Gamifying a Unity-Based Mining Simulation for Mine Safety Training

BY: YIFEI REN

SUPERVISOR:
AARON LANGILLE

SECOND READER:
ALISON GODWIN

MARCH 30TH 2018

LAURENTIAN UNIVERSITY DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

Acknowledgements

Above all, A thanks to Dr. Aaron Langille and Dr. Alison Godwin, for giving me the opportunity to do this project, and patiently guiding me through this researched based project, keeping me on track and on time.

Secondly, I would also like to thank Stefan Klaassen and all other friends that were able to help me with Unity,helping me to understand the existing project, and been there when I am in need.

Table of Content

Chapter 1 – INTRODUCTION

1.1 Games, VR and Simulations impact learning

The history of video games goes as far back as the early 1950s, when academic computer scientists began designing simple games and simulations as part of their research. Video gaming did not reach mainstream popularity until the 1970s and 1980s, when video arcade games and gaming consoles using joysticks, buttons, and other controllers, along with graphics on computer screens and home computer games were introduced to the general public. Since the 1980s, video gaming has become a popular form of entertainment and a part of modern popular culture in most parts of the world. (Kent, S. L. 2010, Egenfeldt-Nielsen, S., Smith, J. H., & Tosca, S. P. 2015)

Computer and video games are a maturing medium and industry and have caught the attention of scholars across a variety of disciplines. In the Journal article of Educational video game design by Dondlinger (2007) states Computer and video games was ignored by educators. When educators have discussed games, they were focused on the social consequences of game play, and ignored the important educational potentials of gaming, but not any more. Much attention has been directed to the use of video games for learning in recent years, in part due to the large amounts of capital spent on games in the entertainment industry, but also with their ability to captivate player's attention and hold it for lengthy periods of time as players learn to master game complexities and accomplish objectives. This review of the literature on video game research focuses on publications analyzing educational game design, namely those that present design elements conducive to learning, the theoretical underpinnings of game design, and learning outcomes from video game play. (Dondlinger, 2007)

From later studies that proved Games are an effective and cost-saving method in education and training. Although from the early studies of Kriz et al. (2006) much is known about games and learning in general, little is known about what components of these games (i.e., game attributes) influence learning outcomes. Creating effective learning environments plays an important role in supporting organizational learning, changing individual and social interpretation patterns of reality, developing knowledge and competencies, and changing the technical systems of organizations. Using gaming simulation and the design of simulation games as a design-in-the-small approach that has always been a powerful method and is instrumental in modeling and changing social systems while aiming at their sustainable development. Gaming simulation as an interactive learning environment propels the principles of problem-oriented learning into action and enhances a shift of existing organizational cultures and structures and in this way contributes to the design-in-the-large processes of organizations. The training program for systems competence through gaming simulation demonstrates that interactive design of simulation games supports change processes in the educational organizations. (Kriz et al.2006) Simulation is an innovative way to offer experiential learning using a constructivist framework in a safe, clinically relevant environment. It is a tool that can provide an interactive, interesting experience with concurrent cognitive, affective, and psychomotor components. Reflective learning, demonstrated by thinking-in-action, thinking-on-action, and thinking-beyond-action using simulation experiences, can be fostered by facilitated debriefing strategies. Debriefing can be structured to enhance student learning and offer opportunities to develop critical thinking, clinical decision making, clinical reasoning, and clinical judgment skills (Dreifuerst and Thomas. 2009).

Furthermore, Prensky (2001) advocated the use of electronic games in teaching, suggesting that its use would be natural for future generations. Gee (2003) also highlighted the potential of video games in learning processes. References to some of the most relevant work and reviews of literature can be found in De Freitas (2006), Habgood (2007), Wastiau, Kearney, and Van den Berghe (2009), and Klopfer, Osterweil, and Salen (2009).

Lastly, virtual reality could improve the virtual experience and further enhance the interaction in both simulations and games to establish the immersion learning effect. Virtual reality (VR) is the use of computer graphics systems in combination with various display and interface devices to provide the effect of immersion in the interactive 3D computer-generated environment. Virtual reality (VR) has also taken a large step forward in recent years, with projects like Oculus Rift and HTC Vive getting significant recent media attention and development funding. Virtual reality is a computer-based environment that consists of dynamic images and sounds and is affected by the actions of the user. Virtual reality can immerse the user more fully into the digital world they're in and add realism to video games or simulations. Virtual training is certainly an effective and growing field. However, the more advanced and immersive simulators can be very expensive. The Northern Centre for Advanced Technology (NORCAT), has a number of different training modules to simulate underground mining tasks. NORCAT currently has 3 different types of simulators for drilling, bolting, and a load/haul/dump 6 truck simulators. Each of these simulators is based on the actual equipment, and the controls are replicas by the appropriate manufacturers. Virtual learning environment (VLE), not only provides rich teaching patterns and teaching contents, but also helps to improve learners' ability of analyzing problems and exploring new concepts. Integrated with immersive, interactive and imaginational advantages, it builds a shareable virtual learning space that can be accessed by all kinds of learners inhabited in the virtual community (Pan, Z., et al. 2006).

1.2 Motivation and Flows improve games

What is Motivation? In 1943, psychologist Abraham Maslow wrote a paper titled “A Theory of Human Motivation,” which proposed a hierarchy of human needs. This is often presented as a pyramid:

