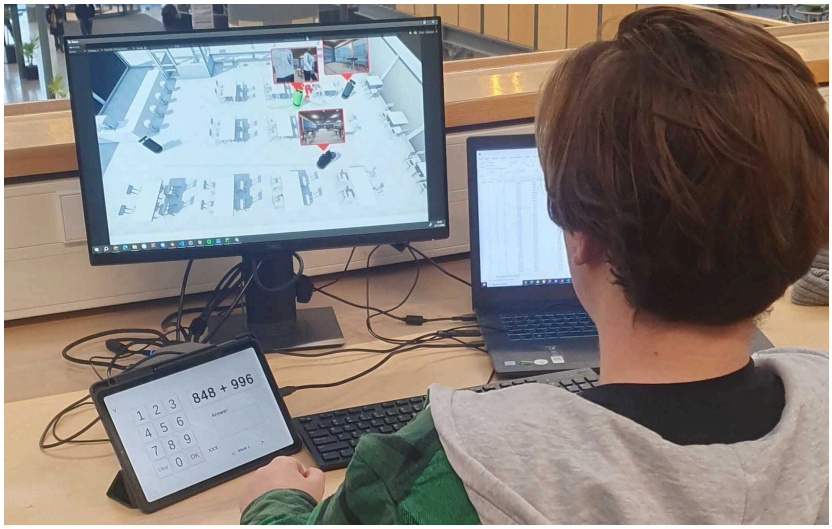


Figure 7.4: one of the participants running the experiment

Post research survey

After that, the user is presented with a popup that takes them to a google survey that asks about their experiences with each tool, which means that the whole process only requires one surveyor to set the experiment up, after which they are free to take notes or continue data analysis for the remaining 45 minutes.

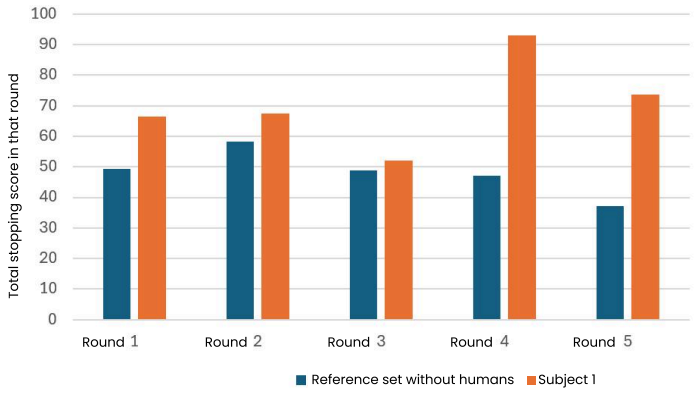
The goal is to reach 15 experiment runs, which should make sure that sufficient samples of the configurations of the latin square can be recorded.

7.3 Results

With the new procedure set in place, and two participants having done an early pilot test, the amount of waves per visualisation method was set from 6 to 5 according to feedback, after which the procedure was found to be satisfactory and ready for large scale testing.

However, when generating the reference data, there still seemed to be some inconsistency across the data, despite all the fixes that were implemented in phase 2. In order to determine the exact amount of noise, a test was conducted that ran one round of the exact same experiment 10 times in a row. This resulted in a mean human stopping score of **38.66, with a standard deviation of 6.65.**

Upon analysis of the first two pilot participants, it was discovered that the mean stopping score of them was **68.53, with a standard deviation of 22.99**. While this cannot be traced back in the data in hindsight, most of these higher scores were likely caused by observed moments of panic, in which the subject found one of the robot agents stuck in a collision with a human agent, and tried to get them loose with custom movement orders, thus constantly triggering and increasing the stopping score.



Conclusion based on the data

While it would be easy to simply draw the conclusion that humans are simply worse when it comes to making decisions based on these visualisation methods, it was personally observed that the human operators were capable of spotting a potential conflict of movement from a distance. Due to clear limits in the available time for this project however, a proper test to prove this with measurable data would not be possible.

Based on the limited user feedback however, it was found that both of them seemed to prefer the LIDAR dots + corner camera method the most in almost every part of the NASA TLX form. It should be noted however that personal observation did not seem to reveal any apparent increase in performance over other visualisation methods. The results of which can be found in the appendix at page 101.

7.4 Creating an isolated environment to study the path desynchronization

Experiment setup

In order to get a closer look on the exact nature of the constant desynchronization that seems to affect the results of the experiment, a new scene was created to act as an isolated setup. This way, it would be possible to have a controlled environment in which it is possible to change certain variables such as the framerate, without having to worry about any of the external factors. This also means that other scripts that are usually computationally demanding to be temporarily turned off, as the main focus is to determine the exact source of the desynchronization.

As a way of comparing results easily over time, the WaveManager has largely been reduced to only send out the orders each round. Another component has also been made, called the PathManager. The PathManager tracks every agent, and logs their location every 0.25 seconds. After that, it connects the new point to the previous point, which eventually results in a line tracing the full path of every agent over the course of a round. This system was found to barely impact the performance at all, which means minimal interference with the results.

The WaveManager then runs the same round of orders 15 times. after which it stops automatically as soon as all orders are completed. As the sample size is set to 15 in the simulation itself, it is then simply a matter of manually taking a screenshot and comparing the results.

Initial observations

With the first test being one where the parameters reflect the situation that was used in the final experiment, it became apparent that there was a lot more deviation from the original path than expected. The current setup shows the paths of the human agents in green, and the paths of the robot agents in red.

From a distance, it is visible that the humans tend to have a lot more deviance in their paths compared to their robot counterparts. While they share the exact same navmesh as the robots, their difference probably comes from their increased speed, which leads to bigger differences near the end of the sample, simply because they have passed more obstacles that potentially distort their course somewhat. Different speeds will be tested to determine the impact it makes on deviance. The slightly larger collision area that the human agents have could also potentially affect the way the agents take their corners. Seeing as this change was initially implemented to assist in the steering assist feature that has since been removed, it should be safe to experiment with that as well.

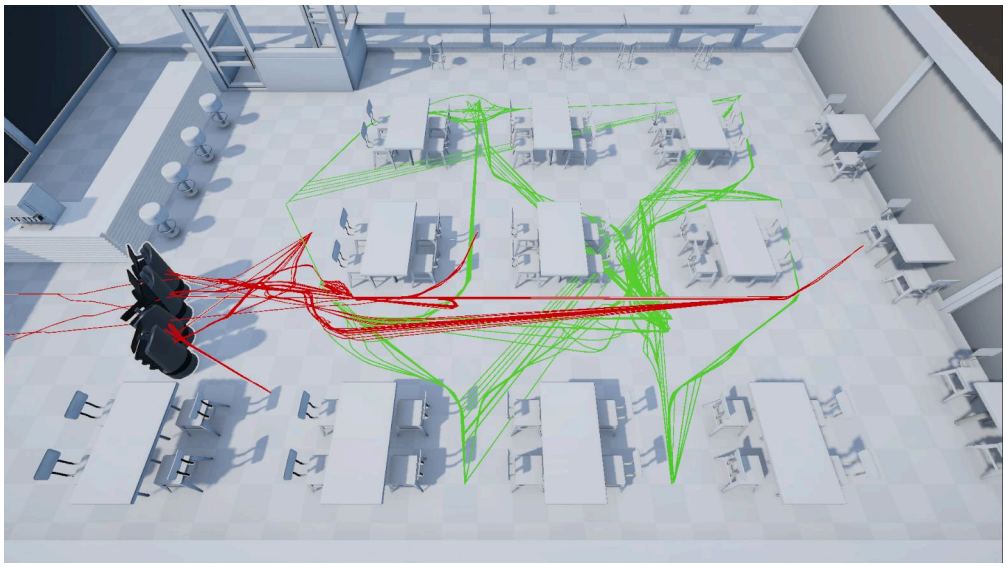


Figure 7.5: Initial test of the movement analysis

Due to the isolated environment working with limited functionality, the part that places the robots towards their starting place does not work. This means that the very first run shows faulty data, up untill the return point. Therefore, the lines on the left side of the image can safely be ignored.

Differences in return lines

Like previously explained, the humans are teleported back to their origin point every time a series of orders is marked complete. As the PathManager always measures the paths in a continuous way, this creates a line between sample points. If one were to zoom in on the point at which human agents get teleported back to their origin, it is visible that this seems to happen within a fairly large area. In a perfect simulation, this should all happen at the exact same point, but there seems to be about 1.5 meters of difference in these points.

