Preventing haul truck-related fatalities in open pit mining using gamified training simulation

by

Lennart Gerrit Vos

A Master's thesis
Lennart Gerrit Vos
Submitted to the department of Resource Engineering
TU Delft, Aalto University, RWTH Aachen
European Mining, Minerals and Environmental Program
Master of Science
21-12-2023

Contents

Abstract

Haul truck driving remains to be the most dangerous occupation in the mining industry (Kecojevic et al., 2007), leading in fatalities over other occupations in the mining industry. As the transition to autonomous haul trucks over the coming decades will not address this problem today, an immediate solution is needed that can be adopted at short notice to help aid in minimizing haul truck accidents. A gamified training simulator was developed that teaches haul truck drivers how to handle the hazardous situations that lead to the most injuries in the mine environment. The simulated nature of the training method allows truck drivers to gain knowledge on hazardous situations and immediately put them into practice by applying the gained knowledge during the different levels. Gamification aspects like a score system and progression through levels were added to increase engagement during the training simulation.

A statistical analysis was made on haul truck accidents from 1983 until 2022. The accidents that proved to be most relevant were included in the content of the training software. A training software was developed using Unreal Engine 5, where trucks could be driven using a Thrustmaster T150 Pro steering wheel and driving pedals. 18 participants from ages 19-65 underwent the training program and two tests of identical difficulty were taken before and after undergoing the training, in order to make a judgement on the progress of the participants by undergoing the training simulation. Participants were judged based on the number of crashes and traffic violations committed, attempts required to complete the test, as well as the time taken to finish the test. A score was calculated based on these factors providing an overall judgement of the participants' level of skill.

Average skill of participants increased by undergoing the gamified training, and training showed particularly good progress for underperformers in the initial test. After completing the gamified training the average score of participants was notably higher and the standard deviation was notably lower. It was concluded that all participants were brought to a comparably high level after completing the training.

Introduction

Haul truck driving is one of the most dangerous jobs in the mining industry. An analysis of equipment-related fatal accidents in U.S. mining operations between 1995 and 2005 by Kecojevic et al. (2007) stated that ''Among the MSHA equipment subcategories, the greatest proportion of fatalities is related to haul trucks (22.3%)''. Hazardous driving conditions often result in fatalities. As many mines transition to autonomous vehicles, the majority of mines still uses haul truck drivers as transitioning takes time and money, which might not be readily available yet. A solution needs to be presented at a very short notice, since the work, and thus fatal accidents as a result of this work will continue to happen until a full worldwide transition is made to autonomous haul trucks. It is essential to bring these numbers down by creating a safer environment and make drivers better equipped to deal with these hazardous situations.

According to the hierarchy of controls from the National Institute of Occupational Safety and Health (2023) hazards can be dealt with in decreasing order of effectiveness by elimination, substitution, engineering controls, administrative controls, and using personal protective equipment. Fatalities and serious injuries can have a magnitude of different causes and not all can be controlled at all times. So while it is of importance that mines continue to seek out ways of eliminating, substituting and engineering controls for all known hazards where possible, accidents may still occur from unknown or poorly managed hazards. According to a study performed by Cohen and Colligan (1998) there exists ''much direct and indirect evidence to show the benefits of training in ensuring safe and healthful working conditions.'' and ''Findings are near unanimous in showing how training can attain objectives such as increased hazard awareness among the worker groups at risk, knowledge of and adoption of safe work practices, and other positive actions that can reduce the risk and improve workplace safety.'' Safety trainings as an administrative control can thus further decrease the risk of an accident in these hazardous situations by better equipping drivers with skills to better recognize and deal with hazards. Training was a $61.1B industry in the U.S. in 2014, with safety-oriented training comprising 38.7% of the global training market (Brown & Poulton, 2018). Safety-oriented training is vital as one of the key forms of dealing with hazards.

Gamification is the application of typical game elements like point scoring and level progression to other areas of activity, like trainings. According to Legaki et al. (2020) gamified training might provide better results than conventional theory-based training. A new experience-based training method has been developed to create full immersion into certain hazardous scenarios to create the concept of flow. Flow is a concept that is defined by Hamari, J. (2014) as ''a state of optimal experience characterized by being fully focused and engaged in an activity''. Engagement is one of the key aspects regular training methods seek out to improve. A gamified training simulation combines learning theory and immediately putting the newly gained knowledge into practice. The gamified elements improve engagement for trainees in order to establish a state of flow; allowing trainees to fully focus on the task at hand and being immersed into the experience.

This research will answer the questions of which are the most important hazards that should be included in a gamified training simulation as practice objectives, can a gamified training simulator provide notable improvements in haul truck driving performance, and is a gamified training simulator an educational and engaging method to improve skills in recognizing and dealing with hazards while driving a haul truck?

A statistical analysis was made on haul truck accidents from 1983 until 2022. The accidents that proved to be most relevant were included in the content of the training software. A training software was developed using Unreal Engine 5, where trucks could be driven using a Thrustmaster T150 Pro steering wheel and driving pedals. 18 participants from ages 19-65 underwent the training program and a similar test was taken before and after undergoing the training in order to make a judgement on the progress of the participants by undergoing the training simulation.

The research will contain a literature review of examples of gamification in the mining industry and other industries, gamification user statistics, gamification in education and a statistical analysis on haul truck accidents. Furthermore, the methodology and development and deployment of the training software will be discussed. The results of the research will be presented, followed by a discussion and a conclusion.

Literature review

Gamification was a central concept in the development of the haul truck training software and can be defined as: ''the application of typical elements of game playing (e.g. point scoring, competition with others, rules of play) to other areas of activity, typically as an online marketing technique to encourage engagement with a product or service'' (Oxford Languages Dictionary). Several examples of gamification will be provided in the mining industry and other industries followed by user statistics on gamification and gamification in education. Finally, a statistical analysis of haul truck accidents will be presented.

Gamification examples in industries other than the mining industry

In 2021, the gamification industry had a market value of $11.9 billion dollars with an expected compound annual growth rate of 12.9% between now and 2025 (Boskamp, 2023). Google, Domino's Pizza, Samsung, Starbucks, and other companies implement gamification as a marketing tactic, corporate training, and a tool to increase engagement. Additionally, gamification is used in industries that include hospitality, retail, finance, healthcare, and education. (Bullock, 2023). Gamification applications have many benefits, as engagement is a clear way to bind clients to products and services. Conversion rates of websites with interactive content are nearly six times higher than those without it (Chang, 2023). More companies are starting to use gamification applications, and the gamification market size is projected to reach $109.8 billion dollars by 2031 (Straits Research, 2023).

Gamification examples in the mining industry

The mining industry has also slowly started to adopt gamification strategies, but these are far from common. Three examples of gamification in the mining industry will be provided that use different aspects of gamification in their strategy.

The first example of gamification in the mining industry uses an added gamified element in day-to-day work. Workers' accomplishments were recorded in a common room so everyone could see how many points they accumulated with certain tasks (Dessureault, 2019). In this American mine where fueling times were recorded a 30% improvement could be seen after the adoption of a leaderboard-based gamified strategy. By simply adding a game element to their routine jobs, results improved massively, both through awareness and competitiveness.

The second example in the mining industry uses a digital game as a training method. Harry's hard choices is an informative desktop game that teaches mine emergency response. Pilot tests indicate high levels of user satisfaction and engagement and anecdotal evidence of good training transfer (Brown & Poulton, 2018).

The third example uses an analog game as a training method. A table-top card game was developed called Resource, Respond, Rescue (Wilson et al., 2020) to facilitate higher order thinking skills for hazard recognition and mitigation. Play testing was conducted over a 10-month period with 211 learners in the mining industry. Results indicated a high level of engagement, with 95% of users able to apply better controls than the minimums to mitigate hazards.

It can be seen that some gamification strategies are being adopted into the mining industry, but the mining industry is still lagging behind compared to other industries in this aspect. This provides an opportunity for the mining industry to tap into unused potential.

Gamification user statistics

Gamification is a strategy that could vastly improve worker performance. In 2019, Apostolopoulos (2019) conducted a survey on gamification in the work place. According to that survey 89% of employees think they would be more productive at work if it were more game-like. 88% of respondents said that gamification makes them happier at work. Game elements at work made 87% of employees feel more socially connected. These numbers all speak to the potential of the method and the things it could accomplish in terms of employee satisfaction and productivity. According to Georgiev (2023), 72% of employees said that gamification motivates them to work harder. Maske (2019) identified that gamification can increase company productivity by up to 50% and employee engagement by 60%. Research done by Heimburger et al. (2019) concluded that 86% of young job applicants think that gamification at work is fun and enjoyable, 71% of employees believe that gamification leads to an increase in energy levels, and 66% of employees say that gamification at work reduced their stress levels (Sharma, 2020). According to statistics performed by the Finances Online (Chang, 2023), 80% of US workers believe game-based learning is more engaging. 60% increase in employee engagement as a result of gamification features. 50% improved productivity because of gamification training features. It can be seen that the adaptation of gamification can bring massive benefits to employee well-being, engagement, involvement, motivation, and productivity. All these factors are massively important, as 50% of business transformation programs fail due to poor employee engagement (Chang, 2023).

Apostolopoulos (2019) additionally surveyed preferred gamification strategies of adult learners, and these included progressing to different levels (30%), getting points/scores for tasks (27%), receiving real-time performance feedback (26%), progress bars (25%), activity feeds (24%), and competing with friends or colleagues (13%). When asked about specifics on what participants in the survey would like most to be gamified, training software (33%), communication software (30%), contact software (15%), project management software (12%), and document software (10%) were the leading answers. It should be noted that gamified training software is most greatly desired as a gamification strategy, ideally combined with getting points or a score for performing tasks. These strategies were both applied in the development of the haul truck training software.

Gamification in education

As it was desired that a gamified training software were to be developed for haul truck drivers, gamification in the mining industry as well as other industries was studied. It was deemed equally important however to study the known effects of gamification in education as a gamified training software would be an educational technique applied in the mining industry. Legaki et al. (2020) has studied the effects of gamified learning in education. The experiment consisted 365 students divided into four different groups that all received different learning methods to prepare them for a test. Group Control only learned from classical lectures, Group Read read a scientific paper, Group Play experienced a gamified application, and Group Read&Play both read the respective paper and used the gamified application. When undergoing classical lectures, students would score an average score of 36.13 out of 100. Group Read scored 46.13, Group Play scored 52.55 and Group Read&Play scored 58.05. A great improvement can thus be seen from adopting a gamification strategy to achieve high knowledge retention, even more so when combined with other strategies as opposed to classical lectures. Adopting the gamification strategy even outperformed reading the scientific paper on the matter, which might not feel as intuitive. Yet again, the potential of gamification strategies cannot be stressed enough.

Statistical analysis on haul truck accidents

When deciding on the specific content for the training software an analysis was made on haul truck accidents. The objective was to find out in which areas accidents were most likely to occur and what were the causes leading up to those accidents in order for these causes to be included properly in the training software. The Mine Safety and Health Administration (MSHA) data collection (Mine Safety and Health Administration [MSHA], 2022), converted by the National Institute for Occupational Safety and Health (NIOSH) into SPSS format was used to get better insight into haul truck accidents. The total dataset contained 169,536 individual reported accidents and near-misses in the mining industry during a period ranging from 1983 until 2021. These cases were narrowed down by filtering for haul trucks specifically. Other filters applied to the dataset included the terms broke-, accident, fail-, damage, colli-, problem, issue and malfunction. 1,025 cases remained that specifically mentioned both haul trucks and one of the other keywords involved in the accident. These terms were used to focus the research as much as possible on haul truck driving-related accidents. Note that for some words only part of the word was used as a search term as to include the words broke, broken, fail, failed, failure, collision, collided, colliding and more. This was done because it became apparent after a manual analysis that vocabulary and keywords used when describing accidents varied greatly, and thus search terms were required to be as broad as possible to not exclude certain accidents in the analysis where vocabulary deviated slightly.

The results of these search terms gave a distribution that can be seen in figure 1. As can be seen, there was not an even distribution for these separate terms, with the three most common terms accounting for 75% of the haul truck accidents, being broke-, accident, and fail-.

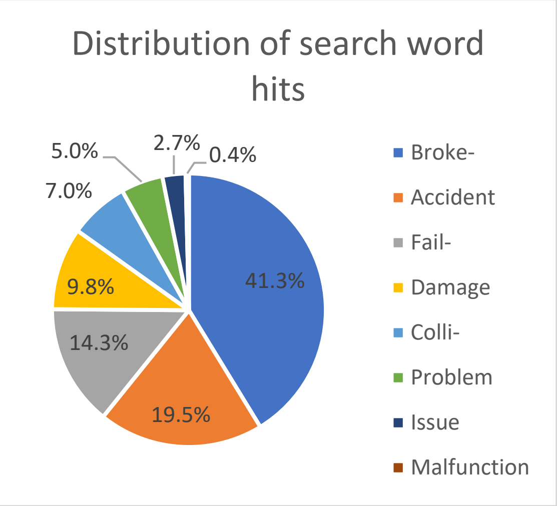


Figure 1: distribution of search word hits

After the dataset was focused on the specific information that was needed, a manual analysis was done of these cases to place them into certain categories to determine what went wrong leading up to the accident.

55 cases were found in multiple keyword categories, thus producing duplicates. These duplicates were dealt with, leaving 970 cases in total. After the manual analysis was completed, the distribution from figure 2 could be found. It can be seen that even though 42% of the cases mentioned a haul truck, the accident was in actuality non-haul truck related. Examples could include accidents happening near a haul truck, vehicles driving into a stationary haul truck and others. For the sake of the statistical analysis of haul truck accidents in a mining environment these accidents do not contribute to the information that was sought after. The same can be said for the next two largest portions of the distribution; where someone injured themselves while operating a haul truck, or accidents that happened during maintenance of a haul truck. Accidents where people injured themselves account for small accidents related to driving a haul truck but did not happen as a direct result of driving said haul truck. Examples include moving a hand past a sharp object that was exposed, or not holding the handrail while ascending or descending the truck and slipping. The third

largest section contained accidents that happened during maintenance, which was not relevant either for the specific contents of the training software.