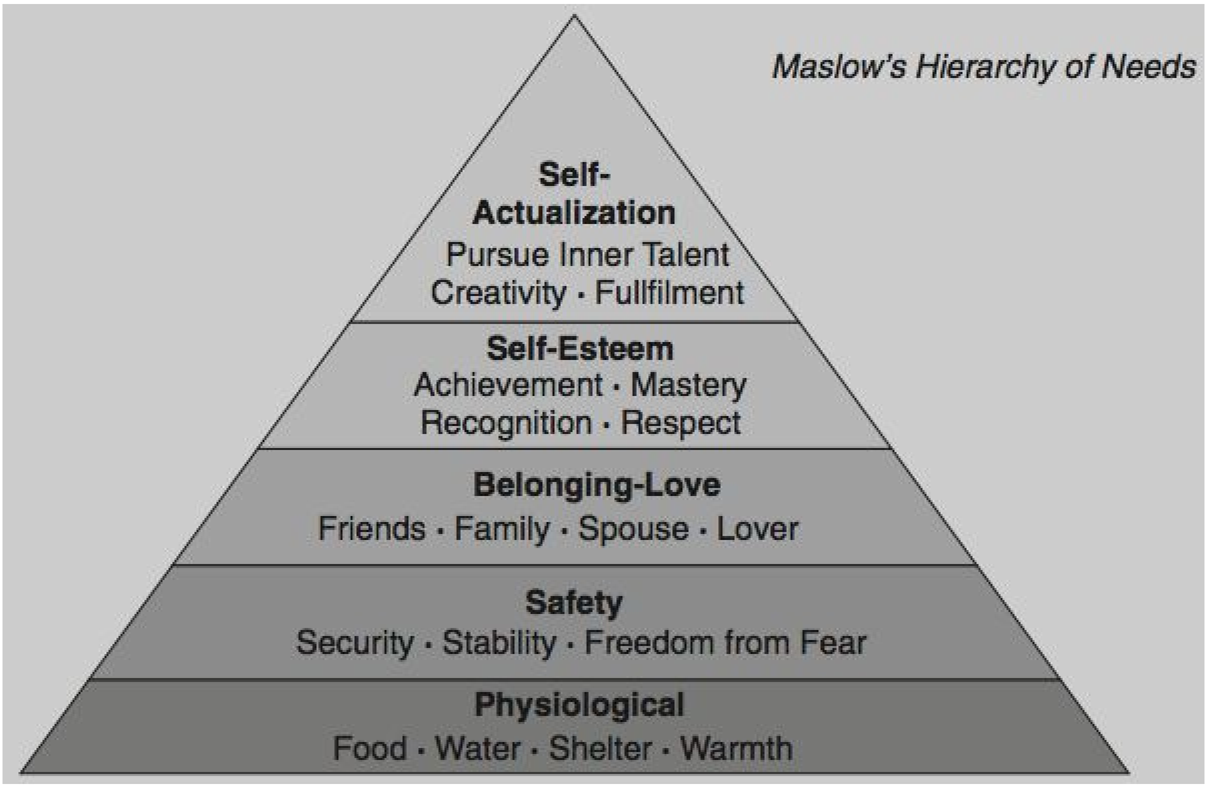

Figure 1: Pyramid of human needs

The idea here is that people are not motivated to pursue the higher level needs on this list until the lower needs are satisfied, same rules applies to games as well.

To think about the player’s true motivations — not just the goals of the game has set forth, but the reason the player wants to achieve those goals by asking questions including; what questions does the game put into the player’s mind, what am I doing to make them care about these questions, and what can I do to make them invent even more questions? For example, a maze-finding video-game might have a time-limit goal such that at each level, players are trying to answer the question: “Can I find my way through this maze in 30 seconds?” A way to make them care more would be to play interesting animations when they solve each maze, so players might also ask the question: “I wonder what the next animation will be?”

Through years the above pyramid has been adapted into the lens of need (Schell, J. 2014):

> *To use the lens of need, designers need to stop thinking about the game, and start thinking about what basic human needs it fulfills. On which levels of Maslow’s hierarchy is the game operating? How to make the game fulfill more basic needs than it already is? On the levels the game is currently operating, how can it fulfill those needs even better? It sounds strange to talk about a game fulfilling basic human needs, but everything that people do is an attempt to fulfill these needs in some way. And keep in mind, some games fulfill needs better than others — your game can’t just promise the need, it must deliver fulfillment of the need. If a player imagines that playing your game is going to make them feel better about themselves, or get to know their friends better, and your game doesn’t deliver on these needs, your player will move on to a game that does. (Schell, J. 2014)*

Furthermore, to help understand motivation in instruction, John M. Keller of Florida State University suggested the ARCS Model of Motivational Design. The ARCS Model identifies four essential strategy components for motivating instruction that encompasses most of the areas of research on human motivation, and a motivational design process that is compatible with typical instructional design models (Paras, 2005, Keller, 1987):

Attention strategies for arousing and sustaining curiosity and interest.

- Learners are more motivated when the instructional design generates curiosity and interest about the content or learning context.

Relevance strategies that link to learners' needs, interests, and motives.

- Learners are more motivated when goals are clearly defined and align with learners' interests.

Confidence strategies that help students develop a positive expectation for successful achievement.

- Learners are more motivated when challenge is balanced in such a way that the learning process is neither too easy as to bore the leaner, or too difficult such that success seems impossible.

Satisfaction strategies that provide extrinsic and intrinsic reinforcement for effort.

- Learners are more motivated when there are rewards for correctly executed actions.

The ARCS Model of motivation was developed in response to a desire to find more effective ways of understanding the major influences on the motivation to learn, and for systematic ways of identifying and solving problems with learning motivation.

Chan and Ahern suggest Csikszentmihalyi's Flow Theory as a method for understanding and implementing motivation. Flow explains a phenomenon that many people find themselves experiencing when they reach a state where there becomes a perfect balance between challenge and frustration, and where the end goal becomes so clear that hindrances fall out of view. The flow theory is a theoretical bridge between the concerns of instructional design and motivational design theory (Paras, 2005). Flow is sometimes defined as "a feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment." The following are some of the key components necessary to create an activity that puts a player into a flow state (Schell, 2014):

● Clear goals: When the goals are clear, the task are easily being focused on. When goals are unclear, players are not "into" the tasks, or not all certain whether the current actions are useful.

● No distractions: Distractions steal focus from our task. No focus, no flow. This means engaging both mind and hands. Menial labor with no thought makes the mind wander; just sitting and thinking can make the hands fidget. These “itchy” feelings are each a kind of distraction.

● Direct feedback: If every time an action is taking place, a wait time is there before knowing what affect the action caused, the focus on the task will quickly become distracted and lost. When feedback is immediate, the task can easily been stay on focused.