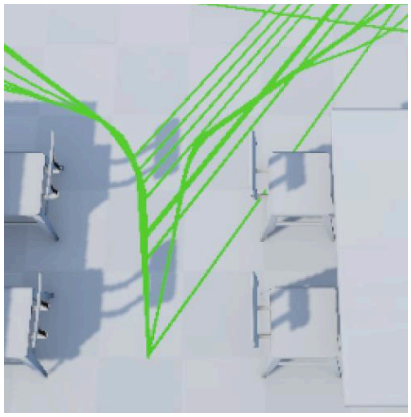


Figure 7.6 deviating return lines

There would, of course, be a certain amount of inaccuracy created by the PathManager, as it only samples every 0.25 seconds. However, as the humans have a walking speed of 1.1 m/s, that should result in a change of about a quarter of a meter at best. While this shifting endpoint seems to be an indicator for the butterfly effect at work, it unfortunately does not point to the source.

High deviance in collision areas

Like previously explained, the humans are teleported back to their origin point every time a series of orders is marked complete. As the PathManager always measures the paths in a continuous way, this creates a line between sample points. If one were to zoom in on the point at which human agents get teleported back to their origin, it is visible that this seems to happen within a fairly large area. In a perfect simulation, this should all happen at the exact same point, but there seems to be about 1.5 meters of difference in these points.

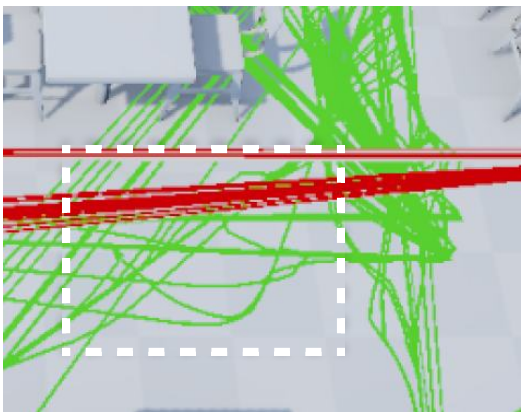


Figure 7.7 (green) path glitches

There would, of course, be a certain amount of inaccuracy created by the PathManager, as it only samples every 0.25 seconds. However, as the humans have a walking speed of 1.1 m/s, that should result in a change of about a quarter of a meter at best. While this shifting endpoint seems to be an indicator for the butterfly effect at work, it unfortunately does not point to the source.

Probable causes

This leads to the belief that there are two main limitations that could potentially cause this. The first being inconsistencies in the pathfinding, and the other one being deviations caused by agents getting stuck or deflecting in a different way.

It is important to at least know what causes this inconsistency in testing data. Therefore, more experiments are conducted in order to narrow down the main cause of these path deviations.

7.5 Isolated experiment on pathfinding consistency

Another theory of the desynchronization was based on inconsistencies with the pathfinding method that was used. As was mentioned on page 54, the A* algorithm should theoretically find their way to the closest corner of any obstacle in order to pass it. Seeing as it does so on the same pre-generated navigation mesh, those corners should theoretically be the same.

Experiment setup

To actually validate this method, an edit of the main agent script was made so that it generates a route, records it, follows the entire route, and runs it again. This was done with the same set of orders that was used in the other validation tests. The main difference of this overlay compared to the others would be that it does not record any deviations in the pathfinding during runtime.

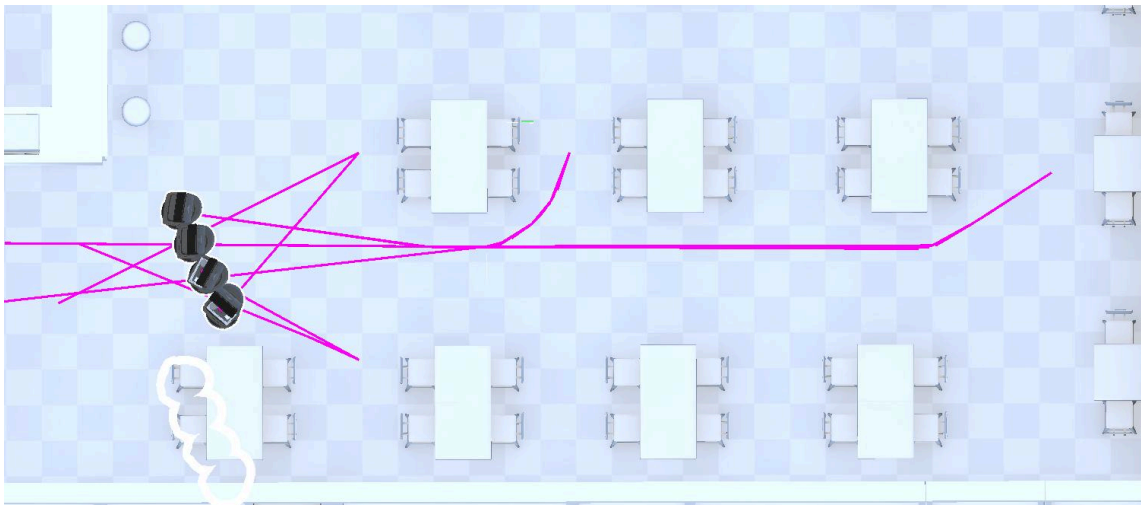


Figure 7.8: The calculated routes with a sample size of 15

Results and conclusion

Figure 7.8 shows a combination of 15 routes that were recorded and overlaid. It should be noted that the lines on the left side of the robots can once again be ignored due to technical limitations. The absence of major deviations besides a slight spread on the right corner means that the pathfinding calculation is almost perfectly consistent.

With that ruled out, the main cause for desynchronization would likely be physics related only.

7.6 Isolated experiment on the influence of framerate on collision behaviour

Now that pathfinding inconsistencies have been ruled out, the root cause of deviations would be collisions in the physics only. If that would be the case, then it would likely be influenced by the lower framerate that was set previously. In order to evaluate this, it is important to compare two different situations to see if a higher framerate would create more consistency in the results.

Experiment setup

To maximise the difference the two rounds would make, it is important to take extreme values. One of the rounds will therefore run at around 10 frames per second, while the other will run at around 40 frames per second. While a complete removal of the framerate lock would likely run between 50 to 60 frames per second, it would allow eventual frame dips to alter the data. Therefore, in the isolated environment, a stable 40 frames per second should be sustainable.

Seeing as the results themselves show too much noise, the choice was made to simply overlay a perfect screenshot taken after the experiment was run. This means that the comparison between the two test runs are done purely on visuals.

Results

As a direct overlay for comparison is hard to create in a single image, the two images were overlaid in photoshop and inverted. Figure 7.9 shows the combination of these two. The darker lines are the routes from the lower framerate, while the bright ones are ones with a higher framerate. Figure 7.10 and 7.11 show the separate results as well.