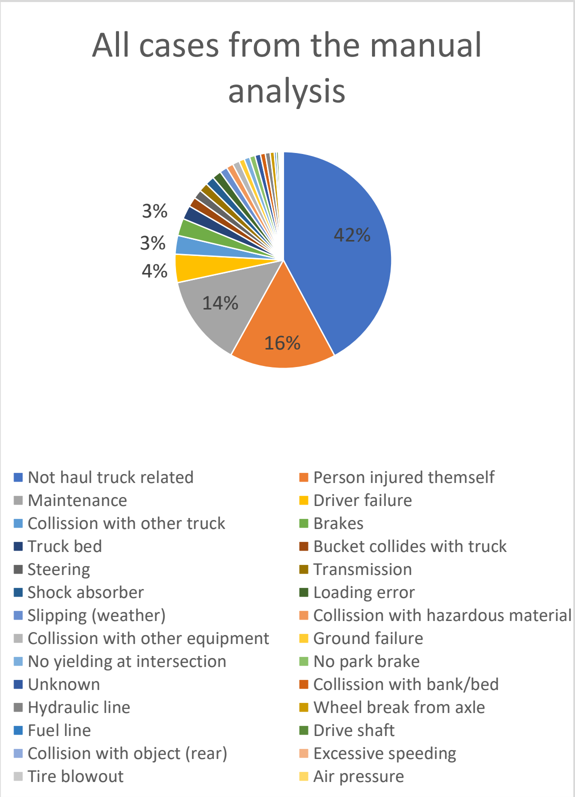


Figure 2: all cases from the manual analysis

After these cases were accounted for, a more clear picture was established of what the content of the training software should include. The training software was intended to handle as many scenarios as possible, without being overwhelming to the user. Not everything could be included so choices were made on what to include based on these statistics.

The following scenarios were decided to be included in the software:

- Driver failure
- Collision with other trucks
- Brake failure
- Steering failure
- Sliding out due to weather
- Collisions with hazardous materials
- Collisions with other equipment
- No yielding at intersections

Driver failure accounts for all cases where nothing necessarily was wrong with the truck, but due to the driver's assessment of the situation, speeding, not following company policies or limited visibility and lack of adaptive driving directly lead to an accident. All these subcategories of driver failure were included in the training software. Adaptive driving is encouraged in all levels both verbally and through level design, as all levels were created in order to be optimally driven when following company policy. Preventing speeding is always enforced as users will receive a penalty for speeding. Limited visibility has an individual level focused on it and making a proper assessment of the situation is trained by completing all scenarios. Examples of driver failure could also include falling asleep while driving, being under the influence of alcohol or drugs, or operating the truck in an incorrect way. These three situations were excluded from the software due to the difficulty to simulate these situations properly. Measures other than simulated training should be taken to account for these situations.
Truck bed failure, the excavator bucket colliding with the truck, transmission failure, shock absorber failure, loading failures, and the lack of applying a parking brake were excluded from the training software for various reasons. Truck bed failure and shock absorber failures were excluded from the software as there is not much a haul truck driver can do when such a failure occurs in terms of adaptive driving, whereas the rest of the software focuses on adaptive driving and dealing with hazardous situations. The excavator bucket colliding with the truck and loading errors were mostly excluded from the software because these accidents were mostly the result of a mistake made by the excavator operators. The software does include positioning the truck as neatly as possible below the excavator arm to make operations for the excavator operator as easy as possible. Finally the choice was made to exclude accidents resulting from drivers not applying a

parking brake because these accidents occur after the shift of a haul truck driver, or when the haul truck driver is on a break. The training software was intended to be made as concise as possible, and therefore extended periods of driving followed by applying the parking brake and leaving the vehicle were not possible due to time constraints.

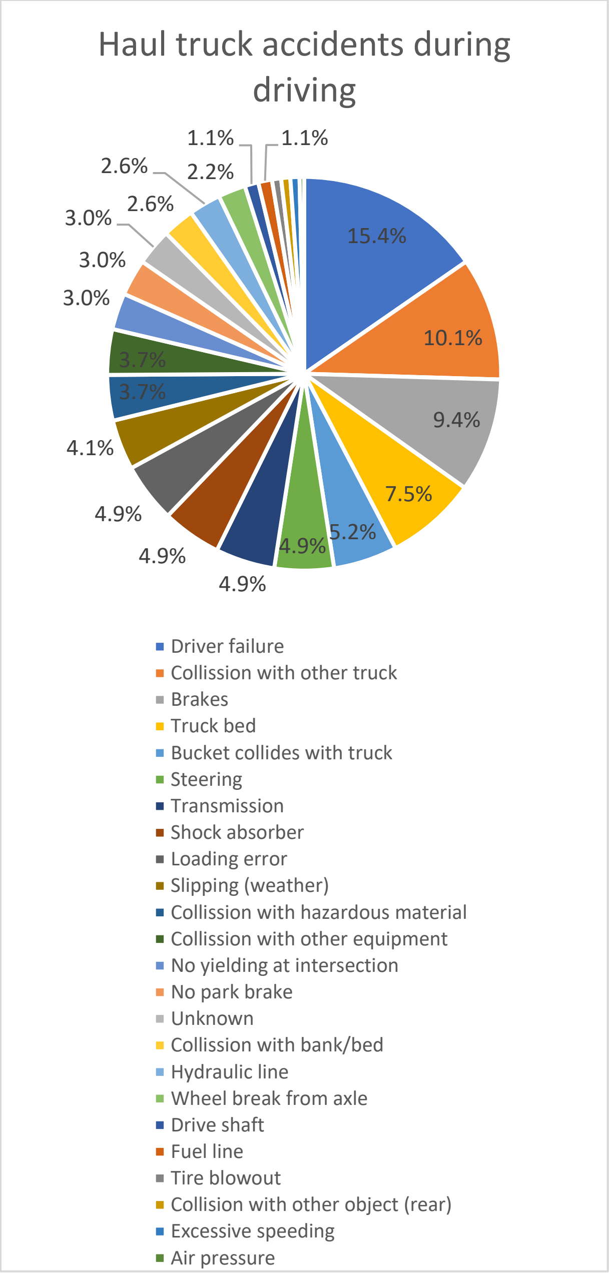


Figure 3: Main cause leading up to the accident

Methodology

The research consisted of three parts: a statistical analysis of haul truck-related accidents, the development of the gamified training software followed by testing of the software by 18 people aged 19-65, and a feedback form answered by these same people.

The statistical analysis on haul truck-related accidents was done by filtering for certain keywords in the MSHA data collection (Mine Safety and Health Administration [MSHA], 2022), and manually grouping accidents in categories. Manual grouping was performed to prevent false positives. Accidents that happened mentioning a haul truck despite not happening while driving a haul truck were excluded from the analysis. The most common haul truck driving-related accidents were chosen to be included in the software.

The training software was developed using the Unreal Engine 5.0 game engine using a free to use license for academic purposes (Epic Games, 2023c). Ease of use and realism of the physics engine were the main priorities when selecting a game engine, and Unreal Engine provided the tools necessary to both support the training software without any major issues while also being fairly easy to use so that a proof of concept could be developed by a single person in a relatively short time.

The Thrustmaster T150 PRO Force Feedback was used as a control mechanism to drive the haul truck. Any inputs were mapped to the pedals and steering wheel to simulate the haul truck controls as closely as possible.

The training software was connected to a 55 inch screen at roughly 2 meters from the users' viewpoint, to provide a good field of view. Users were given an introductory talk on what was to be expected during testing and that a score would be calculated based on their performances. Users were encouraged to try and obtain a high score. Before and during the tutorial, users could ask any question they had regarding the functionality of the software or any other questions they had. Content-related questions during the other levels were not answered as to not influence the results by providing additional information besides the information given to users through the training software. After the completion of each level, users would be permitted to take a short break if they felt it would help them to keep focus for the remaining levels. After all testing had concluded the users were debriefed and any questions could be asked again. All users were asked to fill out the feedback form somewhere later in the day or the day after to ensure some time had passed since completion of the training while also being recent enough to properly reproduce findings and opinions on the training simulation.

Development and deployment

Software layers

The software consists of several layers, the landscape, interactable blueprints, non-interactable blueprints, the user controlled haul truck, the AI driven haul trucks, user widgets, and the levels. All different layers will be discussed in the following sections to provide a clear image of the functionality and extent of the training simulator.

Physics engine

The software made use of standard Unreal Engine 5.0 physics, providing a fully functioning physics engine for the entire development phase. A custom crash mechanism was coded based on calculated forces from the Unreal Engine physics engine.

Landscape

The landscape consists of several individual components, namely the topography, a directional light, a skylight, a volumetric cloud, the sky atmosphere, and the exponential height fog. All these individual components put together created the full landscape layer.

Topography

The topography was constructed by a landscape layer that could be adjusted in Unreal Engine 5 that would provide the solid ground for any object to reside on. It was deemed important for the software to be able to use any given map layout, given that versatility in training environments can enable training in the specific mine the driver would be working at. This would further increase the level of engagement as the drivers will already be familiar with the environment and can thus fully focus on the hazardous scenarios.

A standardized method was developed for importing any mine layout into the training software. Firstly the mine layout was exported into the correct format to create a static mesh of the landscape. A material was constructed that lerps color over a gradient based on the z-value of the absolute world position at any point in the static mesh. When the material was added to the mine layout mesh a height map was created of the entire mine layout. A 2D-render was made of this self-made height map which could be used later by Unreal Engine to make the terrain for the level design. The 2-D render was reintroduced to the Unreal Engine software as a pit shell structure. When creating a new landscape the newly made heightmap was imported from the pit shell structure. As a final step inaccuracies in the pit shell structure were manually edited out, as some erratic spikey behavior emerged as a result of the many transformation steps performed on the original pit shell file.

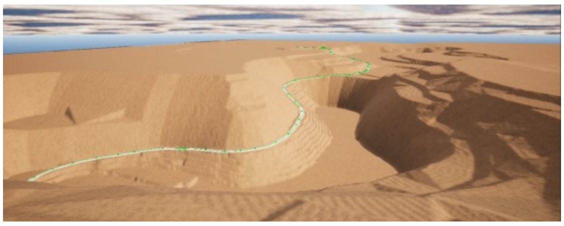


Figure 4: Landscape topography of haul truck simulator

Lighting

The topography was the main component of the landscape layer, but other components were also necessary for creating a realistic environment. A directional light was introduced which would mimic light coming from the sun. This was deemed important for realism purposes as light in real life will also create shading from certain directions where the sun is blocked. It is wired into the human brain to a degree where it is expected at all times to be present, and is even subconsciously used for orientation. Just a directional light would not be satisfactory to provide a real lighting environment, so a skylight was introduced to light the entire scene with omnidirectional light to provide visibility in all situations. In real-life light will be reflected from all surfaces, so an omnidirectional skylight will mimic this behavior.

Sky and cloud simulations

A volumetric cloud was also introduced to create realistic looking clouds. These clouds interact with any of the introduced lights to create a realistic looking sky at all times in order to enable users' ease of submersion. The sky atmosphere that was introduced further improves on this notion as a clear sky also refracts and reflects light in a certain way. Lastly, an exponential height fog was introduced that creates a certain amount of fog based on the absolute vertical location in the level. This ensured that in places of low altitude a more dense fog was present, while in higher places a very thin fog was present that would mimic the atmosphere density when looking into the distance. Additionally, this could provide thick fog when needed for a hazardous scenario.

The three components together in addition to the previously mentioned lighting form the entire sky environment in all of the levels.

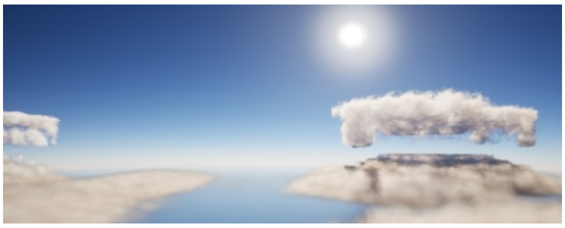


Figure 5: Sky and cloud simulations

Blueprints

Blueprints are standalone sections of code, usually comprised of smaller blocks of code linked together to perform certain tasks. This blueprint visual scripting system is described by Epic Games (2023) as ''…a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. As with many common scripting languages, it is used to define object-oriented (OO) classes or objects in the engine''. A certain amount of blueprints were constructed. Blueprints could be categorized into two main subsets, interactable blueprints and non-interactable blueprints, which refer to whether the player can interact with the blueprints or not.

Figure 6: User-controlled haul truck blueprint

Non-interactable blueprints

Non-interactable blueprints that were constructed included a time counter, traffic cones, haul roads, excavators, traffic signs and buildings. These blueprints were constructed for the sake of functionality of other blueprints or for visual purposes only.

Time counter

A time counter was introduced as a blueprint that uses the tick functionality to track time spent in a level. It was coded that for every second passed from the start of the level, the internal integer would be updated by one in order to measure the time that has passed since the start of a level. This integer value could later on be read again in order to retrieve the time spent on a certain level.

Static meshes for visual purposes

Traffic cones (Sarbassov, 2023), excavators (Dogukan20182, 2021), traffic signs (Studiomf, 2015) and buildings were just introduced as static meshes for visual and orientational purposes for the player. Buildings indicate the location of some offices near the edge of the mine, excavators are usually used as a navigation point to drive towards at most of the levels to be loaded, traffic signs are used to indicate certain traffic rules in the mine and traffic cones were introduced to show designated driving areas at locations where haul roads are not present.

Figure 7: Traffic cone static mesh

Figure 8: Excavator static mesh

Figure 9: Traffic sign static mesh

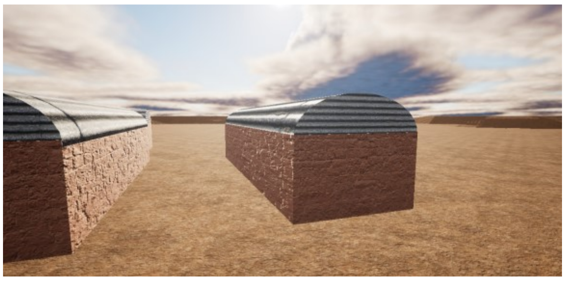


Figure 10: Office static mesh

Haul roads

Haul roads were created by creating a road spline along the desired haul road route and using a 3D road model to follow the created spline to generate the roads in the correct locations. This method was used to make the roads clearly visible to players that were not familiar with the road layout and make the roads more easy to drive on due to their straight surfaces, so the full focus could go to dealing with the hazardous scenarios.