● Continuously challenging: Human beings love a challenge. But it must be a challenge that is achievable. If a challenge seems unachievable, it will give a scent of frustration, and the minds will start seeking an activity that is more likely to be rewarding. However, if the challenge is too easy, broadness will take over, and again, the minds start seeking more rewarding activities. Flow activities must manage to stay in the narrow margin of challenge that lies between boredom and frustration, for both of these unpleasant extremes cause our mind to change its focus to a new activity. Csikszentmihalyi calls this margin the “flow channel.”

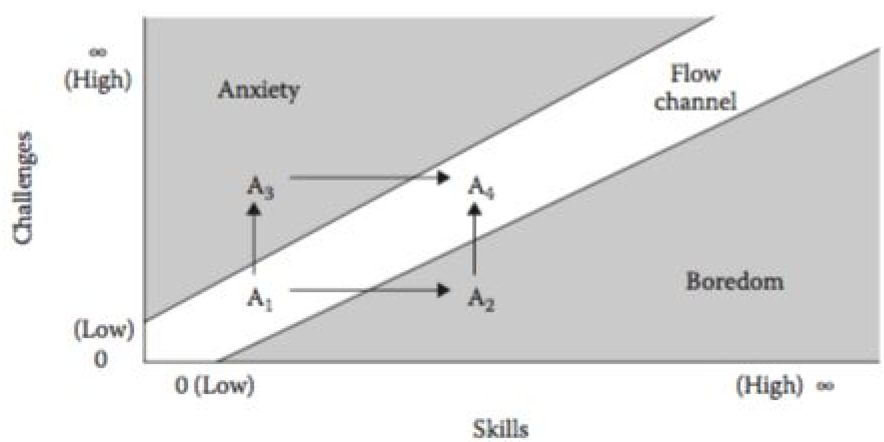

Figure 2: Flow Channel 1.0

Many designers are quick to point out that while staying in the flow channel is important, some ways of moving up the channel are better than others. Including:

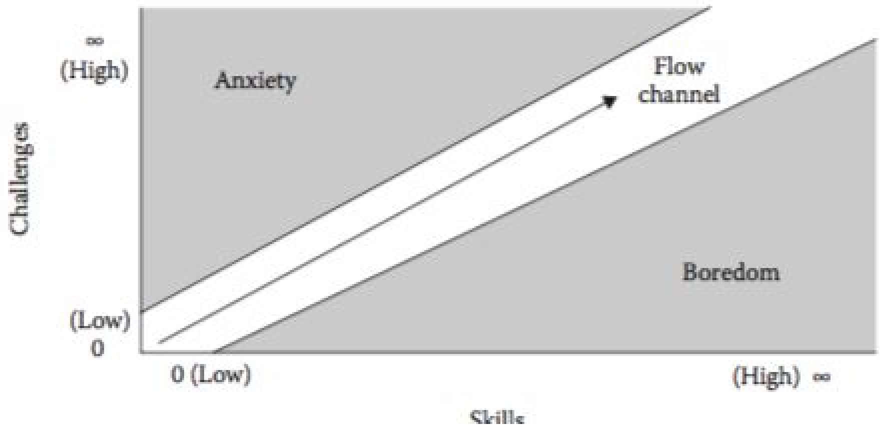

Figure 3: Flow Channel 2.0

It is definitely better than the game ending in anxiety or boredom, but consider the play experience that follows a track more towards this:

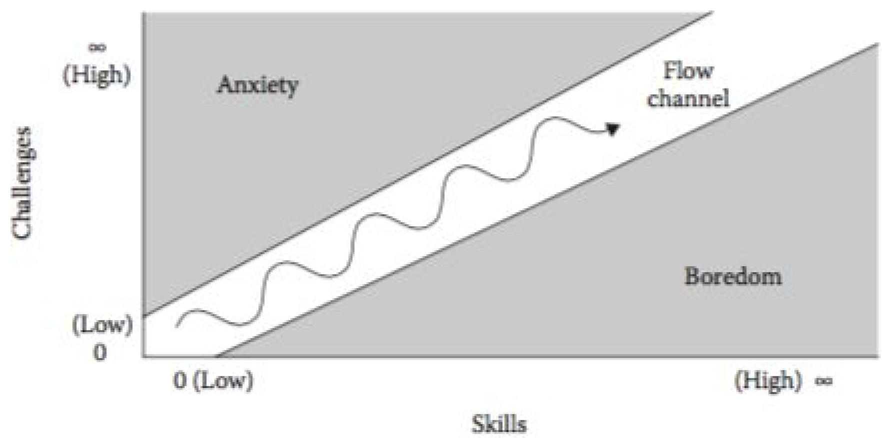

Figure 4: Flow Channel 3.0

This will probably feel much more interesting to a player. It is a repeating cycle of increasing challenge, followed by a reward, often of more power, which gives an easier period of less challenge. Soon enough, the challenge ramps up again. This cycle of “tense and release, tense and release” comes up again and again in design. It seems to be inherent to human enjoyment. Too much tension, and we wear out. Too much relaxation, and we grow bored. When we fluctuate between the two, we enjoy both excitement and relaxation, and this oscillation also provides both the pleasure of variety and the pleasure of anticipation. (Schell, 2014)

Instructional designers can utilize game environments that support flow and enable learning. Norman identifies seven basic requirements of a learning environment (Norman, 2014):

1. Provide a high intensity of interaction and feedback.
2. Have specific goals and established procedures.
3. Motivate.
4. Provide a continual feeling of challenge that is neither so difficult as to create a sense of hopelessness and frustration, nor so easy as to produce boredom.
5. Provide a sense of direct engagement, producing the feeling of directly experiencing the environment, directly working on the task.
6. Provide appropriate tools that fit the user and task so well that they aid and do not distract.
7. Avoid distractions and disruptions that intervene and destroy the subjective experience.

These characteristics of effective learning environments closely match the characteristics inherent in flow-like experiences and motivational design. Houser and Deloach (Paras, 2005) conclude that if we accept Norman's requirements for a learning environment, then we must acknowledge that learning is integrally related to games. Games can make learning look so much like fun that they mask the large amount of learning required to play them successfully, and to further enhance the motivation factor brought by gamification.