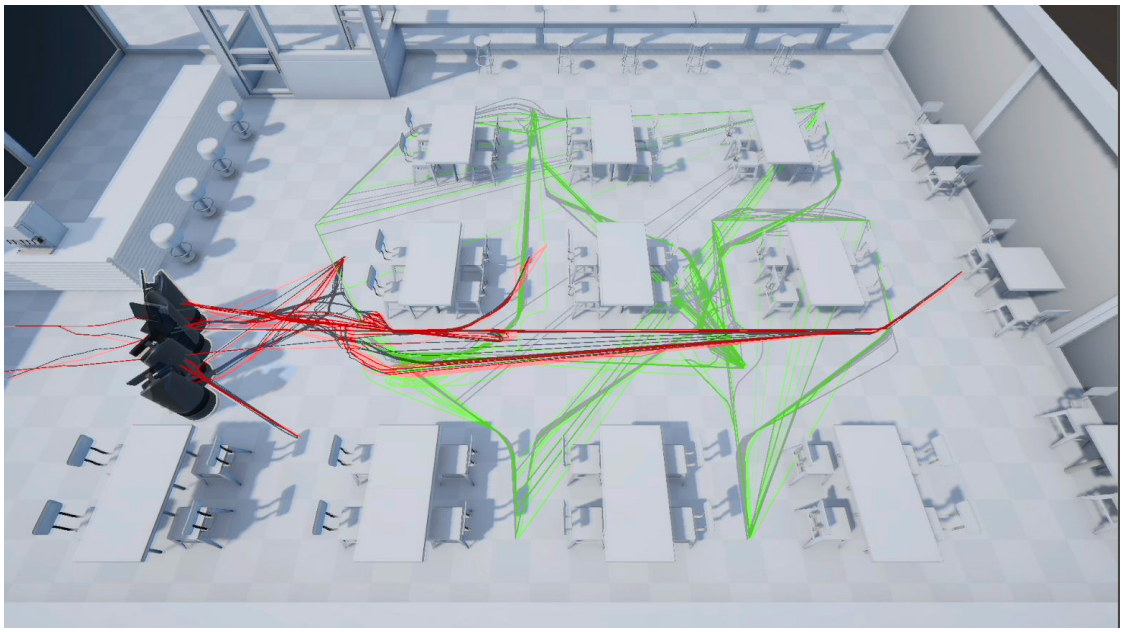


Figure 7.9: A combined image of both routes

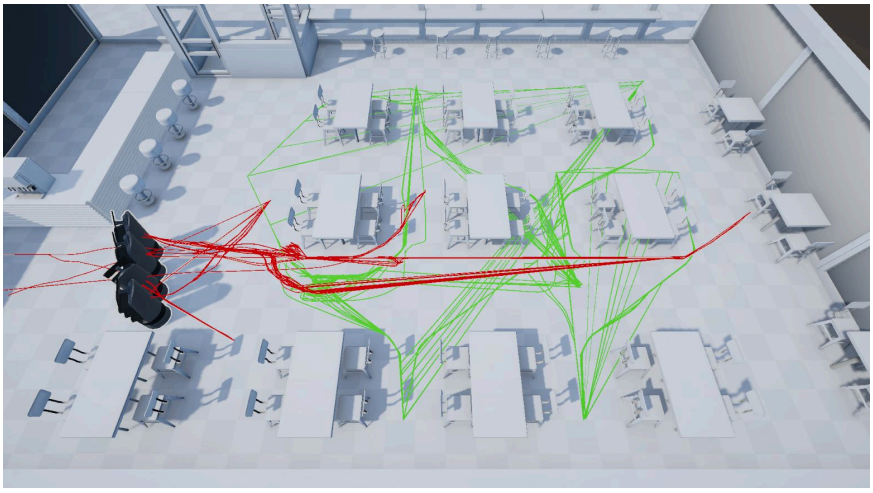


Figure 7.10: 15 paths at 40 frames per second (FPS)

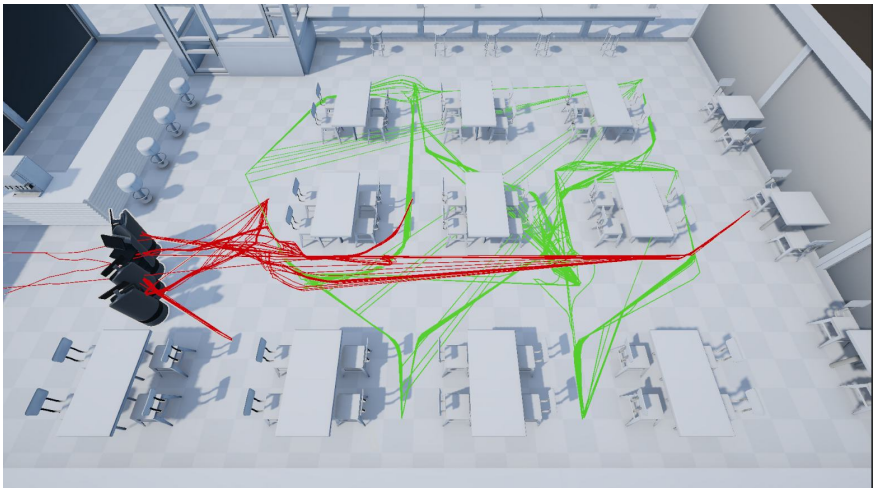


Figure 7.11: 15 paths at 10 frames per second (FPS)

Results and conclusion

As can be observed mainly in figure 7.10, the paths seem to be much more consistent at a higher framerate. The red line in the middle of the image is a good example for this. This curve is a result from an angular deflection between a human and robot agent. The 10 FPS path seems to snap much earlier to the intended path, which means that the angular deflection seems to be different every time.

Another point of evidence is the spread of return lines, which seems to be much closer within one another. The best example is the V-shape from the green lines on the lower left. This is evidence that collisions take a much more consistent time to resolve. Despite not being officially recorded, the 40 fps version was still observed to score between 9.21 and 14.41 on the average obstruction score, despite the increase in stability.

7.7 Conclusion

In the pursuit of creating a testing system that properly quantifies the effectiveness of several visualisation methods, a whole design process was started in itself. While the value of the different visualisation methods has yet to be determined with a quantifiable number, much was discovered during the process of creating a simulated restaurant environment.

It was however observed that the pathfinding and the scoring systems are working as intended. The main bottleneck of the reliability of this experiment setup would be the agent's inability to resolve collision conflicts on their own. This is not completely unexpected, as the robot navigation principles were not the main focus on this research, and were therefore given very basic logic to operate from. In the absence of any real solution from the operator's side, a complete reliance on a simple navigation system is therefore not ideal for a precise test.

However, in the process of creating more consistent results by finetuning the simulated environment, it was discovered that all other aspects are now much more reliable. The new methods to test these paths also make advanced movement analysis much more accessible.

If these points are compared to the original demands and wishes that were taken in mind, it is found that the simulation does indeed support a spectrum of control ranging from fully autonomous robot behaviour to fully teleoperated control, and provide methods to evaluate performance across intermediate control levels. Despite the fact that these cannot be evaluated in a numerical value as of yet.

It is therefore believed that unless more time was available to explore alternative methods, the main value of this project is derived from the creation of a universal simulation that allows for easy integration of various machine vision visualisations, and direct robot control methods. This piece of software, along with the report on the design choices made during this process can therefore be used in future projects mentioned in the following chapter.

08

Discussion, reflection and recommendations

8.1 Discussion

Simplified system
In hindsight, the visualisation methods could also have been tested by a "fake" simulation, where the robot and human agents aren't actually simulated at all, but rather following a pre-programmed animation. The user could then have been tasked to identify a series of potential conflicts, and the rate at which they were accurately able to predict the optimal situation to intervene in could then be used as a way of measuring it's success.

Bigger sample size
Calling a halt to the main tests after just two unusable data sets might have been a bit premature, as they theoretically could have been two outliers. However, focussing too much on a solution that proved to be ineffective multiple times also would have been a common pitfall.

Priority shift in testing process
During the development process of this simulation environment, too much emphasis was eventually put into proving the effectiveness of a smaller portion of the whole product. The main focus on validating the visualisation methods took much more resources than the value that would otherwise be gained by performing qualitative research based on subjective user feedback.

Admittedly, I am aware that this is a scientific thesis, which means that formal tests have to be conducted in such a way that it is possible to prove a distinct relation between changing a variable and getting a clear correlation.

It also would have been wiser to focus on creating value with this simulation as a platform for other researchers to test their projects in at a much earlier stage, but the constant near-readiness of the final experiment kept delaying that final choice longer than it should have.

Finally, a high amount of noise in the data would not have to mean that testing had to be stopped, as the survey also provides data to some degree. However, with the test taking roughly an hour, putting participants through the entire process without actually measuring testing data felt inefficient at the time. Due to time constraints however, further testing was not possible.

Alternative deliverable
As an alternative deliverable for conducting a formal test, a written guide for other students or university staff to incorporate their own features into the simulation would have provided more value from both a research viewpoint, as well as a practical viewpoint. This might still be done at a later stage within

8.2 Recommendations on how to proceed

Random orders might have been the most reliable form of measurement

As the butterfly effect constantly seems to affect the final results, an over reliance on this system might have been a mistake in hindsight. While a randomised system would still cause glitches to happen over time, skipping the chase of a perfectly controlled world would have allowed more time to be spent implementing one of the solutions below.