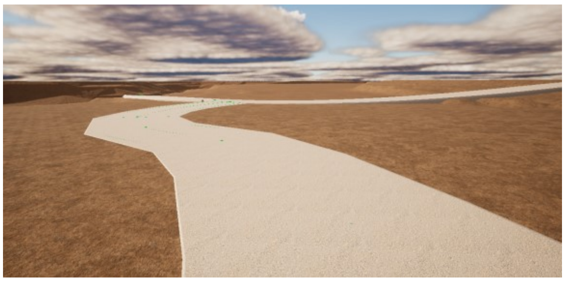


Figure 11: Haul road of haul truck simulator

Road conditions

Ice blueprints were non-interactable in the sense that regardless of player input, the blueprint would always produce the same outputs. A generic box was introduced as a static mesh to the blueprint that was given collision to enable trucks to be driven on top. The blueprint was made invisible to players so players would have to respond to slippery driving conditions without being able to anticipate them. Friction coefficients were altered for the material used in the blueprint to mimic slippery driving conditions. In the training content, icy road blueprints were used to mimic slippery driving conditions after heavy rainfall by adjusting the friction factor.

Interactable blueprints

Interactable blueprints constructed were checkpoints, parking areas, brake failures, steering failures, traffic lights, finish lines, parked trucks, stationary rocks and direction indicators. All of these interactable blueprints were activated upon overlap with the player's haul truck.

Parking areas

Parking area blueprints consist of four corner pieces and a parking icon, making the location of the parking area more visible from further distances. Five box collisions were added to check whether the truck is parked properly at any time. The first box collision is located between the center and the front of the parking areas that will trigger a check whether the truck is parked properly or not. Four other box collisions are situated along the edges of the parking area. When a check is triggered to check for proper parking, overlap between any of these four edge box collisions will cause the check to fail as the truck still has overlap with one or more of the edges of the designated parking area. The truck is supposed to be in the center of the area without touching any of the four box collisions surrounding the parking area blueprint for it to be considered parked properly. Additionally, the check will fail if the truck exceeds 1 km/h; the blueprint will start a loop that will decelerate the truck until it comes to a stop and then considers the park check as successful if the truck is still located in the designated area. Finally, a capsule collision was added with a range far greater than the parking area. Upon overlap with this capsule collision the minimap is disabled and replaced by a park icon to both communicate to the player that they should park somewhere in the vicinity, as well as to ensure that players cannot use the minimap as a helpful tool to make parking easier than it would be in real life. When the check for proper parking is deemed successful a certain variable is adjusted, depending on the level and the function of the designated parking area in said level. The parking blueprint is then disabled so it cannot be triggered multiple times.

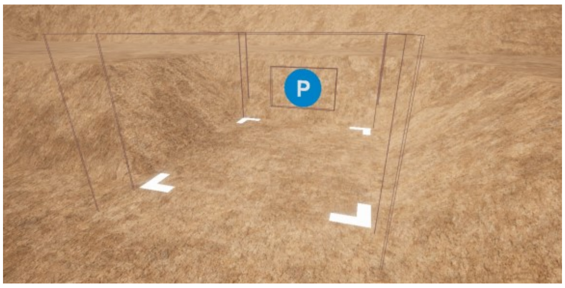


Figure 12: Parking area in haul truck simulator

Equipment malfunction blueprints

The brake failure blueprint is triggered by having the truck overlap with the designated box collision linked to the blueprint. A check is completed whether the truck had already triggered this event before in the level and if so, the blueprint will be disabled. This is done to ensure that leaving the area where brake failure occurs and coming back to it later will not trigger the same brake failure again. If brake failure does get triggered, the blueprint changes a Boolean in the haul truck blueprint which will disable the brake inputs from registering until the brakes are considered repaired again. Additionally, a widget will be shown on the screen letting the player know visually that brake failure has occurred. Steering failure works in exactly the same way as brake failure.

Traffic lights

Traffic light blueprints were constructed by using a 3D-model of a traffic light (Remco, 2022) as a visual representation of the traffic light. The 3D-model was given the correct colors by introducing certain materials to certain parts of the model. Three point light components were introduced at the locations of the traffic lights using the colors red, yellow and green in the correct places. These point light components are activated one at a time using the level's tick, creating a time-based loop resulting in a 20 second red light, a 15 second green light, and a 5 second yellow light. Two different traffic light blueprints were constructed to account for the offset of red and green lights for different roads on the same intersection. As nowhere in the software intersections exist with more than three roads connecting to it, two different blueprints sufficed. The second blueprint uses the same time-based loop as the first one, but in different order to ensure that if blueprint A shows a green or yellow light, blueprint B shows a red light and vice versa. A narrow box collision was introduced orthogonal to the traffic light. When the player overlaps with this box collision, a check is run whether a red light present. If this is the case, the player has run a red light and a widget will be shown to the player that a red light violation was committed. This will also be reflected in the final score.

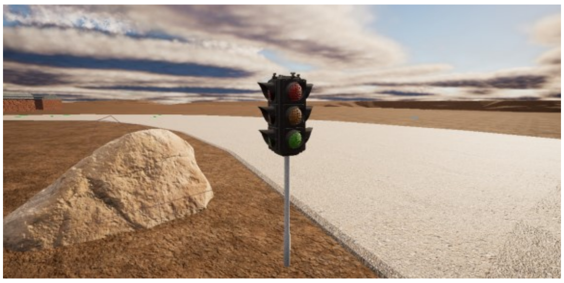


Figure 13: Traffic light blueprint

Finish lines

Finish line blueprints were created to provide a visual cue to indicate the end of a level. When all the objectives in a certain level are completed the finish line blueprint will be activated in terms of visibility and functionality. The word ''Finish'' is projected on a plane of large size at the location of the finish blueprint. Using the tick, the horizontal location of the word respective to the plane it is projected upon will be updated to create a panning effect for additional visibility. A box collision was added to the blueprint for the user's truck to interact with. After all objectives in a certain level are completed the blueprint will, upon overlap with the truck, trigger the finish sequence. Many parameters that were measured will be updated to calculate the score. Firstly, a Boolean for level completion will be flipped to properly indicate to the player in the main menu that the level has been completed. The placeholder values for traffic violations, time elapsed, number of attempts and number of crashes are used to set these values for the current level to calculate the score for the adequate level. All placeholder values are subsequently set to 0. If the level has made use of a Boolean that checks for the level number or an integer that checks for the current checkpoint, these will be set to 0 as well. Finally, the main menu is loaded and put on the screen.

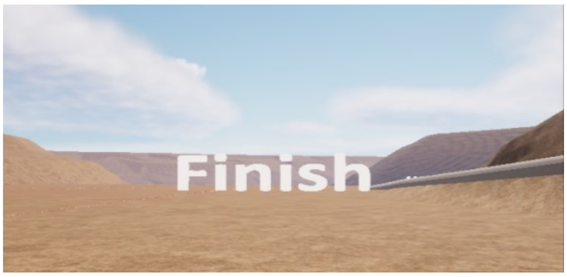


Figure 14: Finish line blueprint

Parked trucks

Parked trucks used in the level were given the same 3D-model as the user-driven truck. A box collision was added tightly around the static mesh. Upon overlap with the user-driven truck a crash will be triggered. It was chosen to trigger a crash regardless of checking the impact of contact because the parked truck blueprints were added in levels that specifically test for control of the truck and punish any collisions with these blueprints by triggering a crash.

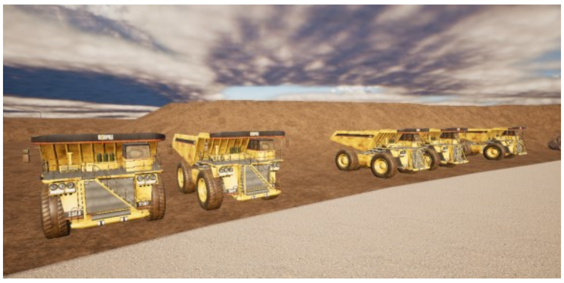


Figure 15: Parked truck blueprints

Stationary rocks

Stationary rocks were given the 3D-model of a rock and the same properties as the parked truck. A tightly fitted capsule collision was added that triggers a crash upon overlap because just as the parked truck blueprints, avoiding these rocks were central to the goal of the level content.

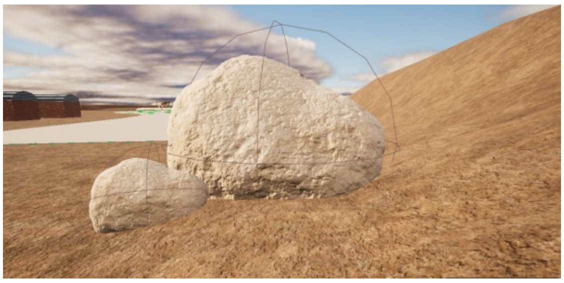


Figure 16: Stationary rocks blueprints

Falling rocks

Falling rocks are visually and practically the same as stationary rocks. Falling rock blueprints were given additional functionality compared to stationary rocks. Firstly, a box collision was added around the static mesh far exceeding the size of the rocks themselves. When the user truck would collide with the box collision, physics would be enabled for these blueprints, meaning that gravitational forces will start to apply to the rocks. Rocks were placed on a hill, meaning that as soon as the user-driven truck would come in contact with the box collision the rocks would fall down. The location and size of the box collision were made such that a truck driving at the speed limit would see the rocks fall down onto the road just in time to slow down and avoid a collision. Whenever the rocks were fallen down, the user would have to navigate around the rocks to proceed the journey. To prevent the user from using the minimap as a tool to drive around the rocks, the minimap would be disabled when the user-driven truck was inside the box collision. A delay of 11 seconds was introduced for disabling the minimap to ensure a disabled minimap would not give away the fact that rocks were about to fall. Instead, the delay ensures rocks can complete their fall, and that users would notice the falling rocks well ahead of a disabled minimap, while at the same time, having the minimap disabled well ahead of being able to drive around the fallen rocks after they have become stationary.

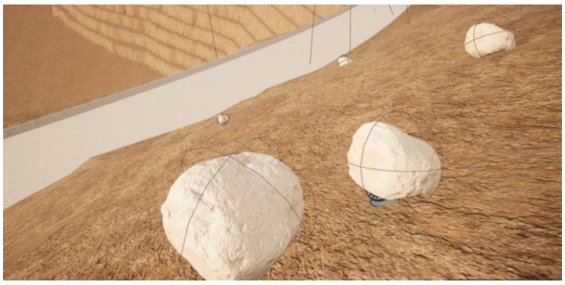


Figure 17: Falling rocks going down a slope

Direction indicators

Direction indicators were added in areas where a turn onto a different road or a transition from gravel haul road to dirt haul road was present. In order for the user to not spend attention on these direction indicators prematurely, a capsule collision was added extending outwards that would enable the visibility of the direction indicator when approaching it. The direction indicator itself was made by having arrows in a certain direction projected onto a plane where the tick would pan the arrows into the direction the arrows were pointing to for added clarity.

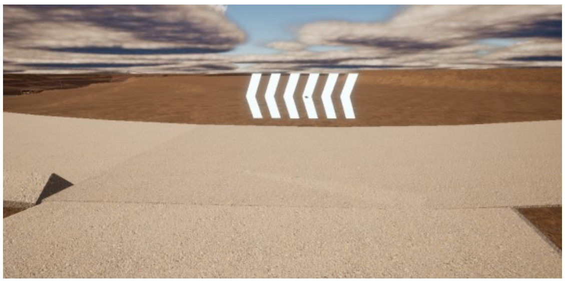


Figure 18: Panning direction indicator blueprint

Checkpoints

Checkpoints were created by having a box collision at certain points in the level where checkpoints would be situated. When the player's haul truck would overlap with the box collision the integer value of the current checkpoint counter within the haul truck blueprint would be adjusted. A placeholder value for the elapsed time would be updated in order to store the time at the point of reaching the checkpoint. The placeholder integer that stores the current number of attempts is preemptively decreased by one to offset the forced start of a new level and the new checkpoint level is loaded in.

User controlled haul truck

The user controlled haul truck is the most complex blueprint in the software and consists of many assets. These assets will be described in the following sections.

Chaos Vehicles Plugin

The chaos vehicles plugin was used as a base system to accurately perform vehicle physics simulations. For the plugin to accurately work, a skeletal mesh, a physics asset, an animation blueprint, a torque curve, and a wheel setup were created.

Truck model

The 3D-model of a haul truck was created by Myagmarsuren (2020), and was introduced in the 3D-modelling software Blender. The UE4 Vehicle Base Rigging addon was used to properly introduce different bones to the model. Bones are needed to subdivide 3D-models into smaller components that can individually rotate and move from each other respectively while still being attached to one another. For optimal performance a balance had to be found between realism and functionality by creating enough bones, while also taking performance into account by not unnecessarily increasing the number of bones. It was decided that a total of 5 bones were to be created as this was deemed to be the least number of necessary bones to meet functionality and realism while also taking engine performance into account. One bone was created for the frame of the vehicle, and four bones were created for the wheels, that would allow for those to move individually from one another and from the frame. Five submodels were created in Blender for each of the bones by selecting each of the planes and vertices making up the model that belong to each bone. The UE4 vehicle Base Rigging addon was then used on each of these submodels in order for the software to recognize these individual parts as bones of a single model. For the functionality of the addon it was vital that the true scale of the 3D-model was applied before determining any bones, as scaling could not be done after this without causing issues with the true locations of the bone joints, and thus the functionality of the vehicle in Unreal Engine. Finally, the Blender model was exported to Unreal Engine using true scaling through FBX format.