1.3 Importance of Game UI display

Furthermore, game environments and feed backs also holds great potential in a way to support gamification and immersive learning experiences. The game's user interface is a system that provides the player with game play relevant information and with the right tools to interact with game. There has been a trend in recent game user interface (UI) design to move system information away from windows, icons and overlays and integrate it into the game-world itself. Along with this trend, the question of whether players prefer interfaces that are integral to the game-world or superimposed onto the screen has become the subject of heated debate in the developer community (Breda, 2008; Fagerholt & Lorentzon, 2009; Wilson, 2006). Those who advocate traditional or superimposed interfaces stress the importance of making the system information explicit and readily visible to the player. The most obvious examples of such interfaces are found in information-heavy types including real time strategy and massively multiplayer online games. The interfaces of these type focus on functionality, with large portions of screen real-estate devoted to clickable buttons, instrument panels or head-up-displays (HUDs) that are clearly separate from the fictional universe.

Proponents of integrated interfaces, alternatively, express concern that any information channel that is not integrated into the fictional world is a threat to player immersion. These designers strive to convey all system information through features that are part of the game-world, including character dialogue, animations or particle effects. This trend is most pronounced in games sporting a first-person view. Peter Jackson's King Kong (Ubisoft Montpellier, 2005) is an example of a game that takes this philosophy to the extreme. Here animation and dialog replaces even the traditional ammo counter and life bar. Games like Crysis (Crytek, 2007), Metroid Prime: Corruption (Retro Studios, 2007) and Assassin's Creed (Ubisoft Montreal, 2007) take a different approach by grounding the HUD in the fiction by making it a part of the avatar's high-tech equipment.

However, Fagerholt & Lorentzon (2009) present a middle ground philosophy. Whenever possible, they say, system information should be integrated as native to the game-world because it allows the player to reason and make in-game choices based on their knowledge of how things work in the physical world. When this approach is unable to present the appropriate information, however, they emphasize that functionality, clarity and consistency are more important than transparency and world integration. But A successful designer of interactive systems know that they can and must go beyond intuitive judgments made hastily and when a design problem emerges. (Shneiderman, B. 2010)

1.4 Use of Head-up-displays (HUD) & Head-down-displays (HDD)

Head-up displays are partially-transparent displays that render information in a manner that allows the viewer to comprehend it while looking into the forward scene. General Motors introduced the first automotive HUD in 1988, and HUDs have since been deployed on a variety of vehicles from a variety of manufacturers, though mostly in the luxury segment or as costly optional upgrades. They typically display only limited, critical information including the speed of the vehicle or the direction of an upcoming turn. However, this limitation has as much to do with the small size and low resolution of current generation HUDs as it does with concerns about driver distraction. In games, the term HUD refers to the method by which information is visually conveyed to the player whilst a game is in progress. The HUD is frequently used to simultaneously display several pieces of information including the main character's health, items, and indicators of game progression and goals. (Fagerholt & Lorentzon, 2009)

Figure 5: HUD example used in Simulations

Figure 6: HUD example used in Games

However, heads-down displays are displays located in the console or machines such that the user must look down to the instrument panel to observe them. It has been established that head-down displays (HDDs), including those commonly placed in the dashboard of commercial automobiles, negatively affect drivers' visual attention. This problem can be exacerbated when screens are "busy" with graphics or rich information (Weinberg, G & Harsham, B & Medenica, Z 2011). The following is a Driving simulator with head-up (HUD) and head-down (HDD) screen positions indicated. A jog dial controller is mounted to the steering wheel.

Figure 7: HUD & HDD example

Chapter 2 - EXISTING MODELS

2.1 Gamification: Game Learn

Recent years have seen a rapid proliferation of mass-market consumer software that takes inspiration from video games. Usually summarized as “gamification”, this trend connects to a sizeable body of existing concepts and research in human computer interaction and game studies, like serious games, pervasive games, alternate reality games, or playful design.

From 2010, Gamification applies elements associated with video games (game mechanics and game dynamics) in non-game applications. It aims to increase people’s engagement and to promote certain behaviors. In 2011, Deterding et al. (2011) defined gamification as the use of game design elements, characteristic for games, in non-game contexts. Although the concept has been explored primarily in the marketing area, the potential of its application has been extended to other areas including Health, Environment, Government or Education. Regarding the game elements, which are used for gamification, social elements are a very remarkable category (Zichermann, 2011), especially to the generation of people who share their everyday in social networks. In fact, various features are common to both social games and gamification: user loyalty, achievements (e.g. rewards programs, points, virtual currency, levels, leaderboards, badges etc.), recruiting users from a user’s social network, etc.

According to Game Learn only 25% of the employees actually complete the online training courses. That means organizations wastes three out of every four US dollars they spent on training. (courses / classes / self-study programs / online training) Actual field trainings are high cost and unsafe for untrained employees. Tradition E-learning does not appeal to students is because it is boring and does not enable experiential learning. To resolve the problem game learning revolution courses and trainings combines three main components Useful content that can directly apply to the job and real life, Gamification is applied to improve engagement and Game based learning simulation is used so student can learn through practice and receive personalized feedback with this method Game-Learn.com has achieved a 94% completion rate and 97% of the students believed what they learned can be applied to their jobs. Game Learn creates a platform that student can access to develop their skills through video games and simulations, providing a real learning adventure. Each game or adventure is designed to develop a variety of skills. The platform covers all the skills you would learn in a standard-training monologue – communication, coaching, flexibility, change management, conflict management, negotiation, innovation, leadership, planning, problem solving, teamwork, safety, and time management. Creating an “addictive” video game that ensure experiential learning. Giving the abilities including monitor users, generate reports, and custom LMS (Learning Management System) can be applied too. It is fast to implement, and easy to manage and turning each skill into a set of techniques, strategies and tools. Furthermore, by using gamification techniques including challenges, rankings or badges to increase the ‘engagement’. Represent “real” situations through advanced simulators to practice the skill and receive feedback.