Creating three parallel simulations or a custom made physics engine could have prevented desynchronization issues

Due to the nature of how the physics engine in unity works, it is not possible to have an object enable collision for one object, but not for another. This means that in order for the robots to navigate around chairs, tables and other static obstacles, they also have to collide with all other agents. Seeing as glitching and getting stuck in unpredictable ways still seems to be the leading case of desynchronization, an extended period of time could have allowed for either one of two solutions:

Two parallel simulations

In this version, the restaurant, which is already a digital twin of a real world, gets yet another copy. One of the copies has the robots driving around in them, and another contains the human agents. Then, a third parallel world would also be created, with clone versions that constantly read the coordinates of the original counterparts, and copy their location. This would almost be the same system that is used to generate the “real world” that is visible with the robot camera view. All sensors would then be placed on the “cloned” versions of these agents, so that they will still have interactions such as stopping, and tracking scores.

Seeing as both types of agents don’t technically exist in the same world, they would practically phase through each other like ghosts, while keeping a minimal impact on the user experience. However, this would not entirely prevent agents of the same type from colliding in their own world. This could technically be mitigated by writing an automated system that generates a simplified private world for every agent, but that would not be possible due to obvious time constraints.

A custom physics engine

Another alternative method of resolving these issues would be to stop using a traditional physics engine for simulations, and programming one’s own engine. Seeing as the pathfinding is already custom made, it would not be impossible to apply basic laws of physics for a simplified environment such as this. It would mainly come down to creating newton’s third law where objects are just moved away by x amount if their coordinates fall between a designated no go zone. If that is combined with the pathfinding system, it would simply be a matter of implementing simple acceleration, and basically move the agents around the calculated route like a train riding a track.

8.3 Other recommendations for future research

Researching tele-operated robot to human communication in a socially complex environment.

During the course of this project, a framework was created to support a seamless transition between a bird's eye view of a restaurant environment, and a direct manual control over the selected robots. This system was built with scalability in mind, and can easily expanded upon using a standardised framework.

This means that it would be relatively easy for a future researcher to test out how the "sidewalk tango" can be resolved by implementing basic communication tools that the robot operator can control. For example, the operator assumes manual control over the robot's movement, and a set of eyes that can easily be moved with the mouse. This would allow the operator to signal basic things like gesturing the user to move past them, expressing shock or panic, or even nodding in acknowledgement towards humans that they get their intentions. These interactions also would not be limited to a screen either, and could very well be a small robot arm, or rotating head. Sometimes a single blinking led can be enough to perform basic communications with other humans.

As the groundwork for multiplayer integration is also already there, a secondary test subject could be asked to move from point A to B in the restaurant, and try to guess the robot's intentions based on the limited forms of communication that the robot operator is presented with.

VR integration and inverse kinematics
To take it even one step further, unity's native virtual reality support would allow the test humans to physically walk around the virtual robot, which not only improves their perception to a more detailed level, but also gives better feedback for the robot operator, using one of the many available inverse kinematics plugins.

Inverse kinematics work by creating a rig, which is a custom skeleton that animates itself based on the VR user's hands and head position. The rig then tries to predict the user's arm and knee bend based on those changes. This could further be expanded upon by adding additional trackers to the VR user, at the cost of more hassle in setting up testing environments.

Feasibility
The multiplayer framework is already present, but probably needs at least a few days to fully integrate and automatically connect to other clients. The VR implementation is already native, which means that it could be achieved in a single working day. If the choice is made to buy a more expensive Inverse Kinematics plugin, integration should not be more complicated than a plug and play, and would cost about 30 euro.

Using the simulation as testing grounds for AI controlled robots.

Seeing as there is already a system in place that automatically generates a navigation mesh in any environment, shows semi realistic graphics, and features programmable human agents walking around, the simulation could prove to be a challenging and highly controlled environment for fully autonomous robots.

With Unity's camera system, a camera feed can be directly plugged into another program, which essentially functions as any real camera. This can also be combined with the already existing LIDAR that is programmed in the simulation. Point cloud data can therefore be processed and sent to an external output, which would be received the same way a device such as an intel RealSense LIDAR camera would.

Not only would this make training robots and AI models faster, it would also provide the tools to give feedback on the success rate of the model, by providing a birds eye view of the simulation, and actively keeping track of human stopping score. This data can be directly used as feedback for the AI model itself as well, which means that it would theoretically be possible to fully automate the training process.

Using the simulation as a study environment for machine vision.

There often seems to be a gap between the ideas of machine vision in controlled environments, and the potential overstimulation it can cause in an actual environment, especially one with moving humans. While classical programs like gazebo naturally integrate in systems such as ROS, taking the time to import these visualisation methods into unity could provide a much more agile environment to quickly prototype these interactions.

Unity already has extensions such as the realvirtual.io suite, which allows simulated sensors to essentially give the same output as the real sensor would. Combined with the "real" environment filled with unpredictable humans that can be programmed to follow specific commands, simple A/B tests for machine vision would simply be limited by how fast the raw sensor input can be visualised using either c# or python.

09

Appendix and references

9.1 Sources

Chen, C., & Cai, R. (2024). Are robots stealing our jobs? Examining robot phobia as a job stressor in the hospitality workplace. International Journal of Contemporary Hospitality Management, 37(1), 94 to 112.
https://doi.org/10.1108/ijchm-09-2023-1454

CK 12 Foundation. (2025, September 11). Flexi answers. How many miles per hour is the average human walking speed?
https://www.ck12.org/flexi/biology/locomotion/how-many-miles-per-hour-is-the-average-human-walking-speed/

El Gazar, H. E., Abdelhafez, S., Ali, A. M., Shawer, M., Alharbi, T. A. F., & Zoromba, M. A. (2024). Are nurses and patients willing to work with service robots in healthcare? A mixed methods study. BMC Nursing, 23(1).
https://doi.org/10.1186/s12912-024-02336-7

Hart, S. G., & Staveland, L. E. (n.d.). NASA Task Load Index.
https://humansystems.arc.nasa.gov/groups/tlx/downloads/TLXScale.pdf
Intersys Limited. (n.d.).

How do AI robot waiters in restaurants help?
https://www.intersyslimited.com/blog/how-robot-waiters-in-restaurants-work-a-glimpse-into-the-future/

Li, W., Ding, H., Gui, J., & Tang, Q. (2024). Patient acceptance of medical service robots in the medical intelligence era. Humanities and Social Sciences Communications, 11(1).
https://doi.org/10.1057/s41599-024-04028-8

Li, Y., & Wang, C. (2021). Effect of customer perception on service robot acceptance. International Journal of Consumer Studies, 46(4), 1241 to 1261.
https://doi.org/10.1111/ijcs.12755

Schellekens, M. (2022). Human machine interaction in self driving vehicles. International Journal of Law and Information Technology, 30(2), 233 to 258.
https://academic.oup.com/ijlit/article/30/2/233/6589379

MacCarthy, M. (2025, 8 augustus). Setting the standard of liability for self-driving cars. Brookings.
https://www.brookings.edu/articles/setting-the-standard-of-liability-for-self-driving-cars/

Lewis, M., Sycara, K., & Walker, P. (2018). The Role of Trust in Human-Robot Interaction. In Studies in systems, decision and control (pp. 135–159).
https://doi.org/10.1007/978-3-319-64816-3_8

Messe Düsseldorf GmbH. (n.d.). Make way for rolling assistants! More robotics within hospital logistics.
https://www.medica-tradefair.com/en/media-news/spheres-of-medica-magazine/medtech-devices/robotics-hospital-logistics

Diligent Robotics. (n.d.). Moxi.
https://www.diligentrobots.com/moxi

Nordhoff, S. (2024). A conceptual framework for automation disengagements. Scientific Reports, 14(1), 8654.
https://doi.org/10.1038/s41598-024-57882-6

Parvez, M. O., Öztüren, A., Cobanoglu, C., Arasli, H., & Eluwole, K. K. (2022). Employees' perception of robots and robot induced unemployment in the hospitality industry under COVID 19. International Journal of Hospitality Management, 107, 103336.
https://doi.org/10.1016/j.ijhm.2022.103336

Personal attitudes towards robot assisted health care. (2008). PubMed.
https://pubmed.ncbi.nlm.nih.gov/18560068/