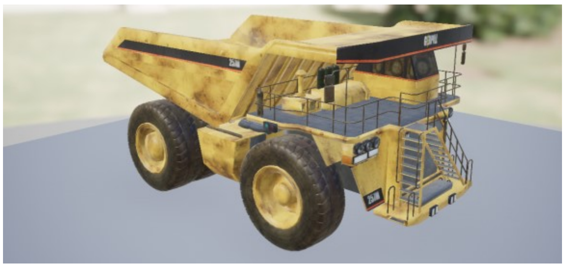


Figure 19: User-controlled haul truck model

Wheel setup

In Unreal Engine 5, the model was imported as a skeleton model. Four wheel blueprints were constructed using the ChaosVehicleWheel blueprint parent class. Two wheel types were set up to distinguish between wheels on the front axle and wheels on the back axle. Rear wheels were not allowed to turn whereas front wheels were. The correct dimensions for either wheel types were set and used to create a manual cylindrical collision mesh which would override the original collision of the wheels from the 3D-model. This would improve smoothness of driving and improve functionality of the software. A maximum steer angle of 36° was introduced, as steering angles of 36-40° are typical for haul trucks of this size. The friction force multiplier, slide slip modifier, slip threshold and skid threshold were determined at 2.0, 1.0, 1000.0, 1000.0 respectively through intensive testing until the behavior of the truck would match real-life behavior. All four wheel blueprints were given a distinctive name to be properly read by the software in later stages.

Physics asset

A physics asset for the haul truck was created by generating sphere collisions for the wheels and a single convex hull collision for the body of the truck that would encapsulate the body closely. In the physics asset the five distinct bones created in Blender were recognized and the wheel bones were renamed in order to match the wheel blueprint names created in an earlier step.

Animation blueprint

An animation blueprint was constructed to animate any movement of the 3D-model properly, meaning in this case that wheels would turn at the correct rate when the vehicle would be driving through the level. The VehicleAnimationInstance was selected as the parent class for the blueprint, and by connecting the Mesh Space Reference Pose to the Wheel Controller to a Component To Local node to the Output Pose the correct animations could be generated for any movement of the mesh due to its compatibility with the chaos vehicles plugin.

Engine torque

A torque curve was created to emulate the torque of a real haul truck engine. The curve was adapted upon further testing to properly reflect the intended behavior of the vehicle in the software.

Game mode

A game mode was created that would default select the haul truck as the pawn that were to be controlled by the player upon the level start to provide a smooth transition when loading in various levels.

Input mapping

As a final step an input mapping context was established and connected to the haul truck blueprint as a collective for storing individual input actions. These individual input actions were made to be triggered by the corresponding real-time inputs from the Thrustmaster T150 PRO Force Feedback. Upon triggering the input actions custom functions within the haul truck blueprint could be activated from where the correct outputs could be coded to create a real-time feedback between the intended driving behavior by the users and the simulated driving output. The haul truck blueprint functions as the control panel for all code in the software as the haul truck functions as a centerpiece in the gamified training simulation, acting as a direct extension of the users themselves. Unreal Engine's blueprint coding enables ease of use, but does not enable any blueprint from directly communicating with one another at any time without the necessary connection nodes. It was therefore deemed beneficial that one blueprint could act as a control panel so as much information could directly communicate with each other at a time as possible. The individual input actions that were coded were a brake, left view, right view, next, retarder, steering and throttle function.

Crash detection mechanism

Additionally, a custom function was created that measures the impact of any collision with the landscape. At any point, the truck is in contact with the landscape by driving over it, but a crash should be initiated when coming into forceful contact with the landscape, like driving into a bench wall at high speeds. The function was altered to only measure the force of impact between the landscape and the body mesh of the truck to ensure that any contact between the two would be unintended and could be punished by the software. With any collision between the haul truck body and the landscape a certain amount of point damage is applied to the body based on the change in velocity upon impact, as higher changes in velocity in less time leads to higher forces on the driver upon impact. When a certain point damage is achieved on any point in the truck body a crash will be insinuated. The threshold for an instant crash was set to be roughly equivalent to a direct crash into a wall at around 30km/h. Additionally, a crash is triggered when the truck is about to topple, even though the point damage through crash impact is not achieved. Through the event tick, a check is run whether the x-axis rotation of the haul truck exceeds 50°, and if this is the case a crash is triggered.

Camera actor

On Event BeginPlay a Camera Actor is spawned and connected to the haul truck mesh so that the imagery on screen will provide a first-person perspective from the driver's point of view from the haul truck cabin. This camera actor will show a real-time image on screen of anything that happens within the scenario to make the training software as real as possible.

AI driven haul trucks

AI driven haul trucks were programmed using the event tick to enable real-time responses to mimic driving behavior as closely as possible.

Steering control

A to the user invisible road spline was drawn along the routes the AI trucks were programmed to drive. During every tick the road spline position would be calculated in relation to each individual AI truck, and the steering input of the truck would be changed accordingly to correct for any deviations from the road spline.

Speed control

For setting the correct speed of the AI trucks a speed limiter macro was created that compares the speed of the AI truck to the speed set by the speed limiter and could yield different outputs based on the outcome of the speed comparison. The first speed limiter macro checks whether the speed of the AI truck is under 20 km/h, and if it is the AI would choose to choose a throttle input. The next speed limiter macro was set to check whether the AI truck would exceed the speed limit or not. If the AI truck would indeed exceed the speed limit the AI would choose to brake with a value that is proportional to the speed excess compared to the speed limit. This ensures that the AI truck would not slam the breaks when only going a couple of decimals over the speed limit and then decide to throttle immediately again in the next tick which would result in visibly shocking behavior. When the AI truck would exceed the speed limit by relatively much, more braking power would be applied than when only exceeding the speed limit by the smallest of margins to smoothly control speed around the speed limit. If the AI truck would not exceed the speed limit of the level, the AI truck would perform a check on whether the truck is driving downhill. If this is the case the AI decides to adjust its speed to 30km/h, mimicking the use of a retarder. To achieve this an additional speed control macro was introduced to brake accordingly. Since the speed limit can change from one moment to another when making the transition from driving straight to driving downhill, the desired speed driven by the AI can suddenly change by a fair margin. To combat erratic braking behavior, the severity of braking would be a result of the currently driven speed, just like the speed limit check. To compensate for large changes in the desired driven speed a quadratic component was introduced to regulate the brake value which lies between 0 and 1 to brake more smoothly until the desired speed is reached. If the truck AI has determined it is either not driving downhill or going under the speed limit for driving downhill, it will check for any other trucks in front of it. Three line traces were introduced as a method of granting the AI some visibility. The first line trace starts at the middle of the front plane of the truck and points orthogonally outward right in front of it. Two other line traces were introduced on the front corners of the AI truck pointed forwards and slightly outward in order to still know what is happening in front of the truck while driving in bends or around corners. These line traces can collide with other AI trucks driving around the level and can let the AI know another truck is driving in front of it and lets the AI know at what distance the other truck is separated from it. If another truck is in front of the AI truck, the truck will brake with a value depending on the distance from the truck in front of it. Finally, the AI will check if it has to make any sharp turn and adjust its speed. Sharp turns in the level were indicated by adding box collisions. When the AI truck would enter such a box collision it would be told in what sort of turn it is situated; a sharp turn or a hairpin. After receiving this information, the final speed comparison will be made. In sharp turns the AI truck will throttle when driving under 20 km/h and brake otherwise, and in hairpins the truck will throttle when driving under 12 km/h and brake otherwise.

User widgets

Widgets are user interfaces that are shown on screen on top of any gameplay. Both adding and removing widgets can be done through code, although it is important to not work with too many widgets at once for the sake of having the software run smoothly. Many different widgets were used in the creation of the software.

Menu widgets

All navigable menus were created through widgets. Buttons were added to the menus in order to progress to the level selection or other options, like showing the statistics, showing the credits, or quitting the software training. The level selection, showing statistics and showing credits would all remove all previous widgets and load their respective widgets. The level selection widget shows a new menu with different buttons for each individual level. Upon loading the widget the widget code will check for each individual level completion, and if certain levels were completed already they would be shown in green. Upon clicking any of the level buttons all widgets would be removed, the selected level will be loaded on screen and the integer denoting the number of attempts for the respective level will be increased by 1. If the level made use of any checkpoints the widget will check for the current checkpoint value and open the respective level belonging to that checkpoint to ensure that the level begins at the most recently achieved checkpoint.

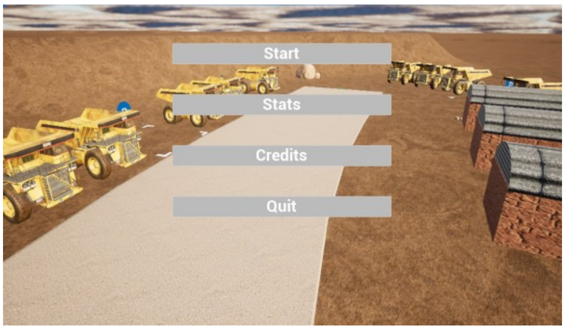


Figure 20: Main menu widget

User statistics widget

By clicking the statistics button in the main menu widget the user can view any achieved scores and other data from previously completed levels. The statistics widget shows 6 columns, consisting of map number, number of attempts, number of crashes, time driven, traffic violations committed and calculated score. These values, except for the score, are calculated from saved data in the game instance. The score is a result of a calculation of these other values. Widgets or blueprints cannot communicate with each other without loading data from one into the other or making specific communication nodes between them. Widgets or blueprints cannot communicate with other widgets or blueprints altogether when they are not loaded in. To combat this limitation, a game instance; a datafile that can always be activated to retrieve data from which coexists in every single level, was made that would hold on to these statistics as well as current placeholder values for these 5 types of data. The game instance also holds on to which levels are completed and the current checkpoint count placeholder.

Rearview mirrors widget

In the levels themselves widgets were present as a visual aid to the users as well. At all times rearview mirrors were shown on the side of the screen made from a material that uses the mirror image of real-time rendering from a camera actor located at each of the rear view mirrors. This ensures that no new widget needed to be created for every new frame containing new information from the back view mirror imagery, but rather the same widget could be used with an automatically updated rendition of the rearview mirrors.

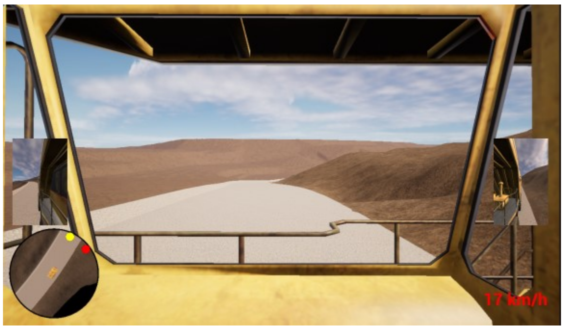


Figure 21: Rearview mirror widgets, minimap widget and speed indicator widget all on screen

Minimap widgets

Another widget that shows on screen during driving is the minimap HUD that shows a zoomed out top-down view of the haul truck with colored dots as indication pointers of the direction of certain important locations in each level. Just like the rearview widget, the minimap HUD was programmed in such a way that no new HUDs would have to be constructed with any updated information, but rather the same HUD could be shown that uses the new information to show a new image. This would greatly improve the optimization of the software. A Material Parameter Collection was used to store the location and rotation of the player. This was done for optimization purposes as the Material Parameter Collection only gets updated when new data is collected and does not spend computational power on overwriting the same value each tick. A custom blueprint interface was introduced to temporarily store the location of the different icons that were to be shown on the minimap. In the HUD code, a custom function was created that would retrieve the relative world location of the things the icons were referring to compared to the player, and broke those values down into vectors. Those vectors were multiplied by the scale of the minimap compared to the true size of the truck on display, and were capped to the edge of the minimap if the size of the vector exceeded the radius of the minimap. The direction and resulting size of the vectors were then used to update the position of the icons relative to the center of the minimap.

During parking and while driving around rocks on the road the minimap was blocked using an overlaying widget that would indicate that either a parking area was nearby or that no data was available in order for the user to not retrieve spatial awareness from the top view provided by the minimap.

Traffic violation widgets

In the event tick of the user driven haul truck the current speed is measured and compared to the speed limit of the level. If at any point the speed limit is exceeded by a margin of 5 km/h a widget would appear on screen for three seconds that a speed limit violation is being committed. This would indicate to the user that the truck were to be slowed down to drive the speed limit again. Whenever this widget is triggered the game instance placeholder value for current traffic violations would be updated by 1.

A similar widget will be shown to the user whenever a red light violation is committed. This is measured by a box collision orthogonal to the traffic light blueprints that will trigger a red light violation whenever the user's haul truck collides with the box collision while a red light is present. Similarly to the speed limit violation, the game instance placeholder value for current traffic violations would be updated by 1.

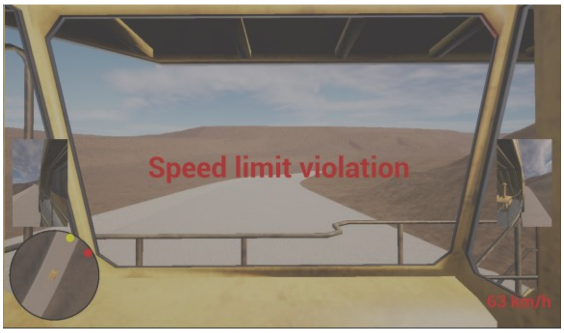


Figure 22: Speed limit violation widget

Equipment malfunction widgets

In certain levels, brake failure or steering failure can occur. When these failures happen, a widget will be shown on screen that either brake or steering failure has occurred. A Boolean in the haul truck blueprint is flipped to indicate that either failure has occurred. The objective at this stage is to get the haul truck to a standstill in order for the situation to be back under control. In the event tick of the haul truck a check will be done whether steering or brake failure is present and then run some checks in order to produce a certain output. A speed control macro is run that checks the speed of the user-controlled haul trucks. When the speed of the truck drops below 1 km/h, a new widget is created that will show a screen with a prompt to contact a mechanic. This widget functions as a clear indicator that the stressful situation of driving a haul truck under equipment malfunction has finished. The user can take their time before proceeding to contact the mechanic when needed, and advance whenever the user is ready.

Figure 23: Prompt widget to contact mechanic

Crash widget

Any time a crash is triggered, a crash widget will be created indicating to the player that a crash has occurred. User inputs will be disabled and the level will continue to run for three seconds. All current user statistics are updated in the game instance blueprint as placeholder values in order to be accessed later. Finally, the widget will be removed and the main menu will be loaded.