Game Learn develops game-based training courses to master soft skills. The video games developed by Game Learn use the g-learning methodology, which combines game-based training with gamification elements and an advanced simulation. More than 200 multinational firms across the five continents have already positively impacted their business using our online simulators and video games. An example that Game learn offers to its client is called “Merchant”, a negotiation conflict resolution course. The game creates a competitive environment that everyone in the class competes to be the best negotiator. The ranking system keep the competitive spirit alive at all times. Each user takes control of their avatar and become the protagonist of their own adventure. During the course as the story unfolds, the student learns negotiation techniques and tools from the different characters in the game. Most importantly students put these techniques and tools into practice, solving situations that involves negotiation based on real life scenarios. The negotiation simulation enables students to practice their new skills, try different negotiation strategies, see the results obtained and receive continuous personalized feedbacks on their performance. Above all, student will be able to repeat again and again with no cost or risk until they have mastered the skill they are learning. By the end of the game, students will have covered the content of a 16 hours face to face negotiation course. Student will have access to the game for a period of 4 to 8 weeks, so they can do it when it's best fit them, this gives them time to absorb and consolidate the new knowledge, that is impossible through a face to face course.

2.2 Visualizing safety assessment by integrating the use of game technology

Construction is one of the top unsafe and dangerous industries around the globe. According to study, it is responsible for 76% of all fatal accidents in industry in the region of Hong Kong, it is about 20 times more than other industries. Li et al. (2012) argued the fatal accident rate could be largely reduced by improved production practices in isolation from the project’s physical design. There is a new safety assessment method can be applied to Li’s approach, called the Virtual Safety Assessment System (VSAS). It is described which offers assistance with safety improvement in production practices. This involves individual construction workers being presented with 3D virtual risky scenarios of their project and a variety of possible actions for selection. The method provides an analysis of results, including an assessment of the correctness or otherwise of the user’s selections, contributing to an iterative process of retraining and testing until a satisfactory level of knowledge and skill is achieved.

Li, H., Chan, G., & Skitmore, M. (2012) developed a new construction safety assessment system by using game engine technology. This aims to provide a new means of assessing construction safety knowledge and safety attitudes of construction workers. The system is developed to suit the use of construction workers as they are the frontline of the industry. In traditional industry safety training It is very difficult, if not impossible, for a contractor with 10 to 20 staff to manage the safety of hundreds of workers simultaneously involved in many different activities in many different places. Safety management teams find it more difficult to control and assess the degree of risk for certain trades, as only a limited amount of information is available before the start of construction. Consequently, a hazardous working environment exists, which eventually leads to construction accidents and fatalities. In this case, the construction workers themselves act as the last protection from construction accidents. In result Visualization has become a solution to numerous construction problems in recent years. It is achieved by the use of new technologies, including Building Information Modelling (BIM) and Virtual Reality (VR). The use of visualization for safety has been studied by Chantawit et al. (2005), with a 4D Computer Aided Design (4DCAD) approach developed for safety planning. The use of visualization has successfully improved the effectiveness of construction management.

VIRTUAL SAFETY ASSESSMENT SYSTEM (VSAS) is developed in a game engine Unity 3D, developed by Unity Technologies. The VSAS aims to visualize the causes that have been identified for safety assessment. The visualization process involves a combination of virtual environments and 3D simulations:

- Visualizing unsafe site conditions
- Visualizing unsafe working behavior
- Visualizing unsafe construction methods or sequencing

Users need to study the entire virtual environment carefully in order to select the correct answers for the training questions. Several clues are inserted into the environment to assist users in their answers. Users can walk through the virtual environment and talk with any virtual workers to obtain further information. In a similar manner to that which happens in the real-world, users can also observe the environment from different perspectives. This virtual experience allows users to identify hazards in a virtual, and risk-free, environment. In conclusion, game technology is useful for simulating high risk activities, including in the training and assessment of aircraft pilots and motor vehicle drivers. In this study, a virtual safety training system was successfully developed and evaluated by trials and post-use interviews. The results indicate that VSAS helps pinpoint the weaknesses of users.

The Current simulation already includes the characteristics of the VASA model, to further improve this model, gamification elements, play style, Motivation and Flows will play a greater part in the future gamified model.

2.3 Virtual Reality Training Applications for the Mining Industry

In a mining context, a primary aim of developing virtual environments is to allow mine personnel to practice and experience mine situations, activities and processes that can be encountered in the day to day operations at a mining site. Safe and efficient planning and production are fundamental to profitable mine operations and VR provides an intuitive means of exploring the diverse and disparate information associated with mining processes. (Van Wyk, E., & De Villiers, R. 2009, February)

By meeting with various role players in the mining industry, as well as interviews with mine safety, health and environmental (SHE) representatives, Van Wyk, E., & De Villiers, R.(2009) identified the following training issues:

- The safety of untrained employees in on-the-job training.
- A time constraint due to productivity pressures.
- Classroom-based training traditionally did not provide sufficient interactivity.
- Language barriers.
- Monitoring trainee progress.

Mining in the 21st Century is a high-technology industry that strives to reduce risk and improve safety through the use of improved processes and procedures. This risk reduction process is aligned with the innovative use of technologies often developed for other industries. Interactive computer-based visualization systems and their content are an example.

The work described in Van Wyk, E., & De Villiers, R.'s paper (2009) sets out to add to the development of future VR training systems. It aims to answer questions about the real-life tasks and the context that these systems should support, the functionality of such systems and their levels of realism. The primary research questions are: What are the contextual requirements and constraints for VR training systems for the mining industry? How can realism be enhanced in simulation training? The data collection was done in an integrated fashion, using each data collection instrument or session to address several of the aspects requiring investigation. There are 6 methods used for this process. Furthermore, from observations, the underground work environment can be described as dirty, dark, wet, noisy, hot, uncomfortable and dangerous. Hazards related to the work environment are Working in confined area, working in steeply inclined excavations, handling heavy material and equipment and Working in the proximity of moving machinery.

Latter, it emerged from interviews with mine managers that improved and technologically advanced training systems are required to assist them to improve the safety records at their mines. With the findings regarding the contextual requirements and constraints for VR training systems, as well as analysis of previous accident statistics, various areas were selected for the development of VR prototypes. The prototypes simulate the generic hazard awareness in conventional mining, causes of ground falls, reconstructions of previous fatal accidents, pedestrian and driver hazards regarding trackless moving machinery and lastly hazards related to smelting plant safety.