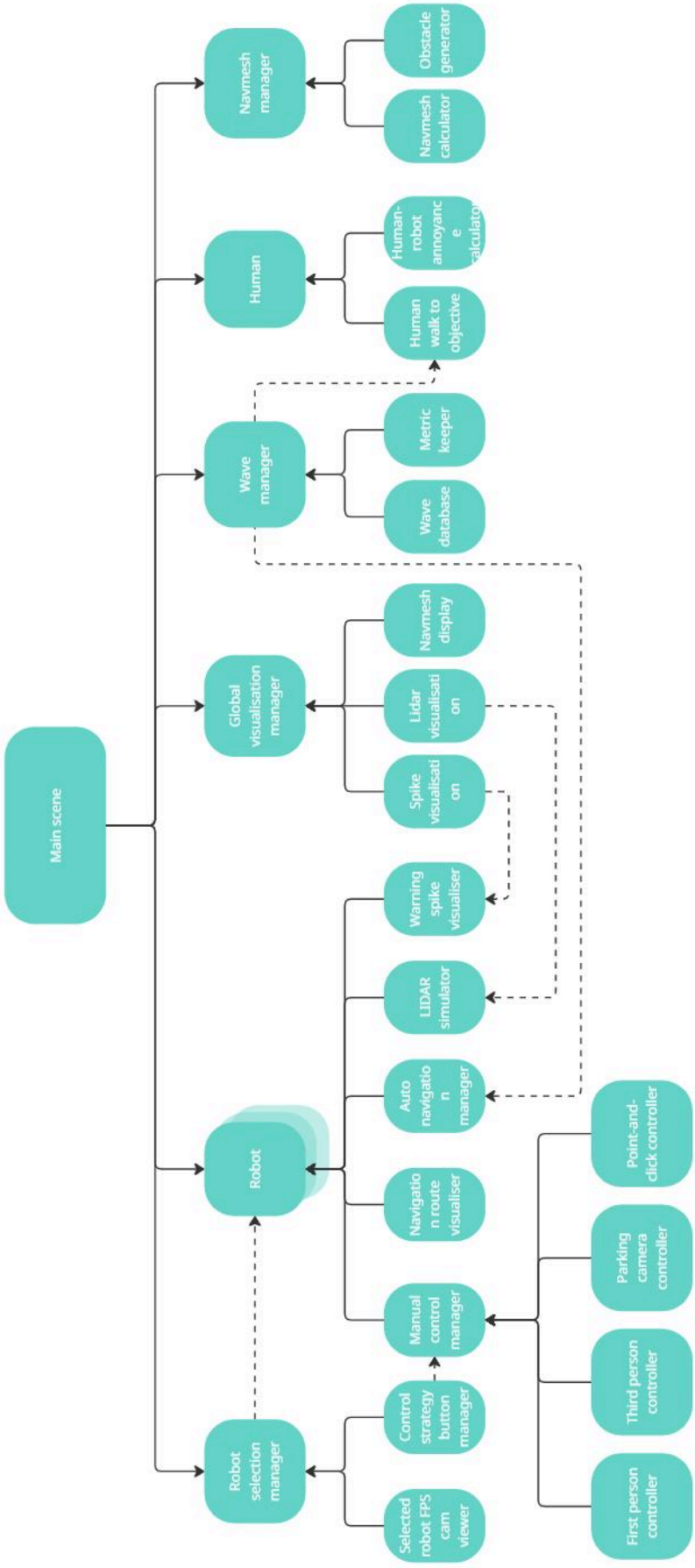
PROVEN Robotics. (2024, January 10). What are the advantages and disadvantages of robot waiters?
https://provenrobotics.ai/robot-waiter-advantages-and-disadvantages/

Wikipedia contributors. (2025, December 15). Preferred walking speed. Wikipedia.
https://en.wikipedia.org/wiki/Preferred_walking_speed/

Esterwood, C., & Robert, L. P. (2023). The theory of mind and human–robot trust repair. Scientific Reports, 13(1), 9877.
https://doi.org/10.1038/s41598-023-37032-0

Naneva, S., Gou, M. S., Webb, T. L., & Prescott, T. J. (2020). A Systematic Review of Attitudes, Anxiety, Acceptance, and Trust Towards Social Robots. International Journal Of Social Robotics, 12(6), 1179–1201.
https://doi.org/10.1007/s12369-020-00659-4

Software architecture



The data that showed a deviation in the amount of stops

23		Standard deviation	7.390685
12		Mean	24.8
34			
28			
18			
36			
28			
18			
25			
26			

Example snipped of the excel file that feeds the WaveManager

#HUMAN 1	3	5	7
#HUMAN 2	0	0	0
#HUMAN 3	4	6	8
Wave	Table	Time take	Total stops
1	11		
1	5		
1	1		
1	2		
1	3		
1	9		
1	10		

Data from the main experiment subject 1

Round	Wave	Table	Time taken	Average De	Total Human Stopping Score
1	1	3	54	85.79	114.69
1	1	8	54.47	85.79	114.69
1	1	6	145.05	85.79	114.69
1	1	10	89.64	85.79	114.69
1	2	7	69.11	64.11	97.58
1	2	4	71.35	64.11	97.58
1	2	0	36.32	64.11	97.58
1	2	12	79.66	64.11	97.58
1	3	9	72.13	35.13	6.53
1	3	1	18.49	35.13	6.53
1	3	2	15.3	35.13	6.53
1	3	5	34.58	35.13	6.53
1	4	10	87.85	56.27	40.14
1	4	3	47.54	56.27	40.14
1	4	7	57.21	56.27	40.14
1	4	0	32.46	56.27	40.14
1	5	6	102.95	74.71	114.78
1	5	11	100.17	74.71	114.78
1	5	9	73.88	74.71	114.78
1	5	2	21.86	74.71	114.78
1	6	8	52.96	49.92	28.59
1	6	0	29.22	49.92	28.59
1	6	4	45.02	49.92	28.59
1	6	12	72.45	49.92	28.59
1	7	5	36.42	59.48	62.26
1	7	7	55.14	59.48	62.26
1	7	3	49.81	59.48	62.26
1	7	11	96.55	59.48	62.26
2	1	10	110.08	63.4	143.13
2	1	6	58.31	63.4	143.13
2	1	1	19.68	63.4	143.13
2	1	4	65.52	63.4	143.13
2	2	2	18.25	46.67	80.85
2	2	12	73.67	46.67	80.85
2	2	7	63.96	46.67	80.85
2	2	0	30.79	46.67	80.85
2	3	3	62.58	66.07	73.22
2	3	9	80.09	66.07	73.22
2	3	11	80.8	66.07	73.22
2	3	4	40.8	66.07	73.22
2	4	1	18.45	37.35	30.71
2	4	5	31.4	37.35	30.71
2	4	0	31.39	37.35	30.71
2	4	0	31.39	37.35	30.71
2	4	8	68.17	37.35	30.71
2	5	12	79	45.86	52.69
2	5	3	47.99	45.86	52.69
2	5	4	39.42	45.86	52.69
2	5	2	17.03	45.86	52.69
2	6	6	69.37	67.36	76.56
2	6	9	86.28	67.36	76.56
2	6	3	84.53	67.36	76.56
2	6	0	29.26	67.36	76.56
2	7	11	78.16	38.27	14.22
2	7	4	35.32	38.27	14.22
2	7	1	18.92	38.27	14.22
2	7	2	20.69	38.27	14.22
3	1	7	56.37	47.2	21.54
3	1	5	36.54	47.2	21.54
3	1	2	18.12	47.2	21.54
3	1	11	77.77	47.2	21.54
3	2	0	39.6	61.1	36.83
3	2	10	76.87	61.1	36.83
3	2	6	73.76	61.1	36.83
3	2	8	54.16	61.1	36.83
3	3	12	88.37	65.65	68.26
3	3	4	53.41	65.65	68.26
3	3	3	46.67	65.65	68.26
3	3	9	74.16	65.65	68.26
3	4	7	70.28	55.41	49.29
3	4	0	39.29	55.41	49.29
3	4	11	79.03	55.41	49.29
3	4	5	33.03	55.41	49.29
3	5	4	48.3	45.63	69.54
3	5	2	23.05	45.63	69.54
3	5	1	25.02	45.63	69.54
3	5	12	86.16	45.63	69.54
3	6	2	19.13	45.2	40.5
3	6	10	78.03	45.2	40.5
3	6	0	38.74	45.2	40.5
3	6	3	44.92	45.2	40.5
3	7	9	73.14	55.39	79.24
3	7	8	57.42	55.39	79.24
3	7	6	72.4	55.39	79.24
3	7	1	18.58	55.39	79.24
4	1	1	18.99	37.45	21.26
4	1	7	62.48	37.45	21.26