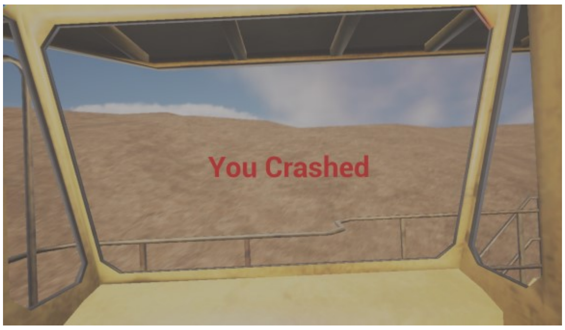

You Crashed

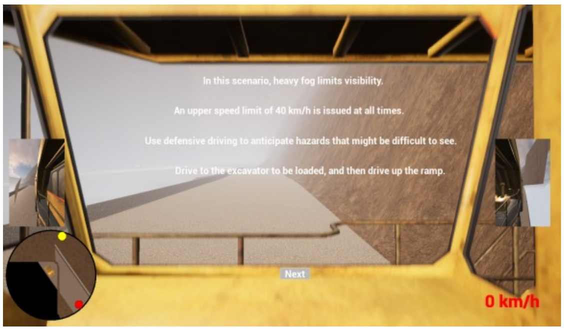
Figure 24: Crash widget

Loading widget

At some instances loading screen widgets were used. The user would see the loading screen as a signal that no actions are required and that certain action is to be expected when the loading has finished. In reality, the software does not require these loading times of a couple of seconds. The loading widgets are merely used when a checkpoint is reached during certain levels, and the user were to be put in a different location for the next section. Changing the position of the user instantaneously was judged to possibly disorient the player. The loading screen here functions as a signal that the previous section was performed successfully, and that a new section is about to commence.

Start widgets

Start widgets were created for each individual level and were loaded right from the start of the level. Start widgets consisted of a certain section of text providing explanation of what was about to happen within the level and what was expected from the user to successfully complete the level. For some levels multiple start widgets were used whenever lengthier explanations were necessary that could not fit within one widget. Widgets could be closed using a button, proceeding to the next widget with further explanation. When all widgets were read the player could use the same button to advance to the driving section where the newly gained information could be put into practice. The tutorial start widgets additionally provided an overview of all the controls of the Thrustmaster T150 PRO Force Feedback in order to prepare the user for the section of driving that would follow.


In this scenario, heavy fog limits visibility.

An upper speed limit of 40 km/h is issued at all times.

Use defensive driving to anticipate hazards that might be difficult to see.

Drive to the excavator to be loaded, and then drive up the ramp.

Figure 25: Start widget for level 4: limited visibility

Widget performance optimization

During development, it was noted that there could be a risk in the way widgets were created and removed by the software. At some moments it was possible for many of the same widgets to stack on top of each other, massively hindering the software's performance on older computers. Unreal Engine's tick functionality is a powerful tool, but when frames are dropped by older computers it could lead to problems by not executing the tick as often as would be expected during development. Especially when frame drops lead to skipping the removal of widgets, many simultaneous widgets could pile up, massively hindering the functionality of the software. A workaround was created to tackle this issue: a custom function was created that accounted for all current widgets that should be shown on screen. Every 5 seconds the custom function would remove any widget that was loaded in the software, and immediately load just the widgets that were needed at that time back in. This would ensure that at the maximum, widgets could only pile up for 5 seconds of time before they would be removed again, thus greatly boosting performance on older, less powerful hardware.

Levels

From a user’s perspective the software consists of a tutorial and 8 different levels. The choice to develop the training software using separate levels rather than a continuous experience was multipurpose. The distinction was used for clarity, allowing the user to focus on distinct predetermined hazards, as per the title of the level. It would also provide a platform for players to immediately see their progress, as with any level’s completion, the user would be one step closer to mastering the training, and completing levels could be a clear motivator to keep advancing into more training. In reality, not 9, but a total of 19 different levels exist. Unreal Engine’s definition of levels is not synonymous with the user oriented progression through different levels by accomplishing certain goals and mastering certain skills. According to Unreal Engine ‘’A Level is all or part of your game's "world". Levels contain everything a player can see and interact with, like environments, usable objects, other characters, and so on. In video games, it's common to have multiple levels with clearly delineated transitions between them’’ (Epic Games, 2023b). In reality, a total of 19 levels were used for developing the software. Certain levels that, from the player’s perspective was 1 level, could in reality be comprised of multiple levels. The main menu for example was designed as a standalone level as it is functionally very different from levels where driving is required, so was therefore also coded very differently and needed its own level to properly function. Also any conventional level that made use of checkpoints actually made use of several different levels, as every individual checkpoint was made into a separate level. Using more separate levels like this instead of just the 9 levels users would be familiar with allowed for more control over the conditions present at the start of each checkpoint, as well as better performance, as some blueprints or widgets would only have to be loaded in for certain sections of the level, and thus could be excluded from certain checkpoints to save on computing power.

Optimization

The software was developed on a 5-year old HP ZBook Studio G5 laptop with an Intel(R) Core(TM) i7-8750H CPU processor at 2.20GHz with 16.0 GB of RAM and NVIDIA Quadro P1000 GPU of 11.9 GB. This hardware was used as an example of older hardware. The goal was to, at all times, be able to run the software smoothly on at least 30 FPS through additional optimization. On this older hardware the software typically runs between 32 and 38 FPS on larger levels, like level 1 and level 8, while running on more FPS on smaller levels. This is enough to execute the software, but a better experienced will be achieved through better hardware.

For comparison the software was also tested on a brand new Alienware Aurora R15 gaming-desktop with a 13thGen Intel(R) Core(TM) i9 13900KF at 3.0GHz to 5.8GHz with Intel Thermal Velocity Boost, 64.0 GB of RAM and a NVIDIA GeForce RTX 4090 GPU of 24GB. The gaming desktop could easily run the software at all times at 120FPS. The experience was smoother, and users reported that the haul truck appeared to have better handling due to the improved frame rates. Differences were easily noticeable and the gaming desktop performed better than the old laptop.

Software content

The software consists of 8 levels and a tutorial. The tutorial was made to make participants that were testing the software familiar with the controls and the workings of the software. A section of mine was to be driven until the excavator was reached where participants were required to line up under the excavator bucket to be loaded. After having their truck loaded, participants were instructed to drive to the finish. During the tutorial participants were motivated to try out all the controls that were taught to them at the start of the tutorial and get familiar with how the vehicle responded to user inputs.

During this phase any questions could be asked if things were not clear immediately, to ensure no mistakes were made as a result of confusion of the controls. During the tutorial no score was kept track of. All the other levels did include a score system.

Level 1 was designed to test the entry level of any participants. Without prior knowledge of what hazardous situations would occur when and how to tackle those situations optimally, participants were requested to drive the level and responds to dangerous situations to the best of their ability. Participants encountered dangerous driving behavior from other haul trucks, rockfall onto the haul road in front of them, steering failure, brake failure, driving in slippery conditions, driving with limited visibility and parking with hazards in blind corners while always following the mine regulations and traffic rules.

Levels 2 up until 7 were each designed around one specific hazardous situation. The participants would get a small text of theoretical knowledge on how to tackle a certain situation and would then be asked to put the knowledge into practice by finishing the level. When the level was not finished due to a crash or some other reason, participants would have to attempt the same level again until the level was completed. This way, participants would have more practice on levels, and thus more experience with those hazards they were least capable of mitigating.

Level 2 was focused on parking and objects in blind spots. Participants had to park 5 consecutive times in varying environments before going to the finish. The first three areas were located in a general parking area, with many parked haul trucks occupying the area. Trucks were parked close together, leaving little space on either side of the designated parking spaces the user had to park in. The final two parking spaces were located on top of a dump hill. The user would have to back the truck to the edge of the dump hill while not colliding with any boulders located on top of the hill to park in the correct spaces.

Level 3 was focused on rock falls and objects on the roads. Participants had to give the right of way to opposing traffic when part of the road was blocked by fallen down boulders. Participants also had to actively respond to a rockfall and come to a stop in a timely manner to avoid the rocks from hitting the truck. After the rockfall had concluded, participants had to drive around the rocks without hitting them before proceeding their journey.

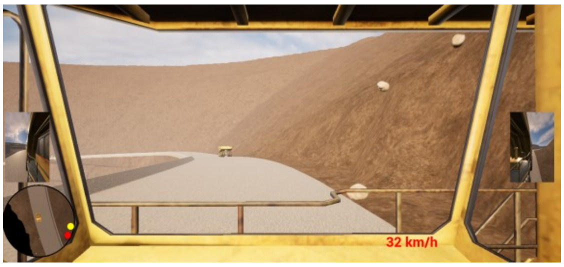


Figure 26: Falling rocks in level 1

Level 4 was focused on limited visibility due to heavy fog. Participants had to drive at a lower speed and focus on their environment despite limited visibility to avoid hitting obstacles. A fog density of 98% was simulated, therefore the overall speed limit was capped to 40 km/h (Lange, 2021) during this scenario.

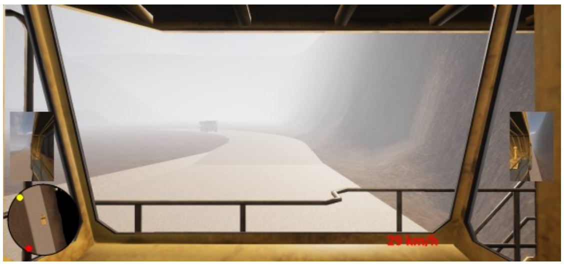


Figure 27: Limited visibility due to fog in level 4

Level 5 was focused on slippery driving conditions due to heavy rainfall. Participants had to adapt their speed to avoid slide outs, and when slide outs did occur participants had to stay in control over their vehicle. The friction coefficient of certain sections of road were altered in such a way that slide-outs would occur when going over 40 km/h. This would incentivize participants to drive the altered speed limit of 40 km/h. Speeding would lead to slide-outs, providing an immediate negative association to speeding.

Level 6 was focused on mistakes made by other haul truck drivers. Participants had to use defensive driving to anticipate mistakes of others and react accordingly. At the first intersection participants encountered, a colleague haul truck was programmed to pass a red light. Participants had to check whether the intersection was free, despite having a green light themselves, in order to prevent a collision. For the remainder of the level colleague haul trucks were programmed to follow the road spline less strictly, resulting in trucks deviating more from their lane, and sometimes showing unexpected behavior, incentivizing users to pay extra attention to other trucks.

Level 7 was focused on equipment malfunctions. Both steering failure and brake failure occurred at two different occasions. In both occasions drivers had to safely bring the truck to a standstill so a mechanic could be called. When steering failure occurred, the truck could be brought to a standstill by simply breaking without much trouble if participants were abiding to the speed limit and responding in time. If participants were late the truck could possibly drive off the road since steering was disabled. When brake failure occurred the road berm or bench wall had to be steered into to prevent the truck from picking up more speed while travelling down the ramp. The difficult part was to find the right angle to hit the barrier as to not crash into the wall or go over the berm while also decreasing speed enough to the point the truck would not further accelerate out of control.

Level 8 was intended to test all previously gained knowledge. The level design was similar to level 1, and everything was made in a way where it was as close in difficulty to level 1. The hazards that occurred in level 1 also occurred in level 8, except the timing, order and location of these hazards were different in the level so participants would not be able to anticipate what would happen when. By making the difficulty similar to level 1, results could be compared between levels 1 and 8 to see if training maps 2 till 7 had an effect on the performance of the participants.

Deployment

The suggested operating procedure for the deployment of the gamified training software first of all requires the correct hardware to be purchased in order to use the controls, where pedals and a steering wheel are a minimum necessity. The software needs to be adapted in a few months time to incorporate the layout of the mine in question, as well as add any specific scenarios based on the needs of the mine and the struggles they encounter during operation. Depending on the level of risk at the site, the gamified training simulation will have to be completed by all haul truck drivers several times per year in order to convert short term knowledge retention into long term knowledge retention and to keep the newly acquired skills at a high level. Haul truck drivers that perform worse during the test can be required to have a shorter period in between training sessions than haul truck drivers that perform very well. After each training session the haul truck drivers can receive an evaluation of their performance by the safety trainer to discuss certain scenarios that came up, or answer any questions or receive any feedback the haul truck drivers may still have.

Results

The software was tested to provide a proof of concept for the possibility and effectiveness of a similar program to train haul truck drivers through gamified learning. There were 18 people that completed the entire training program. Five people were over the age of 30, while 13 people were under the age of 30. The youngest participant was of age 19, while the oldest participant was of age 65. Four people were female, and 14 people were male. All participants were in possession of a driving license, and all but one participant had no experience in mining or prior knowledge of the industry.

The score system

The participants were told that a score would be calculated based on four different criteria that are listed in decreasing order of importance; the most important factor for deciding the score was the number of crashes participants endured before finishing the level. The second most important factor was the amount of attempts people did before finishing the level. Thirdly, any traffic violations like speeding and running a red light were included into the score system. The final and least important contributor to the score was the time it took people to finish the level. A score was used for several reasons. Firstly it's a well understood concept to people that a higher score correlates with a better performance. Secondly a score was used to more easily compare different performances between different participants. If one participant drove faster on average but another participant committed more traffic violations, it might be difficult to compare results against one another. A third reason for introducing a score is that it is a desired gamified element that people prefer to see in a gamified solution (Apostolopoulos, 2019) and can work as a motivator to perform well.

A basic formula for determining the score was made as follows:

$$S = 10{,}000 - \left((A-1)*500 + C*1500 + V*300 + T*2\right)$$

$$S = Score\ (points)$$

$$A = Number\ of\ attempts$$

$$C = Number\ of\ crashes$$

$$V = Number\ of\ traffic\ violations$$

$$T = Time\ (seconds)$$

This formula was constructed to determine a relative importance to several different types of performances to make drawing comparisons between different results easier. It can be noticed that all score components have a negative contribution to the overall score, starting with a base score of 10,000 and deducting points based on these different components.