Figure 8: Example of Working Environment (Van Wyk, E., & De Villiers, R. 2009)

Figure 9: Example of Questionnaire (Van Wyk, E., & De Villiers, R. 2009)

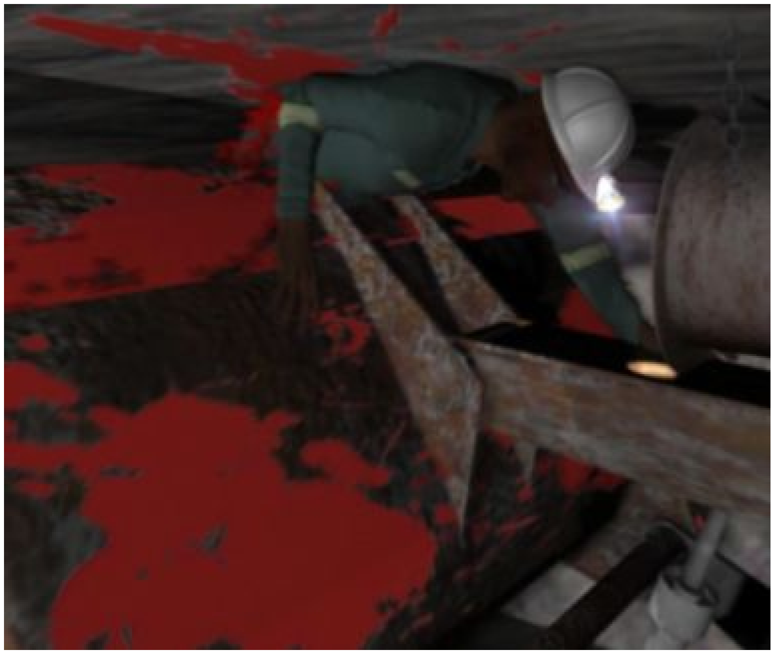

Figure 10: Example of Working Accident (Van Wyk, E., & De Villiers, R. 2009)

This model is enhancing on realism. It is essential to capitalize on the wide-ranging capabilities of computing in order to maximize realism within training systems. This contributes to the authenticity and impact of the simulated trainee experience. In mining prototypes, realism can be addressed on three levels:

- Realism in computer-generated imagery (CGI)
- Realism in content
- Realism in experience

The various VR prototype systems were installed and used at mines and smelting plants. They were all well received and provided an interesting and engaging alternative to conventional training programs. Simultaneously, the approach improved the safety culture and awareness of the workforce.

Chapter 3 – From Simulation To Game

3.1 Difference between Simulations & Games

A simulation is any attempt to mimic a real or imaginary environment or system (Alessi & Trollip, 1991; Reigeluth & Schwartz, 1989; Thurman, 1993). A simulation usually serves one of two purposes: scientific or educational. Although Rieber, L. P. (1996) states, a simulation may be designed as an expandable simple case of a system that appropriately matches a learner's prior knowledge and experiences, this, in and of itself does not satisfy the requirements of self-regulated learning. The learner may not be interested in choosing initially to participate in the activity or may not choose to persist in the activity for extended periods of time at a meaningful level. The learner must find the activity to be intrinsically motivating. Gredler, M. E. (2004) Briefly state, Games are competitive exercises in which the objective is to win and players must apply subject matter or other relevant knowledge in an effort to advance in the exercise and win. Simulations, in contrast, are open-ended evolving situations with many interacting variables. The goal for all participants is to each take a particular role, address the issues, threats, or problems that arise in the situation, and experience the effects of their decisions.

Academic games may fulfill any of four purposes:

1. to practice and/or refine already-acquired knowledge and skills,
2. to identify gaps or weaknesses in knowledge or skills,
3. to serve as a summation or review, and
4. to develop new relationships among concepts and principles.

Well-designed games are challenging and interesting for the players while, at the same time, requiring the application of particular knowledge or skills. Same terminology was described by Schell, J. (2014) as Lens of Flow, Lens of Balance and Lens of Challenge. Five design criteria that are important in meeting this requirement in Game design:

1. Winning should be based only on the demonstration of knowledge or skills. When chance factors contribute to winning, the knowledge and, effort of other players are devalued. But Schell, J. (2014) may argue Chance is one of the well-used mechanic in games. It is normally dealt last because it concerns interactions between all of the other six mechanics: space, time, objects, actions, rules, and skills. Chance is an essential part of a fun game because chance means uncertainty, and uncertainty means surprises. And as discussed earlier, surprises are an important source of human pleasure and the secret ingredient of fun. Simultaneously designers should ask themselves, are players here to be judged (skill) or to take risks (chance)? Skill tends to be more serious than chance: is my game serious or casual? Are parts of the game tedious? If so, will adding elements of chance enliven them? Do parts of the game feel too random? If so, will replacing elements of chance with elements of skill or strategy make the players feel more in control? This is described as Lens of Balance between Skill and Chance by Schell, J. (2014).
2. The game should address important content, not trivia. To use The Lens of Problem Solving, designer must think about the problems their players must solve to succeed at the game, for every game has problems to solve. Schell, J. (2014) also state, “Games have a lot of rules—how to move and what you can and cannot do—but there is one rule at the foundation of all the others: the object of the game. Games are about achieving goals—you must be able to state your game’s goal and state it clearly.” However, one of the authors of Challenges for game Designers (Brathwaite, B., & Schreiber, I. 2009) put forth the following tentative definition for Games: “An activity with rules. It is a form of play often but not always involving conflict, either with other players, with the game system itself, or with randomness/fate/luck. Most games have goals, but not all (for example, The Sims and SimCity). Most games have defined start and end points, but not all (for example, World of Warcraft and Dungeons & Dragons). Most games involve decision making on the part of the players, but not all (for example, Candyland and Chutes and Ladders). A video game is a game (as defined above) that uses a digital video screen of some kind, in

some way." But sometimes, games without goals or clear content may be argued as a simulation instead of a game (for example, Minecraft).