Data from the main experiment subject 1

4	1	4	37.01	37.45	21.26
4	1	5	31.32	37.45	21.26
4	2	12	80.55	75.8	163.32
4	2	2	20.37	75.8	163.32
4	2	10	130.29	75.8	163.32
4	2	6	71.98	75.8	163.32
4	3	5	44.38	52.58	62.39
4	3	8	48.16	52.58	62.39
4	3	0	33.66	52.58	62.39
4	3	11	84.13	52.58	62.39
4	4	4	47.09	69.5	143.57
4	4	3	48.74	69.5	143.57
4	4	9	161.38	69.5	143.57
4	4	1	20.78	69.5	143.57
4	5	11	122.3	59.99	165.37
4	5	1	19.55	59.99	165.37
4	5	6	66.21	59.99	165.37
4	5	0	31.91	59.99	165.37
4	6	2	24.73	57.82	23.07
4	6	3	47.27	57.82	23.07
4	6	10	83.77	57.82	23.07
4	6	12	75.51	57.82	23.07
4	7	7	52.97	60.12	72.31
4	7	6	106.11	60.12	72.31
4	7	8	50.62	60.12	72.31
4	7	0	30.79	60.12	72.31
5	1	3	70.87	54.93	33.89
5	1	10	75.02	54.93	33.89
5	1	4	37.31	54.93	33.89
5	1	1	36.52	54.93	33.89
5	2	0	35.79	53.44	31.85
5	2	12	75.03	53.44	31.85
5	2	9	83.6	53.44	31.85
5	2	2	19.33	53.44	31.85
5	3	11	85.67	56.65	61.38
5	3	2	17.69	56.65	61.38
5	3	6	87.62	56.65	61.38
5	3	4	35.61	56.65	61.38
5	4	8	48.18	61.37	172.51
5	4	7	125.99	61.37	172.51
5	4	0	34.14	61.37	172.51
5	4	5	37.17	61.37	172.51
5	5	10	85.52	68.43	64.41
5	5	1	21.21	68.43	64.41
5	5	9	74.47	68.43	64.41
5	5	6	92.51	68.43	64.41
5	6	4	39.57	42.37	29.85
5	6	0	37.28	42.37	29.85
5	6	5	40.34	42.37	29.85
5	6	8	52.29	42.37	29.85
5	7	6	87.01	79.84	121.74
5	7	3	65.24	79.84	121.74
5	7	7	80.36	79.84	121.74
5	7	12	86.74	79.84	121.74

Data from the main experiment subject 2

Round	Wave	Table	Time taken	Average De	Total Human Stopping Score
1	1	3	53.89	70.67	59.12
1	1	8	54.47	70.67	59.12
1	1	6	94.74	70.67	59.12
1	1	10	79.59	70.67	59.12
1	2	7	61.38	60.85	84.62
1	2	4	69.77	60.85	84.62
1	2	0	37.05	60.85	84.62
1	2	12	75.19	60.85	84.62
1	3	9	71.73	35.1	5.25
1	3	1	18.55	35.1	5.25
1	3	2	15.49	35.1	5.25
1	3	5	34.64	35.1	5.25
1	4	10	85.76	56.16	30.21
1	4	3	48.61	56.16	30.21
1	4	7	57.75	56.16	30.21
1	4	0	32.53	56.16	30.21
1	5	6	107.54	70.25	95.21
1	5	11	78.19	70.25	95.21
1	5	9	73.27	70.25	95.21
1	5	2	21.99	70.25	95.21
1	6	8	54.47	52.01	38.93
1	6	0	35.87	52.01	38.93
1	6	4	40.51	52.01	38.93
1	6	12	77.18	52.01	38.93
1	7	5	36.66	53.87	26.96
1	7	7	54.03	53.87	26.96
1	7	3	44.2	53.87	26.96
1	7	11	80.6	53.87	26.96
2	1	10	112.18	63.19	127.31
2	1	6	66.86	63.19	127.31
2	1	1	20.57	63.19	127.31
2	1	4	53.13	63.19	127.31
2	2	2	18.49	46.41	72.74
2	2	12	74.04	46.41	72.74
2	2	7	62.18	46.41	72.74
2	2	0	30.92	46.41	72.74
2	3	3	60.94	69.81	62.55
2	3	9	94.5	69.81	62.55
2	3	11	82.49	69.81	62.55
2	3	4	41.31	69.81	62.55
2	4	1	19.21	40.87	20.31
2	4	5	41.25	40.87	20.31
2	4	0	37.54	40.87	20.31
2	4	8	65.49	40.87	20.31
2	5	12	81.85	46.32	55.7
2	5	3	47.65	46.32	55.7
2	5	4	36.26	46.32	55.7
2	5	2	19.53	46.32	55.7
2	6	6	59.01	54.73	45.83
2	6	9	78.19	54.73	45.83
2	6	3	49.26	54.73	45.83
2	6	0	32.48	54.73	45.83
2	7	11	82.9	39.16	21.52
2	7	4	36.6	39.16	21.52
2	7	1	21.86	39.16	21.52
2	7	2	15.29	39.16	21.52
3	1	7	57.32	48.95	30.02
3	1	5	36.67	48.95	30.02
3	1	2	15.43	48.95	30.02
3	1	11	86.4	48.95	30.02
3	2	0	34.92	59.51	28.87
3	2	10	77.67	59.51	28.87
3	2	6	73.74	59.51	28.87
3	2	8	51.69	59.51	28.87
3	3	12	81	65.41	104.64
3	3	4	40.98	65.41	104.64
3	3	3	47.63	65.41	104.64
3	3	9	92.02	65.41	104.64
3	4	7	60.68	51.98	39.74
3	4	0	31.38	51.98	39.74
3	4	11	78.83	51.98	39.74
3	4	5	37.02	51.98	39.74
3	5	4	51.07	42.17	42.62
3	5	2	15.27	42.17	42.62
3	5	1	19.53	42.17	42.62
3	5	12	82.82	42.17	42.62
3	6	2	26.53	46.51	44.93
3	6	10	80.55	46.51	44.93
3	6	0	32.54	46.51	44.93
3	6	3	46.4	46.51	44.93
3	7	9	84.32	54.74	62
3	7	8	54.85	54.74	62
3	7	6	60.07	54.74	62
3	7	1	19.73	54.74	62
4	1	1	19.45	39.06	17.28
4	1	7	63.2	39.06	17.28
4	1	4	38.16	39.06	17.28

Data from the main experiment subject 2

4	1	5	35.44	39.06	17.28
4	2	12	71.5	58.07	22.83
4	2	2	20.31	58.07	22.83
4	2	10	77.37	58.07	22.83
4	2	6	63.11	58.07	22.83
4	3	5	38.86	52.43	49.42
4	3	8	50.67	52.43	49.42
4	3	0	35.71	52.43	49.42
4	3	11	84.49	52.43	49.42
4	4	4	46.59	62.01	83.49
4	4	3	55.39	62.01	83.49
4	4	9	121.57	62.01	83.49
4	4	1	24.5	62.01	83.49
4	5	11	104.47	56.63	77.82
4	5	1	23.08	56.63	77.82
4	5	6	67.14	56.63	77.82
4	5	0	31.82	56.63	77.82
4	6	2	24.8	58.16	25.98
4	6	3	47.71	58.16	25.98
4	6	10	79.31	58.16	25.98
4	6	12	80.81	58.16	25.98
4	7	7	58.83	52.29	50.5
4	7	6	63.73	52.29	50.5
4	7	8	53.97	52.29	50.5
4	7	0	32.64	52.29	50.5
5	1	3	46.16	44.52	25.85
5	1	10	71.67	44.52	25.85
5	1	4	40	44.52	25.85
5	1	1	20.25	44.52	25.85
5	2	0	29.34	47.46	4.57
5	2	12	69.86	47.46	4.57
5	2	9	73.51	47.46	4.57
5	2	2	17.12	47.46	4.57
5	3	11	118.1	64.31	94.78
5	3	2	27.18	64.31	94.78
5	3	6	72.31	64.31	94.78
5	3	4	39.65	64.31	94.78
5	4	8	54.85	45.23	8.69
5	4	7	59.76	45.23	8.69
5	4	0	33.32	45.23	8.69
5	4	5	32.97	45.23	8.69
5	5	10	84.44	62.36	28.7
5	5	1	20.97	62.36	28.7
5	5	9	78.11	62.36	28.7
5	5	6	65.91	62.36	28.7
5	6	4	40.25	43.84	32.96
5	6	0	29.31	43.84	32.96
5	6	5	32.5	43.84	32.96
5	6	8	73.29	43.84	32.96
5	7	6	71.03	62.75	68.86
5	7	3	46.04	62.75	68.86
5	7	7	53.72	62.75	68.86
5	7	12	80.2	62.75	68.86

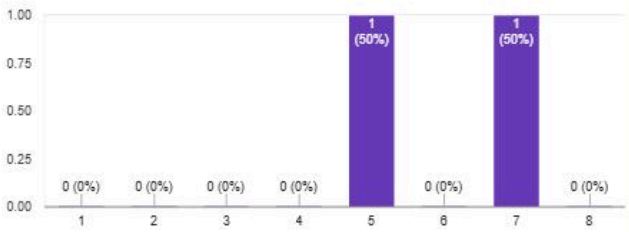
Survey: Corner camera + LIDAR

Corner camera + LIDAR Dots

How mentally demanding did you consider this visualisation method to be?