Component	Abbreviation	Unit	Weight
Score	S	Points	-
Attempts	A	-	(A-1)*500
Crashes	C	-	1500
Violations	V	-	300
Time	T	Seconds	2

Figure 28: score weight system

These values were chosen because the most important thing this learning method teaches is preventing crashes in hazardous scenarios. A large value of 1500 was chosen to ensure that even in a scenario where another person crashed less, but committed more traffic violations and drove slower overall would still be awarded with a better score, since preventing crashes is the absolute most important aspect of the learning method. Each time a crash occurred the level had to be restarted to attempt to reach the finish again without crashing.

For each attempt an additional score of 500 points was deducted. It can thus be noted that each crash results in a total score loss of 2000 points. The score deduction for attempts and crashes was made separate because of the possibility of a participant having to restart a level without a crash ever occurring. This could for example be the effect of the participant losing control over the vehicle and driving off of the haul road and getting stuck somehow. In the formula for calculating the score, it is assumed that levels logically cannot be completed in zero attempts, thus A-1 is used to reflect that at least one attempt is needed.

A score deduction of 300 was chosen for each violation participants made. This factor was deemed to be important, as many rules that can be violated are made in order to bring additional safety in the mine, so following those should be encouraged in training. The score deduction is deliberately chosen significantly lower than crashing, as the consequences of undergoing traffic violations are often lower than those of crashes. It should be noted that all levels were created in a way where any violation would have a hidden additional effect that crashing would be more likely. When running a red light the risk of driving into another haul truck drastically increases, and when speeding loss of control over the vehicle is more easily established. A crash due to committing a traffic violation will thus be the worst offense, as not only mine rules are being disobeyed, the loss of control as a result of that decision is not handled in a way to avoid a crash either. This offense is represented in the score too, as points are deducted for the violation, the crash, the additional needed attempt and the extra time it takes to complete the level after restarting.

Time was chosen to have an influence on the score as well. The score deduction was deliberately chosen to be very low at 2 points per second needed to complete the level. The main reason to include time in the score decision was to disincentivize driving extremely slow just in order to stay in control of the situation at all times and maximize reaction time for any scenario. While it is good to reduce speed in possible hazardous situations, it would not be realistic for participants to constantly drive under half the speed limit just in order to prevent crashes. The weight of time was made small to ensure it does have an impact when abused to the limit, but small enough as to slowing down to assess a hazardous situation more properly would have virtually no impact on the overall score. It also helped to disincentivize to try and drive the speed limit at all times just to get a good time score, as one single crash or speed violation would already ensure the total score would turn out lower as opposed to taking a couple of extra seconds and ensuring the situation can be dealt with in a safe manner instead. It should be noted that a theoretically perfect score would thus be 10,000 since no further points are deducted. However this score is not achievable since time is one of the factors that deducts points from the overall score and levels cannot be completed in zero seconds. After all tests had been completed a new score system was established to make a more intuitive system to people using the final product. The new score system ensured that the relative weight of these components would stay the same to one another, as those were perceived to provide a quite realistic scoring system. All 144 scores were analyzed and the best and worst overall scores were examined. A new score system was scaled so that the worst overall score of all tests would equate to a score of zero, while the best overall score would be equivalent to 10,000. By doing this, a more intuitive score system would be established, where a score of 10,000 would be achievable and people could pursue to get as close as possible to this score in their goal to drive as best as possible. The same thing could be said about a score of zero. When a very low score will be achieved on any of the levels it will make intuitive sense that more practice would be needed in this specific area and possible additional training sessions could be considered. Using this method the new formula to calculate the score will be as follows:

$$Score_{new} = \frac{6{,}994 + Score_{old}}{6{,}994 + 9{,}826}$$

$$S_{new} = \frac{(6{,}994+(10{,}000-((A-1)*500+C*1{,}500+V*300+T*2))}{(6{,}994+9{,}826)} * 10{,}000$$

$$S_{new} = \left(0.416 + 0.595 - \frac{((A-1)*500+C*1{,}500+V*300+T*2)}{16{,}820}\right) * 10{,}000$$

$$S_{new} = 10{,}103 - \frac{((A-1)*500+C*1{,}500+V*300+T*2)}{1.682}$$

$$S = Score\ (points)$$
$$A = Number\ of\ attempts$$
$$C = Number\ of\ crashes$$
$$V = Number\ of\ traffic\ violations$$
$$T = Time\ (seconds)$$

Using this new system a table was constructed with all the newly calculated scores in Appendix A. As can be seen in Appendix A, the system now properly accounts for scores that would have previously yielded negative, and also provide a perfect score that would be realistically achievable, as the lowest and highest scores are 0 and 10,000 respectively.

Overall improvement

For looking at overall improvement the difference was taken in the total score between level 1 and level 8. Figure 29 has all scores sorted based on performances in test 1 from low to high. As can be seen from figure 29, the test results from test 1 range between 0 and 8,992. The test results from test 8 range between 5,317 and 9,577. For both scores a linear trendline was made to visualize the difference in results between test 1 and test 8. People who performed poorly in test 1 can be seen to improve a lot in test 8. It is not so obvious to see if people that performed very well in test 1 improved their scores in test 8. This will be looked into further in the following passage. Differences in performance varied much greater in level 1 than in level 8, suggesting people were brought to a similarly high level by undergoing training. This is backed up when looking at the standard deviation, which is 2,906 for test 1, and 1,153 for test 8. A lower standard deviation for level 8 means that people had more similar scores on this level than opposed to level 1. The average score for test 8 is also higher than that of test 1, with an average of 5,285 for test 1 and 7,917 for test 8. It can thus be concluded that average skill of participants increased by undergoing training and that training showed particularly good progress for underperformers in test 1.

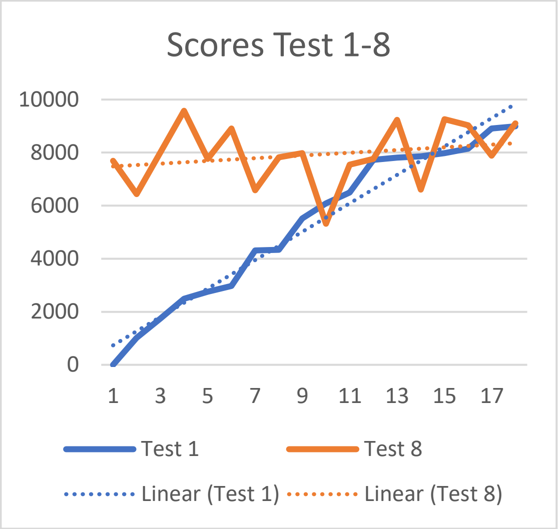


Figure 29: scores for tests 1 and 8, sorted for test 1

When looking at the average scores of the two age groups in figure 30, the group aged under 30 had an average score of 5,242 for test 1 while the age group over 30 had an average score of 5,397 for test 1. For test 8, the group aged under 30 had an average score of 8,151, while the group aged over 30 had an average score of 7,308. It can be seen that the differences in score for either age groups for test 1 are practically the same. There is some difference in score for test 8 for either age groups, with the group under 30 years old on average scored 844 points more. When comparing the performance improvement rates from test 1 until test 8, the group under 30 improved by 999 more points in score on average compared to the group over 30. When looking at the differences between male and female performances, males scored an average of 5,441 for test 1 while females scored an average of 4,741 for test 1. For test 8 males had an average score of 7,863 while females had an average score of 8,105. The male group scored 700 points more on average for test 1, while the females scored 269 points more than the males for test 8. When comparing the performance improvement rates from test 1 until test 8, the female group improved by 941 more points in score compared to the male group.

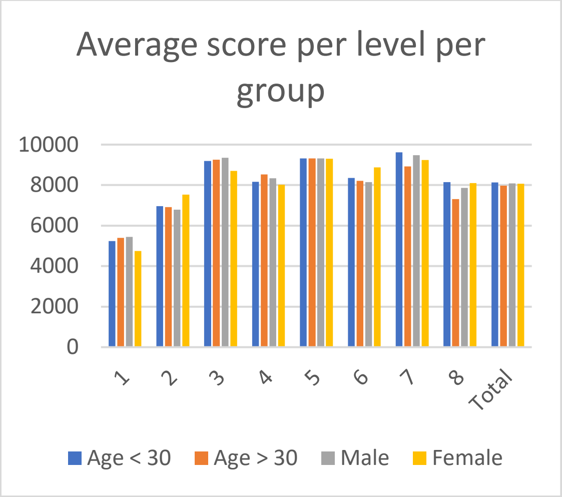


Figure 30: Discrepancies in average score based on age and sex per level

As score is a combined calculation of multiple criteria an analysis of the individual criteria was also done in order to look for certain criteria that were performed particularly good or bad in by certain groups. When taking the average of all levels, people aged over 30 took on average 24% more additional attempts to complete a level than people aged under 30. Females took 17% more additional attempts on average than males. People aged over 30 crashed the haul truck on average 18% more often than people aged under 30. The biggest discrepancies were found when comparing the number of traffic violations committed per group however. The group of ages 30 and up committed by far the least amount of traffic violations, with the group under 30 committing 74% more traffic violations on average. Males committed 34% more traffic violations than people aged over 30, and females committed 120% more traffic violations. There were no significant discrepancies found in average time taken and average score when taking the average of all the different levels.

Individual level results

In figure 31, the average score per level can be seen. The difference between levels one and level eight has been discussed previously as this is the lead indicator of improvement by completing the learning course. From levels 2-7 it can be seen that people had the most difficulties with level 2, whereas level 7 yielded the highest scores, followed closely by levels 5 and 3.

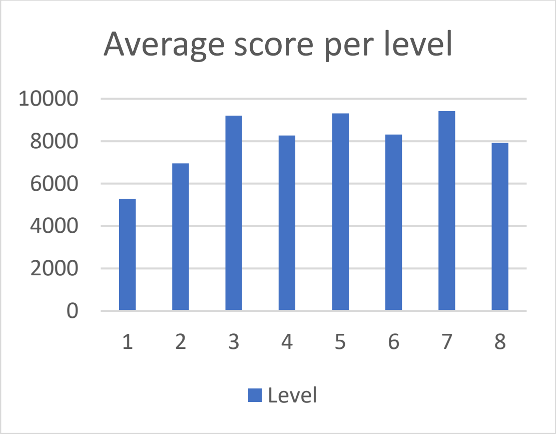


Figure 31: Average score per level

The average number of crashes can also be a leading indicator when assessing the difficulties of the individual levels as minimizing crashes is emphasized most in the software and the scoring system. Level 1 had saw the most amount of crashes, followed by levels 2 and 8 respectively. Level 5 has not seen a single crash throughout the entire test period and can thus be seen as the easiest level to drive safely and keep in control of the vehicle.

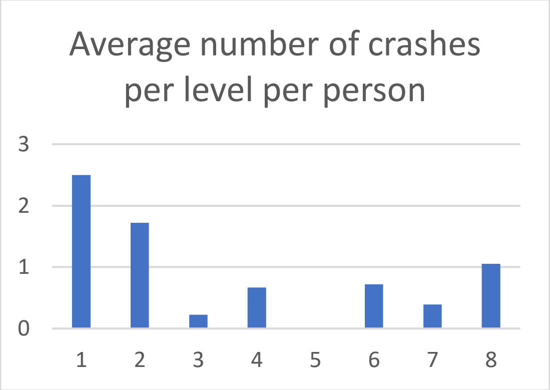


Figure 32: Average number of crashes per level per person

The average number of attempts per level shows strong correlation with the amount of crashes per level as crashes were the main reason for needing additional attempts to completing the level. Resetting due to going off of the haul road and getting stuck was fairly uncommon.

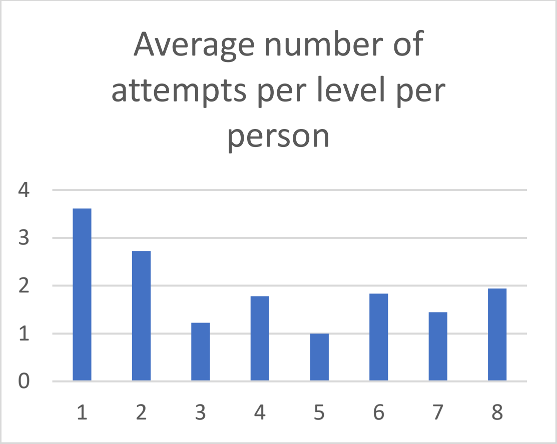


Figure 33: average number of attempts per level per person

The number of traffic violations per level can be seen in figure 34. Level 1 shows the most amount of traffic violations. Despite levels 4 and 5 being viewed as easier compared to other levels, which can be deducted from the amounts of attempts and crashes on those levels, many traffic violations were committed here. This is mostly due to upholding a different speed limit than other levels. In levels 4 and 5 the speed limit is set to 40 km/h rather than 50 km/h, due to limited visibility due to fog and slippery driving conditions due to rainfall.

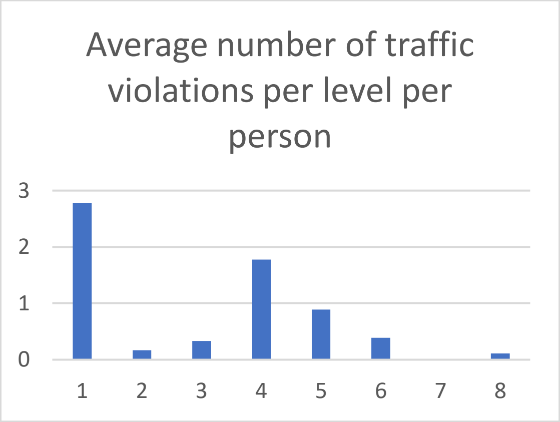


Figure 34: Average number of traffic violations per level per person

The average time per level completion shows some correlation with the number of attempts per level, since more attempts spent per level mean more time in total spent on completing a level as well. Level 7 was the shortest level and also one of the levels with few attempts before completion on average, which explains the little time spent on this level on average. Levels 1 and 8 were the longest levels in terms of time to complete the level, however we can see that more time on average was spent on level 2 than on level 8. This offset can mostly be explained by the fact that level 2 saw 0.66 more crashes on average than level 8, thus resulting in more attempts and thus more time. It was also determined that levels 1 and 8 received checkpoints at several stages during the level, meaning that if any hazardous situation was not handled correctly, the haul truck was set back to just past the previous checkpoint, rather than at the start of the level. Level 2 did not have any checkpoints, so crashes late in the level resulted in a lot of time loss since the level was reset from the start instead.