3. The dynamics of the game should be easy to understand and interesting for the players but not obstruct or distort learning. The goal is to provide a practical, yet challenging exercise; added "bells and whistles "should be minimal and fulfill an important purpose. Again, Flow, Motivation and Engagement are very important to game design.
4. Players should not lose points for wrong answers. Punishing players for errors also punishes their effort and generates frustration. "A prince should be quick to reward and slow to punish." quote from Ovid. Schell, J. (2014) explained that players want to be judged favorably. Rewards are the way the game tells the player "you have done well." There are several common types of rewards that games tend to give - Praise, Points, Prolonged play, A gateway, Spectacle, Expression, Powers, Resources, Status, Completion. Each is different, but they all have one thing in common—they fulfill the player's desires. In contrast, Punishment is equally important. It could be used to could create endogenous value, possible punishment increases challenge, and Taking risks might be exciting and engaging. Punishment can be taking in a form of Shaming, Loss of points, Shortened play, Terminated play, Setback, Removal of powers and Resource depletion. However, Schell, J. (2014) state that reward is always a better tool for reinforcement than punishment.
5. Games should not be zero-sum exercises. In zero-sum games, players periodically receive rewards for game-sanctioned actions, but only one player achieves an ultimate win. In fact, Competition, Collaboration, exploring others and themselves are the reasons people play games (Schell, J. 2014). Therefore, balancing the game are exceedingly important and feedback are the bridge to express that importance. The feedback a player gets from the game is many things: judgment, reward, instruction, encouragement, and challenge, and all players should feel rewarded.

Unlike games, simulations are evolving case studies of a particular social or physical reality. Gredler, M. E. (2004) states, “The goal, instead of winning, is to take a bona fide role, address the issues, threats, or problems arising in the simulation, and experience the effects of one’s decisions.” Unlike games, the deep structure of a simulation is a dynamic set of relationships among several variables that reflect authentic causal or relational processes. Second, simulations require participants to apply their cognitive and metacognitive capabilities in the execution of a particular role. Thus, an important advantage of simulations, from the perspective of learning, is that they provide opportunities for students to solve ill-defined problems. Lastly, feedback in a simulation on a participants’ actions is in the form of changes in the status of the problem and/or the reactions of other participants. However, the design and validation of simulations are time-consuming, Gredler, M. E. (2004) still believes, simulations bridge the gap between the classroom and the real world by providing experience with complex, evolving problems. Furthermore, Gredler, M. E. (2004) has conclude, a key feature of educational games is the opportunity to apply subject matter knowledge in a new context, AKA Gamification, by applying game elements onto simulations to further improve the learning experience.

Leemkvil, et al. (2000) also did a study among games, simulations and case studies, as defined above. A simulation is a form of implementing a model over time. A case study describes by Leemkuil, et al. (2000) is an actual or hypothetical problem situation taken from the real world. Game simulations or case studies can found in a pure form or as hybrids with other. In the literature on games, the simulation games is the most often encountered hybrid.

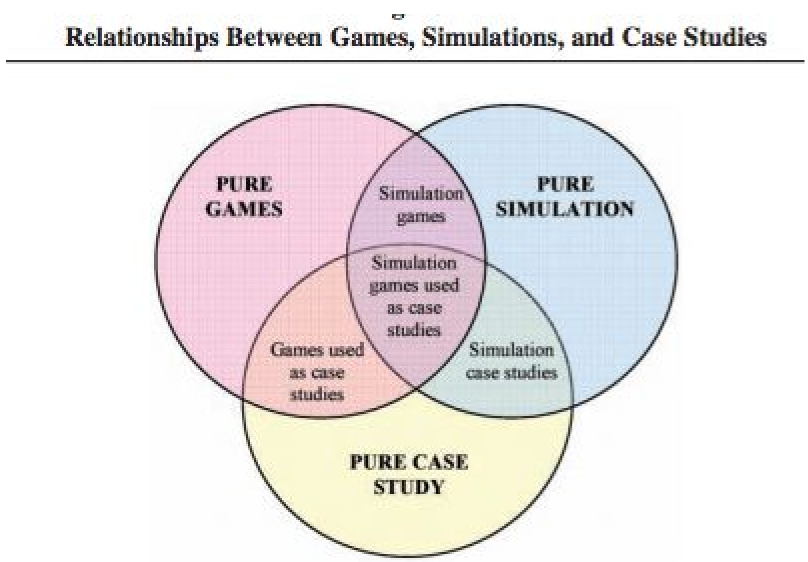

Figure 11: Relationships between Games, Simulations, and Case Studies

In distribution statement *The effectiveness of instructional games: A literature review and discussion,* a further study that justify gamification can help and improve effectiveness, Hays, R. T. (2005) states “it is more likely that games will be instructionally effective if the specific characteristics of the game (e.g., setting, player roles and activities, rules, etc.) overlap with specific instructional objectives. However, this overlap must be consciously structured on the basis of a thorough analysis of the reasons for the instruction and the instructional objectives to be met.”

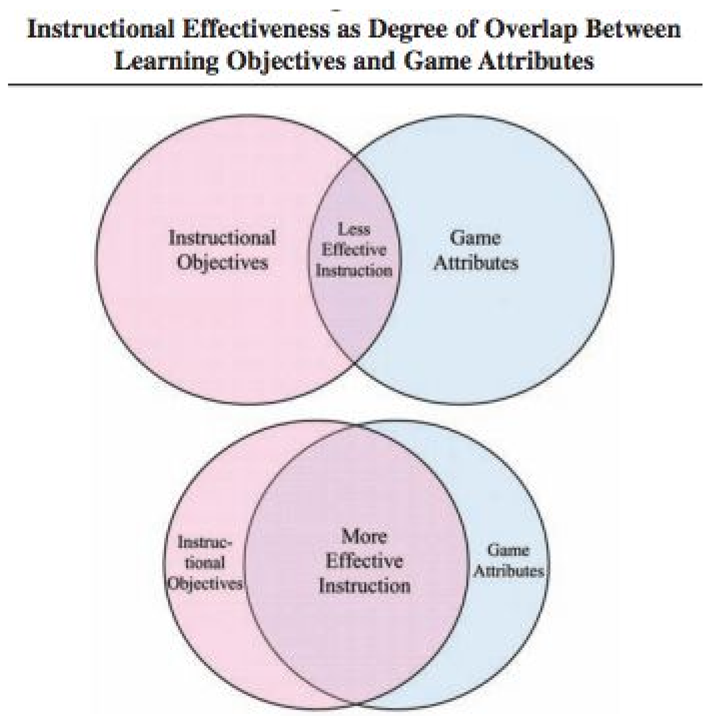

Figure 12: Instructional Effectiveness as Degree of Overlap between Learning Objectives and Game Attributes

3.2 User Interface & Human computer interaction (HCI)

Before converting a simulator to a game, Designers need to identify the importance of requirements, different types of requirements first. Rogers, Y., Sharp, H., and Preece, J. (2011) suggested a way to get designers started on the design, by asking themselves what, how and why. What would be to understand as much as possible about users, task, and context. Also produce a stable set of requirements. To do so, understanding user's abilities, background, attitude to Games and setting the right level of system used are essential. The system level can be classified as novice, expert, casual, and frequent. And where How is all the human computer interaction happens. Lastly, why is to understand the stage where failure occurs.