2 responses

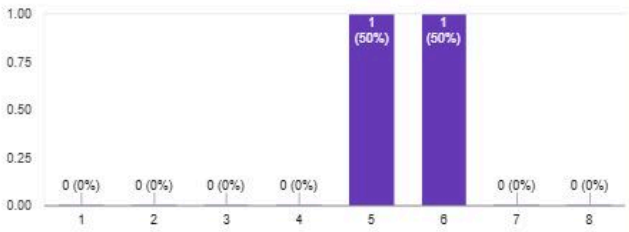
Copy



Did this visualisation method make you feel relaxed?

2 responses

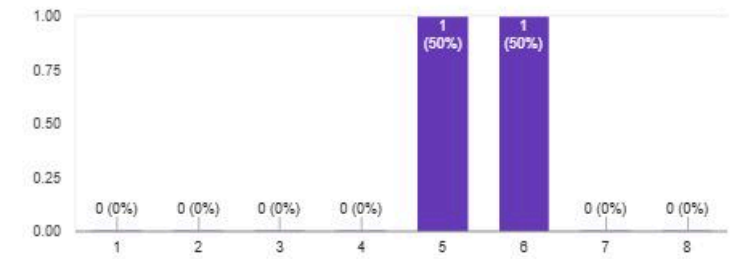
Copy



Did you feel like this visualisation method made you perform better?

2 responses

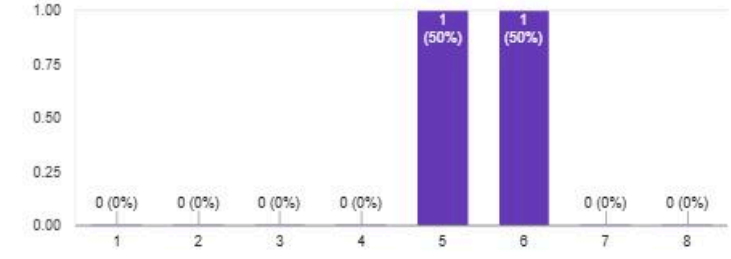
Copy



Did you find yourself frustrated by this visualisation method?

2 responses

Copy



Do you have any questions or remarks on this visualisation method?

2 responses

mostly the switching between robots is annoying but the placement of the dots are very nice

I felt like I had to shift my eyes back and forth from time to time

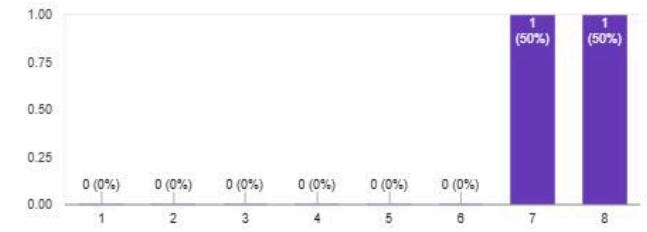
Survey: Popup camera + LIDAR

Popup camera + LIDAR dots

How mentally demanding did you consider this visualisation method to be?

Copy

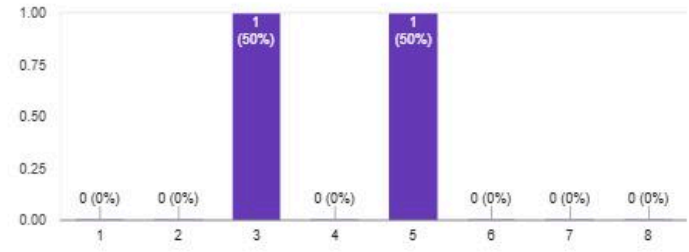
2 responses



Did you feel like this visualisation method made you perform better?

Copy

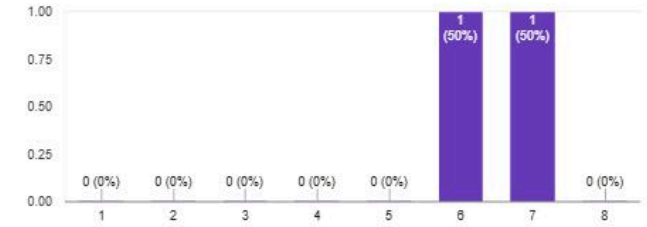
2 responses



Did this visualisation method make you feel relaxed?

Copy

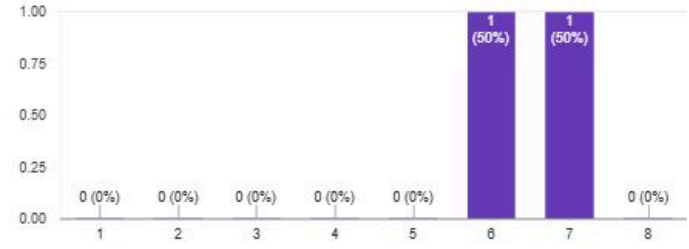
2 responses



Did you find yourself frustrated by this visualisation method?

Copy

2 responses



Do you have any questions or remarks on this visualisation method?

0 responses

No responses yet for this question.

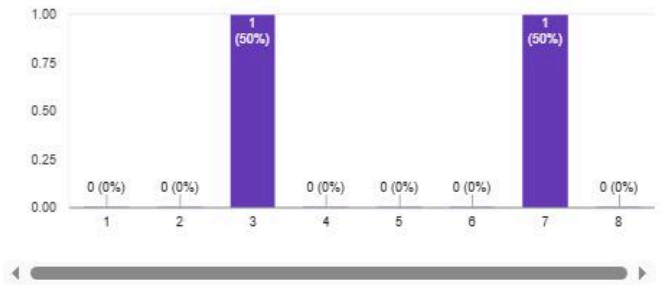
Survey: Corner camera + Risk bar

Corner camera + Risk visualisation

How mentally demanding did you consider this visualisation method to be?

Copy

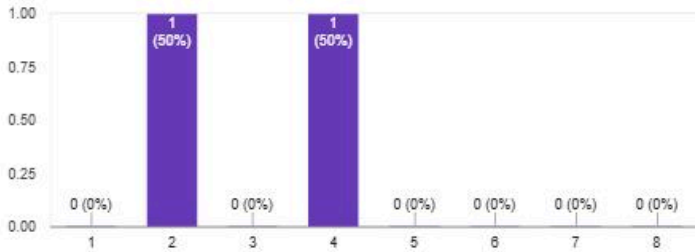
2 responses



Did you feel like this visualisation method made you perform better?

Copy

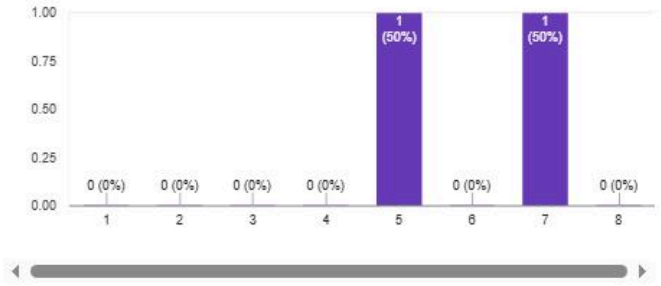
2 responses



Did this visualisation method make you feel relaxed?