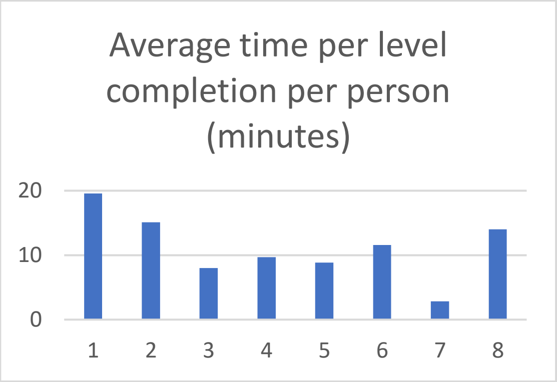


Figure 35: Average time per level completion per person

Response form for participants

All participants were asked to fill in a response form some period after completing the training software to give further insight into user experience as well as some further data. The response form was filled out by 16 of the same participants and the results were as follows. The first question was to list the main hazards that can occur when driving a haul truck in a mine. This question was meant to test knowledge retention through gamified learning as opposed to classical learning in classrooms or reading. The six specific scenarios where the levels were based on were sought after, and the more hazards could be listed, the better the knowledge retention would be. 94% of respondents were able to list at least 3 out of the 6 main hazards that could occur, with the average being 4 hazards listed. Additional things were listed that were not any of the main hazards that could occur, like driving off the haul road, or committing speeding or red light violations. These answers were disregarded for the sake of looking at knowledge retention, but do indicate additional knowledge retention. 94% of respondents were able to list equipment malfunction as a main hazard, while slippery driving conditions were only listed by 50% of respondents.

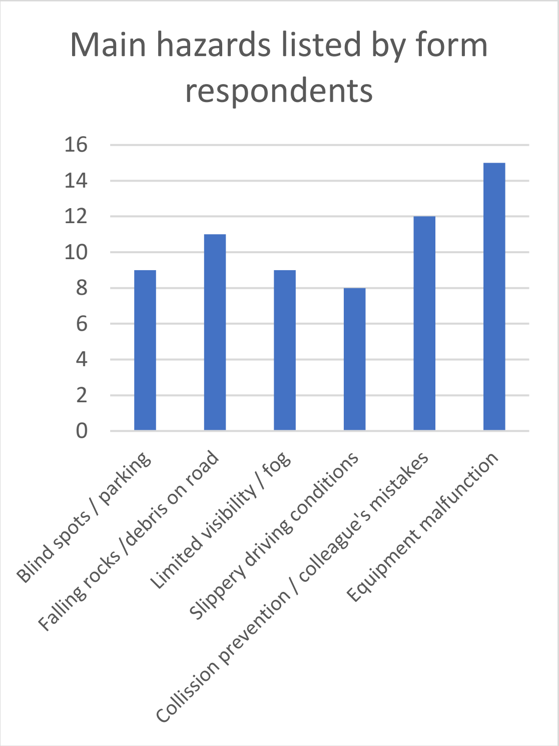


Figure 36: Main hazards listed by form respondents

The respondents were asked to rank the training software based on difficulty. A value of one represents difficult and a value of 5 represents easy. No respondent found the training to be easy or difficult but rather something in between.

	Amount	Percentage	Grade
Difficult	0	0	1
Somewhat difficult	4	25	2
Not easy/not difficult	6	37.5	3
Somewhat easy	6	37.5	4
Easy	0	0	5

Figure 37: Perceived difficulty of the training software by form respondents

The respondents were also asked to rank the software based on clarity to see if improvements would be needed in the communication of information to users in the software. Most respondents found the software to be clear at all times with a smaller portion reporting the software to be somewhat clear.

	Amount	Percentage	Grade
Vague	0	0	1
Somewhat vague	0	0	2
Not vague / not clear	0	0	3
Somewhat clear	6	37.5	4
Clear	10	62.5	5

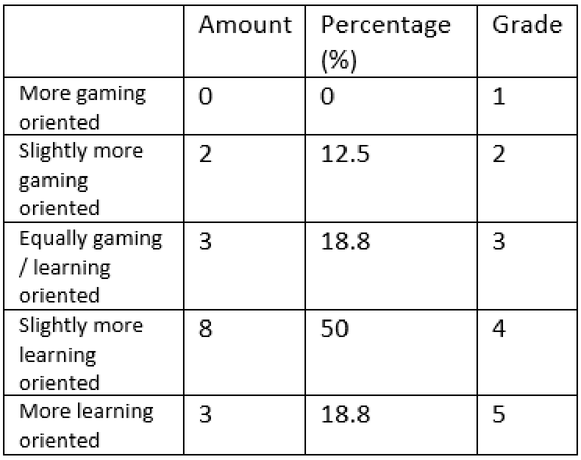
Figure 38: Perceived clarity of the training software by form respondents

Figure 39 shows the perceived level of engagement users experienced while testing the software. The majority of respondents found the software to be decently engaging.

	Amount	Percentage	Grade
Not engaging at all	0	0	1
Slightly engaging	0	0	2
Somewhat engaging	2	12.5	3
Decently engaging	10	62.5	4
Very engaging	4	25	5

Figure 39: Perceived engagement of the training software by form respondents

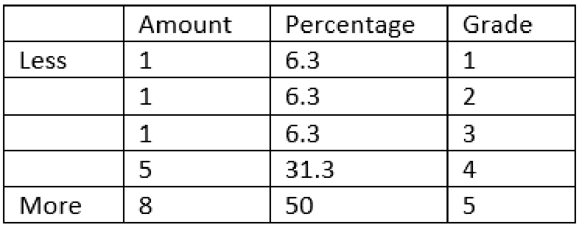
Respondents were also asked to list whether their opinion of the software was more oriented towards gaming, learning, or somewhere in between. Opinions were split, but 50% of the respondents replied that the software felt slightly more learning oriented. No respondent replied that the software felt more gaming oriented.



	Amount	Percentage (%)	Grade
More gaming oriented	0	0	1
Slightly more gaming oriented	2	12.5	2
Equally gaming / learning oriented	3	18.8	3
Slightly more learning oriented	8	50	4
More learning oriented	3	18.8	5

Figure 40: perceived focus on gaming or learning of the training software by form respondents

Finally, the respondents were asked whether they would like to see gamification as a method of learning things be applied more in their own education or career. Respondents were overwhelmingly positive, with only 1 respondent not wanting gamification as a method of learning things in their own education or career.



	Amount	Percentage	Grade
Less	1	6.3	1
	1	6.3	2
	1	6.3	3
	5	31.3	4
More	8	50	5

Figure 41: openness to less or more gamification as a method of learning things in form respondents' own education or career

Discussion

Software content

The scenarios that were decided on were the result of a statistical analysis of the MSHA database. It is difficult to compare the contents of the training simulation to industry training contents, as training contents differ from one mine to another. To form some base of comparison, the training simulation content was compared to the MSHA training guidance (Mine Safety and Health Administration [MSHA], 2021). MSHA's training guidance included objects in blind spots, communication, traffic management, seat belt usage, and dumping practices. Objects in blind spots and parking on dump hills were included in the training software as well. Communication, traffic management and seat belt usage were not included in the training simulation, as communication would require multiple people in the training simulation, traffic management was done in the level design itself, and therefore not of concern of the haul truck drivers, and seatbelt usage could not be enforced as the Thrustmaster T150 PRO Force Feedback did not include a seatbelt. It should be noted however, that the MSHA guidance is not a standard or regulation, but just advisory and a reflection of their current thinking.

Bellanca et al. (2023) created training videos based on near-miss events. A total of four near-misses were turned into an instructional training video, and included driving over a bump in the road, the start of a fire inside of the haul truck, a mechanic car parked in a truck's blind spot, and a collision with another haul truck. These four scenarios included these specific contents to show examples of a near-miss from practice, while then drawing a general lesson from the training video. The gamified training simulation also covered blind spots and a collision with another truck, but did not include driving over a bump, and the haul truck catching on fire. The gamified training software was focused on more severe hazards than driving over a bump. While the software did not include a scenario of the truck catching fire, it did include other scenarios where equipment malfunction had occurred.

Score results

From the results can be seen that a significant improvement can be seen specifically in the underperforming group from level 1. The total average score was 5,285 for test 1 and 7,917 for test 8. When disregarding the top 22% score from level 1, the adjusted averages would become 4,365 for test 1 and 7,660 for test 8 respectively, showing a way lower score for test 1 on average, but only barely lower for test 8. When disregarding the top 50% of scores from level 1, the adjusted averages would become 2,792 for test 1 and 7,864 for test 8 respectively, showing an even lower score for test 1 on average, but a higher score for test 8 than when disregarding only the top 22%. The adjusted average when disregarding the top 50% of test 1 scores is virtually equal to the overall average when all scores were to be included. This implies that all participants were brought to a similar level of skill by the end of the training module. This is further supported by the standard deviation, which was 2,906 for test 1 and 1,153 for test 8. Another explanation could be that the top 50% and even more so the top 22% of performers from test 1 did not have much room to improve in the first place, so it could be the case that this group could see more improvement when the overall difficulty of the levels were to be raised, in order to enable more room of improvement.

	Test 1	Test 8	Improvement
All	5,285	7,917	+2,632
Disregarding top 22%	4,365	7,760	+3,295
Disregarding top 50%	2,792	7,846	+5,072

Figure 42: Table of improvement of underperformers

When looking at the scores with regards to the age of the testers, scores for test 1 were similar for ages under and over 30 years old. For test 8 the age group under 30 scored slightly better on average than the age group over 30, scoring 843 more points on average. While this is not a conclusive difference, there could be a possibility that younger people benefit slightly more from learning through a gamified training simulator. Study shows that workers with less than five years of appropriate mining experience constitute 44% of all fatalities that occurred during the period of 1995–2005 (Kecojevic et al., 2007). If younger people perform better after undergoing the gamified training simulation these numbers could be brought down massively. Females also seemed to perform worse in test 1, but outperformed males in test 8. It could therefore be a possibility that females benefit slightly more from learning through a gamified training simulator. It should be noted that both in the cases of age and sex, differences in performance are not conclusive as the group sizes were small. Additionally, the time of day of testing was different per participant and might also have influenced results on a case to case basis. More testing needs to be done to conclusively reach results, where differences found in this study could provide a hypothesis for a follow-up study.

	Test 1	Test 8	Improvement
All	5,285	7,917	+2,632
Under 30	5,242	8,151	+2,909
Over 30	5,397	7,308	+1,911
Male	5,441	7,863	+2,422
Female	4,741	8,105	+3,364

Figure 43: Table of improvement for age groups and sex

It should also be noted that the weights of the individual criteria were set in a way that would promote certain behavior during driving. Slight adjustments to these weights might lead to improvements with regards to the score more correctly representing the overall performance of the user, as the currently chosen relative weights provide a decent attempt at formulating a score, but these weights might not be optimal. Further research is required to optimize the relative weights of the measured criteria.

Knowledge retention

The participants were never informed during testing that they would have to memorize the main hazards in a mine environment setting. However, an average knowledge retention rate of 66% was observed. When comparing this to the study conducted by Legaki et al. (2020), knowledge retention from classical lectures produced an average score of 36.13 out of 100, whereas the group that underwent reading on the topic as well as a gamified application scored 58.05 out of 100. These numbers seem to be in line with what was found in the study. Interestingly enough, the content of level 5, slippery driving conditions was mentioned the least by only 50% of participants. This was the same level where no crash had occurred for any participant either. It could be that participants did not perceive slippery driving conditions as a main hazard, as this was the level where many participants scored well on. This could stress the importance of a good difficulty balance. The training software should be difficult enough, to the point where participants will be trained properly should such a hazard occur during work, but also to properly make the connection on which aspects of haul truck driving are most hazardous and should be remembered.

Haul truck driver feedback

After testing was concluded on people without prior knowledge of haul truck driving, the training software was presented at Flandersbach mine. A haul truck driver with 40 years of experience in the field underwent some of the same scenarios as the people during the testing had done to provide comments and feedback on the gamified training simulator. A summary of the feedback was given in Appendix B. Testing was done on both the older hardware and the new hardware. After 40 minutes of testing had been completed, 20 minutes of questions and answers followed, along with any additional comments or feedback.

Most of the feedback revolved around the feel of the truck and the handling compared to what the truck driver was used to. During testing on the older hardware, the speed was perceived as ‘slow’, but after switching to the newer hardware this perception was not present anymore. The discrepancy between the two can be explained by frame drops on the older hardware which can make driving feel a bit more sluggish, as less information reaches the eye per second as opposed to the smoother experience on better hardware.

Additionally, the driver was used to a slightly different handling of the haul truck. Braking was reported to be a bit over-sensitive, and the rearview mirrors should have reportedly shown a wider view and be angled more downwards. These points of feedback can be incorporated into a future version of the software. The haul trucks at the mine in question used haul trucks that automatically turned on the retarder when a high rpm was achieved, unlike the manual retarder used in the training software. It was added that during slippery driving conditions, the automatic retarder would not function properly and instead a manual retarder would have to be used. The haul truck driver commented that including this aspect would be a great addition to the content of the software as this was something he was personally experiencing at times.

Despite the differences in handling adaptations to the style of driving were easily made and the scenarios were completed successfully. Additionally, the haul truck driver added that the contents of the different scenarios were realistic and could definitely be something that could be encountered in a real-life situation. When asked about the concept of a gamified training simulator as a means to train haul truck drivers in extraordinarily dangerous situations the response was that the concept was promising and could definitely be used to reach this goal and increase safety in the mine.

Possible improvements

The haul truck training simulator was meant as a proof of concept to show that gamified learning could be a great option for practicing extraordinarily dangerous situations when driving a haul truck. Shortcomings of the software were analyzed and the following findings were done.