Copy

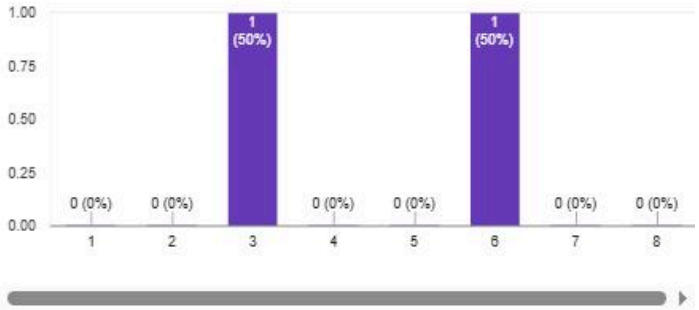
2 responses



Did you find yourself frustrated by this visualisation method?

Copy

2 responses



Do you have any questions or remarks on this visualisation method?

1 response

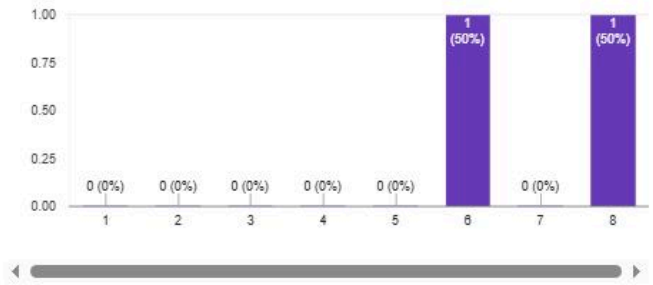
kennie zien! (the bars are confusing and I want to see the location of moving people)

Survey: Popup camera + Risk bar

Popup camera + risk visualisation

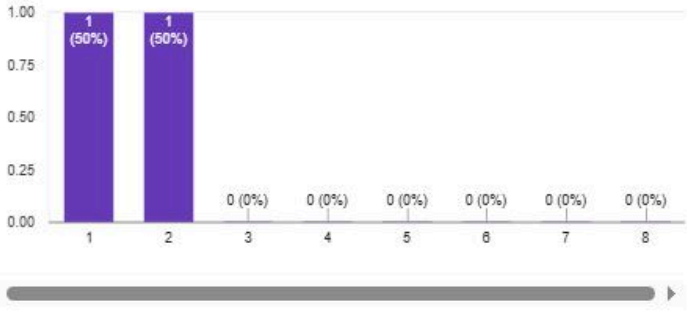
How mentally demanding did you consider this visualisation method to be? Copy

2 responses



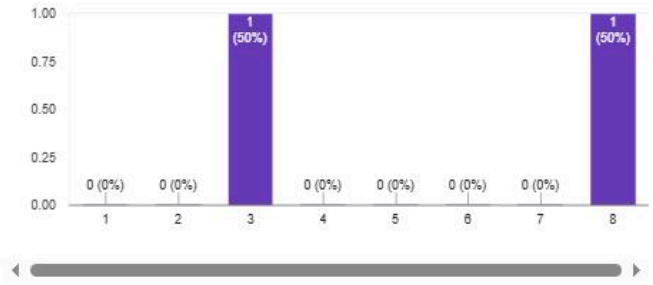
Did you feel like this visualisation method made you perform better? Copy

2 responses



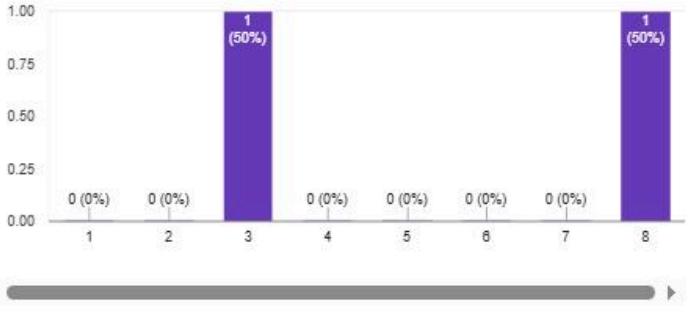
Did this visualisation method make you feel relaxed? Copy

2 responses



Did you find yourself frustrated by this visualisation method? Copy

2 responses



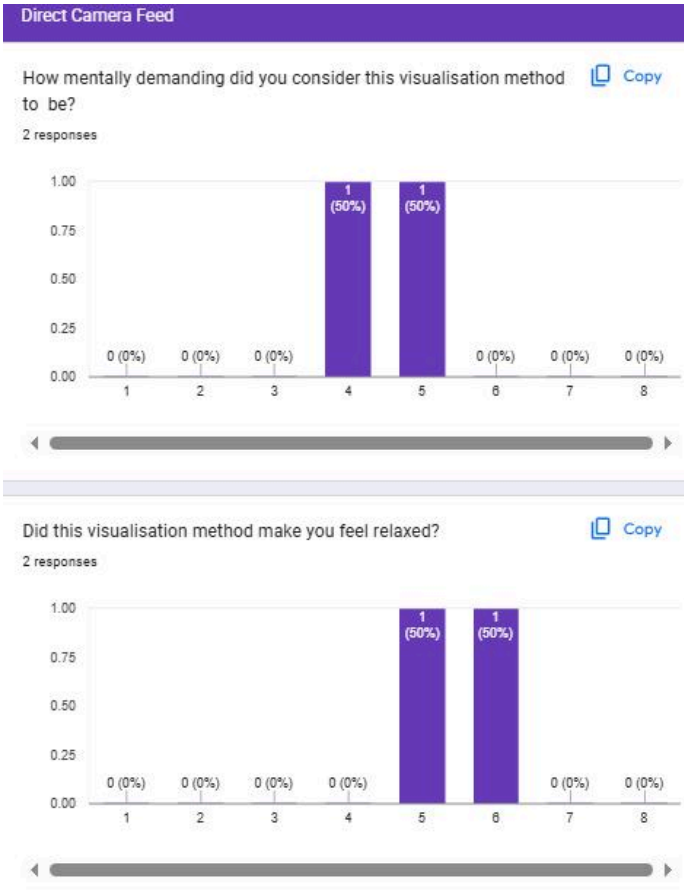
Do you have any questions or remarks on this visualisation method?

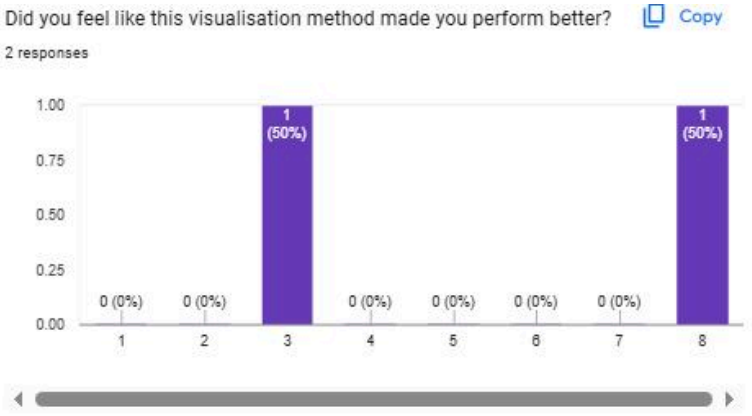
2 responses

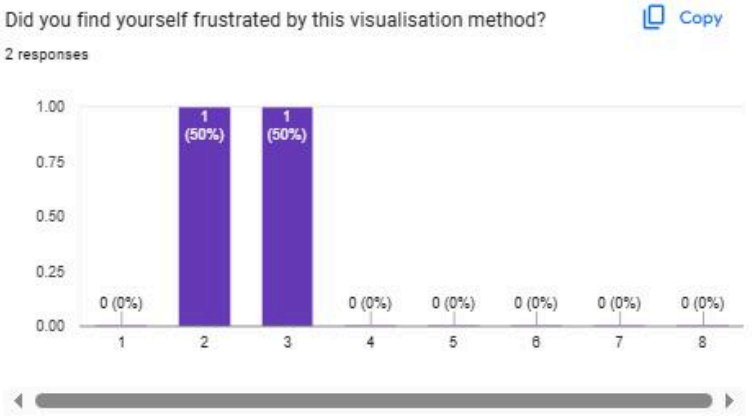
this was really overwhelming and the least informative i think (a suprising combination of too much information and too little usefull information)

The popup cameras add a lot of stress factor, which is not really softened by the bar sticking out of the robot, which worked well with the camera in the corner

Survey: Direct camera feed







Do you have any questions or remarks on this visualisation method?

1 response

line sometimes hard to see, frustration was also higher due to exhaustion

Top down view of the low FPS path tracing test