The controls of the haul truck could be improved upon. The Thrustmaster T150 PRO Force Feedback did not have a clutch, but this was deemed important to have. While most haul trucks have automatic transmission, driving backwards posed problems at times. It was programmed to use the brake pedal when at a standstill to drive backwards. This allowed for the opportunity for people to make mistakes that would not have happened if the controls were identical to those in real haul trucks, as it could cause confusion for some people. Confusion with controls occurred most in the group of people older than 30, as they had the most driving experience. A control mechanism needs to be made that has to be identical to the control mechanism found in haul trucks to eradicate the possibility of confusion solely because of the controls. This includes virtual reality glasses so the surroundings can be scanned at all times, rather than having to use mechanical buttons that need to be used to look around, which is yet again an opportunity for people to make mistakes they would not have made in a real-life scenario. Additionally, a visual could be made that indicates whether the retarder is turned on, something similar as can be found on the dashboard, when a driver is unsure whether it is turned on or off and wants to check. Sound of the engine could be added so an auditory cue can also be used to indicate the speed of the vehicle to make the simulation as close to reality as possible. For all maps, more checkpoints can be added, so less of the levels need to be redone when another attempt is needed. Not only will this reduce frustration if a particular level is deemed to be difficult, it will also cut down on total time needed to complete the course which would be more efficient. The level of detail for each level can also be increased. A good example of this is the number of rocks that are visible on slopes. If a small number of rocks are visible, users might be able to expect something is going to happen to those rocks, like falling onto the road. If rocks are present on all levels at all times, in all levels, this cannot be a predictor of what is about to happen. Finally, a multiplayer option could be created so haul trucks can simultaneously undergo the training software. AI-driven haul trucks would then instead be driven by colleague drivers, creating a similar environment as during mining operations.

It was apparent from the results that people that did not perform well during test 1 quite some improvement was made at test 8. For people that already performed quite well without any training not much was to be gained. This implies that a difficulty setting could vastly improve the training module. People that are more experienced, or perhaps already undergone the training software once or twice could get the possibility to undergo the training at a more difficult setting to further challenge those people and give them harder challenges to overcome. Finally, the software could be further improved by coding it such that more driving scenarios can be recognized by the software to give specific tips or explanations on what went wrong and how such a situation could be handled better. This would remove the need to have someone knowledgeable present to debrief the participants on what happened and what went wrong. This is an important step to make when making such training software scalable en mass.

In conclusion, possible improvements include:

- Control mechanism identical to that of haul trucks
- Virtual reality glasses
- Design of a widget that shows whether the retarder is on or off
- Engine sounds
- More checkpoints on all maps
- Improved level of detail in each level
- A difficulty setting or overall increased difficulty
- Interactive software that provides tips based on the kind of crash that has occurred
- Multiplayer option for simultaneous training

Conclusion

The research aimed to find the answers to the questions: which are the most important hazards that should be included in a gamified training simulation as practice objectives, can a gamified training simulator provide notable improvements in haul truck driving performance, and is a gamified training simulator an educational and engaging method to improve skills in recognizing and dealing with hazards while driving a haul truck?

Based on the findings from the statistical analysis on haul truck accidents the three main causes leading up to haul truck accidents were driver failure, collision with other trucks and brake malfunction. Through a consideration of total hazard coverage and length of the training software, it was decided that a balance could be found by creating six specified training levels; objects in blind corners, objects on haul roads, limited visibility, slippery driving conditions, collision prevention and equipment malfunction.

Based on the test results it could be concluded that a gamified training simulator can have substantial benefits to the way haul truck drivers can recognize and mitigate hazards through defensive driving. The largest improvement was found for testers that performed the worst during the blank test, meaning that after completing the training course testers were brought to a similarly high level. Younger people and females seemed to perform slightly better after completing the training course, although the difference was not by much.

Both testers of the software as well as the haul truck driver that provided feedback on the gamified training software agreed that the method of learning was realistic, engaging and valuable for reaching the desired goal of recognizing and mitigating hazards in the mine during haul truck driving. From the respondents feedback form it was found that when asked to list the main hazards in the mine, an average knowledge retention rate of 66% was observed. The software was deemed not too easy and not too difficult for inexperienced haul truck drivers, but should be made more difficult for more experienced haul truck drivers. Communication in the software was generally perceived to be clear, and the level of engagement was said to be decently engaging. Half of the respondents agreed that the gamified training software felt more learning oriented as opposed to gaming oriented. Also half of the respondents replied that they would be open to more gamified learning in their personal education or careers.

The optimal experience of the training simulator is on hardware that can easily run the software without any frame drops to make the experience feel more smooth and more fast paced. The proof of concept was deemed successful based on all findings. Changes should be made in terms of brake sensitivity of the truck to make it more realistic as well as the imagery presented through the rearview mirrors.

Further improvements include a better control mechanism to operate the haul truck that is more closely resembling that of a real haul truck. Virtual reality glasses could also be used as a substitute for the screen in order to have a better field of view and an even more realistic experience. More checkpoints could be added to levels as a quality of life improvement, as well as an increase in the level of detail in each map. Another improvement that could be made is the addition of a difficulty setting to have truck drivers complete the training simulation at a difficulty that mimics their experience.

In conclusion, the programmed gamified training simulator can be seen as a proof of concept, and could play a major role in preventing haul truck-related fatalities in open pit mining in the near future.

References

1. Apostolopoulos, A. (2019). The 2019 Gamification at Work survey. *TalentLMS Blog*. https://www.talentlms.com/blog/gamification-survey-results/
2. Bellanca, J. L., MacDonald, B., Navoyski, J., Hrica, J., Orr, T. J., Demich, B., & Hoebbel, C. (2023). Using Near-Miss events to create training videos. *Mining, Metallurgy & Exploration*, *40*(4), 1091–1099. https://doi.org/10.1007/s42461-023-00774-6
3. Boskamp, E. (2023, June 28). *25 Gamification Statistics [2023]: Facts + Trends You Need To Know*. Zippia. https://www.zippia.com/advice/gamification-statistics/
4. Brown, L., & Poulton, M. M. (2018). Improving safety training through gamification: an analysis of gaming attributes and design prototypes. In *Advances in intelligent systems and computing*. https://doi.org/10.1007/978-3-319-94223-0_37
5. Bullock, M. (2023, October 11). *World-Renowned companies using gamification successfully*. Spinify. https://spinify.com/blog/world-renowned-companies-using-gamification-successfully/
6. Chang, J. (2023, August 16). Points, badges, levels, and rewards—these are some of the components of gamification widely used nowadays by o. *Financesonline.com*. https://financesonline.com/gamification-statistics/
7. Cohen, A., & Colligan, M. J. (1998). *Assessing Occupational Safety and Health Training* (No. 98–145). DHHS (NIOSH). Retrieved December 7, 2023, from https://stacks.cdc.gov/view/cdc/11254
8. Dessureault, S. (2019). Rethinking fleet and personnel management in the era of IoT, big data, gamification, and Low-Cost tablet technology. *Mining, Metallurgy & Exploration*, *36*(4), 591–596. https://doi.org/10.1007/s42461-019-0073-7
9. Epic Games. (2023a). *Blueprints Visual Scripting*. docs.unrealengine.com. Retrieved November 14, 2023, from https://docs.unrealengine.com/5.0/en-US/blueprints-visual-scripting-in-unreal-engine/
10. Epic Games. (2023b). *Levels*. docs.unrealengine.com. Retrieved November 18, 2023, from https://docs.unrealengine.com/5.0/en-US/levels-in-unreal-engine/
11. Epic Games. (2023c). *Unreal Engine (UE5) licensing options*. Unreal Engine. Retrieved December 7, 2023, from https://www.unrealengine.com/en-US/license
12. Georgiev, D. (2023, May 20). Top Gamification Statistics of 2023: Next level gaming. *Review42*. https://review42.com/resources/gamification-statistics/
13. Hamari, J. (2014, November 10). *Flow in gamification*. Gamification Research Network. http://gamification-research.org/2014/08/flow/
14. Heimburger, L., Buchweitz, L., Gouveia, R., & Korn, O. (2019). Gamifying Onboarding: How to increase both engagement and integration of new employees. In *Advances in intelligent systems and computing* (pp. 3–14). https://doi.org/10.1007/978-3-030-20145-6_1
15. Kecojevic, V., Komljenovic, D., Groves, W. A., & Radomsky, M. C. (2007). An analysis of equipment-related fatal accidents in U.S. mining operations: 1995–2005. *Safety Science*, *45*(8), 864–874. https://doi.org/10.1016/j.ssci.2006.08.024
16. Lange, A. (2021, April 21). *How does fog affect mining?* weatherzone.com. Retrieved December 17, 2023, from https://www.weatherzone.com.au/news/how-does-fog-affect-mining/533967
17. Legaki, N. Z., Xi, N., Hamari, J., Karpouzis, K., & Assimakopoulos, V. (2020). The effect of challenge-based gamification on learning: An experiment in the context of statistics education. *International Journal of Human-Computer Studies*, *144*, 102496. https://doi.org/10.1016/j.ijhcs.2020.102496

18. Maske, P. (2020, July 28). *Benefits of gamification in training - Paradiso LMS gives you some ideas!* Paradiso eLearning Blog. https://www.paradisosolutions.com/blog/benefits-of-gamification-in-training/
19. Mine Safety and Health Administration [MSHA]. (2022, October). *MSHA Data File Downloads*. Centers for Disease Control and Prevention. Retrieved October 9, 2023, from https://www.cdc.gov/niosh/mining/data/default.html
20. Mine Safety and Health Administration [MSHA]. (2021). PROTECTING MINERS - POWERED HAULAGE EQUIPMENT SAFETY GUIDANCE: MSHA Guidance on Mitigating and Preventing Powered Haulage Equipment Accidents. In *msha.gov*. Retrieved December 19, 2023, from https://www.msha.gov/sites/default/files/events/Powered%20Haulage%20Guidance.pdf
21. National Institute for Occupational Safety and Health (NIOSH). (2023, January 17). *Hierarchy of Controls*. www.cdc.gov. Retrieved November 19, 2023, from https://www.cdc.gov/niosh/topics/hierarchy/default.html
22. Sharma, A. (2021). Back To Basics: How Gamification Can Improve Employee Engagement In 2020. *eLearning Industry*. https://elearningindustry.com/how-gamification-improve-employee-engagement-2020
23. Straits Research. (2023, August 29). Gamification Market Size is projected to reach USD 109.8 billion by 2031, growing at a CAGR of 27.2%: Straits Research. *GlobeNewswire News Room*. Retrieved November 19, 2023, from https://www.globenewswire.com/en/news-release/2023/08/29/2733707/0/en/Gamification-Market-Size-is-projected-to-reach-USD-109-8-billion-by-2031-growing-at-a-CAGR-of-27-2-Straits-Research.html
24. Wilson, L. P., Brown, L., Reed, R. J., & Burgess, J. L. (2020). Gamification of Hazards Recognition in Mining with a Tabletop Card Game. In *Advances in intelligent systems and computing*. https://doi.org/10.1007/978-3-030-51064-0_21

3D-models

1. Dogukan20182. (2021, October 22). *liebherr excavator Free 3D model*. https://www.cgtrader.com/free-3d-models/vehicle/industrial-vehicle/liebherr-excavator
2. Myagmarsuren. (2020, May). *Caterpillar 257M 3D Model*. 3DCADBrowser. https://www.3dcadbrowser.com/3d-model/caterpillar-257m
3. Remco. (2022). *Free C4D 3D Model: Stop Light*. https://www.thepixellab.net/free-c4d-3d-model-stop-light
4. Sarbassov, A. (2023, June). *Traffic cone*. https://sketchfab.com/3d-models/traffic-cone-080079da4e524b39b30fd22579862463
5. studiomf. (2015, March 27). *Traffic Signs 01*. https://www.turbosquid.com/3d-models/free-traffic-signs-3d-model/910452

Appendix A

	Levels							
Participants	1	2	3	4	5	6	7	8
1	2969	8977	7571	6813	9391	9385	9639	8914
2	6501	7844	8696	9241	9074	8272	9639	7548
3	2493	6298	9401	8930	8565	8771	8815	9577
4	8144	7908	9417	6801	9364	8166	8283	9026
5	6083	8273	9656	9392	9535	6883	9995	5317
6	7856	8224	9635	6325	9595	9411	9988	6603
7	8992	6507	9631	9473	9245	5633	9993	9103
8	1740	1201	9818	6990	9552	8226	10000	8002
9	5514	6606	9409	6510	9207	9345	9996	7981
10	7811	7327	7580	9611	9530	8227	9981	9233
11	1018	4870	9616	5998	9505	9171	9979	6438
12	8907	9024	9564	7527	9351	9408	8700	7886
13	7975	9442	9706	9511	9390	8067	9962	9254
14	2750	7471	8188	9390	9310	7528	7310	7767
15	4307	2922	9377	9153	9131	6624	8698	6586
16	4341	8914	9377	8413	9207	8257	8629	7825
17	0	6203	9436	9383	9390	9054	9980	7686
18	7732	7056	9662	9357	9328	9246	9992	7757

Appendix A: score results for all levels (x) and participants (y)

Appendix B

The following passage provides a summary of the feedback provided by a haul truck driver that was asked to test and review the training simulation. The training software was presented at Flandersbach mine. A haul truck driver with 40 years of experience in the field underwent some of the same scenarios as the people during the testing had done to provide comments and feedback on the gamified training simulator. Testing was done on both the older hardware and the new hardware. After 40 minutes of testing had been completed, 20 minutes of questions and answers followed, along with any additional comments or feedback.

The feedback consisted of the following points:

- Scenarios ending in a bad way could be acceptable as a learning method
- On the old hardware the driving felt too slow, on the new hardware this problem did not occur
- It was commented that the rearview mirrors should be angled wider and more downwards
- The contents of the individual scenarios felt realistic
- In some scenarios the weight of the truck felt too low when interacting with the surface
- A comment was made on the difference in operating the haul truck in the training software as opposed to the trucks at Flandersbach mine. Haul trucks at Flandersbach could brake when the retarder is on, and do so automatically when reaching high rpm
- During slippery driving conditions the automatic retarder at Flandersbach mine does not work properly; a manual retarder is used in that scenario. The truck driver was especially interested in the addition of such a scenario, since he struggled with this himself
- Truck driver thinks the training software can work for improving safety