Human Computer interaction is one of the factors that both Games and Simulations share. It is an important factor that both design can't live without. An Interaction Design described by Rogers, Y., Sharp, H., and Preece, J. (2011) is a process, a gold-directed problem solving activity informed by intended use, target domain, materials, cost and feasibility. Also it is a creative activity and a decision making activity to balance tradeoffs. The four approaches they recommended are user centered design, activity centered design, systems design and genius design. This report will be mainly focused on the User centered approach. A user centered approach is based on early focus on users and task and establishing requirements.

Instructing types of Human Computer interaction is a great way to carry out the user centered design. It issues commands and selection options, where users instruct a system and tell it what to do, i.e. time, print files, save files and the next action. It is a very common conceptual model, underlying a diversity of devices and systems. The main benefit is that instructing models supports quick and efficient interaction. It is good for repetitive kinds of actions performed on multiple objects. (Rogers, Y., Sharp, H., & Preece, J. 2011) The kind of interface used to support the mode are usually menu based. There are few important aspects in designing a menu based model, or a Navigating the interface.

3.2.1 Navigating the interface (Shneiderman, B. 2010)

Since navigation can be difficult for many users, providing clear rules is helpful. General principles, and specific rules. This sample of the guidelines gives useful advice and a taste of their style:

- Standardize task sequences. Allow users to perform tasks in the same sequence *and manner across similar conditions.*
- *Ensure that embedded links are descriptive. When using embedded links, the link* text should accurately describe the link's destination.
- Use unique and descriptive headings. Use headings that are unique from one another and conceptually related to the content they describe.
- Use check boxes for binary choices. Provide a check box control for users to make a choice between two clearly distinguishable states, including "on" or "off."

- Develop pages that will print properly. If users are likely to print one or more pages, develop pages with widths that print properly.
- Use thumbnail images to preview larger images. When viewing full-size images is not critical, first provide a thumbnail of the image.

These guidelines are clarified by examples and supported by research studies. A goal for guidelines writers is to be clear and comprehensible, using meaningful examples. However, controversies over guidelines are lively, often leading to revisions and the creation of alternatives.

3.2.2 Organizing the display (Shneiderman, B. 2010)

Display design is a large topic with many special cases. Smith and Mosier (1986) offer five high-level goals as part of their guidelines for data display:

- Consistency of data display. During the design process, the terminology, abbreviations, formats, colors, capitalization, and so on should all be standardized and controlled by use of a written (or computer-managed) dictionary of these items.
- Efficient information assimilation by the user. The format should be familiar to the operator and should be related to the tasks required to be performed with the data. This objective is served by rules for neat columns of data, left justification for alphanumeric data, right justification of integers, lining up of decimal points, proper spacing, use of comprehensible labels, and appropriate measurement units and numbers of decimal digits.
- Minimal memory load on the user. Users should not be required to remember information from one screen for use on another screen. Tasks should be arranged such that completion occurs with few actions, minimizing the chance of forgetting to perform a step. Labels and common formats should be provided for novice or intermittent users.
- Compatibility of data display with data entry. The format of displayed information should be linked clearly to the format of the data entry. Where possible and appropriate, the output fields should

also act as editable input fields.

- Flexibility for user control of data display. Users should be able to get the information from the display in the form most convenient for the task on which they are working. For example, the order of columns and sorting of rows should be easily changeable by the users.

3.2.3 Getting the user's attention (Shneiderman, B. 2010)

Since substantial information may be presented to users for the normal performance of their work, exceptional conditions or time-dependent information must be presented to attract attention. These guidelines detail several techniques for getting the user's attention:

- Intensity. Use two levels only, with limited use of high intensity to draw attention.
- Marking. Underline the item, enclose it in a box, point to it with an arrow, or use an indicator including an asterisk, bullet, dash, plus sign, or X.
- Size. Use up to four sizes, with larger sizes attracting more attention.
- Choice of fonts. Use up to three fonts. • Inverse video. Use inverse coloring.
- Blinking. Use blinking displays (2-4 Hz) or blinking color changes with great care and in limited areas.
- Color. Use up to four standard colors, with additional colors reserved for occasional use.
- Audio. Use soft tones for regular positive feedback and harsh sounds for rare emergency conditions.

A few words of caution are necessary. There is a danger in creating cluttered displays by overusing these techniques. Some web designers use blinking advertisements or animated icons to attract attention, but users almost universally disapprove. Animation is appreciated primarily when it provides meaningful information, including for a progress indicator. Novices need simple, logically organized, and well-labeled displays that guide their actions. Expert users prefer limited labels on fields so data values are easier to extract; subtle highlighting of changed values or positional presentation is sufficient. Display formats must be tested with users for comprehensibility.

3.2.4 Manu Interface

Game User interface is a great extension of getting the user's attention (3.2.3). Also it is the interaction link between the user and the machine. Few elements may involve are Windows, Icons, Menus and Pointing devices. And GUIs are extensions of these building blocks, by adding colors, 3D, sounds, and animations, even adding new graphic elements, such as toolbars, docks, and rollovers. Windows were invented to overcome Physical constraints of a computer display, by enable more information to be viewed and tasks to be performed. However, multiple windows can make it difficult for users, and there are some techniques that can help to solve such problem, such as listing, and Shrinking. (R. Grewal 2017)

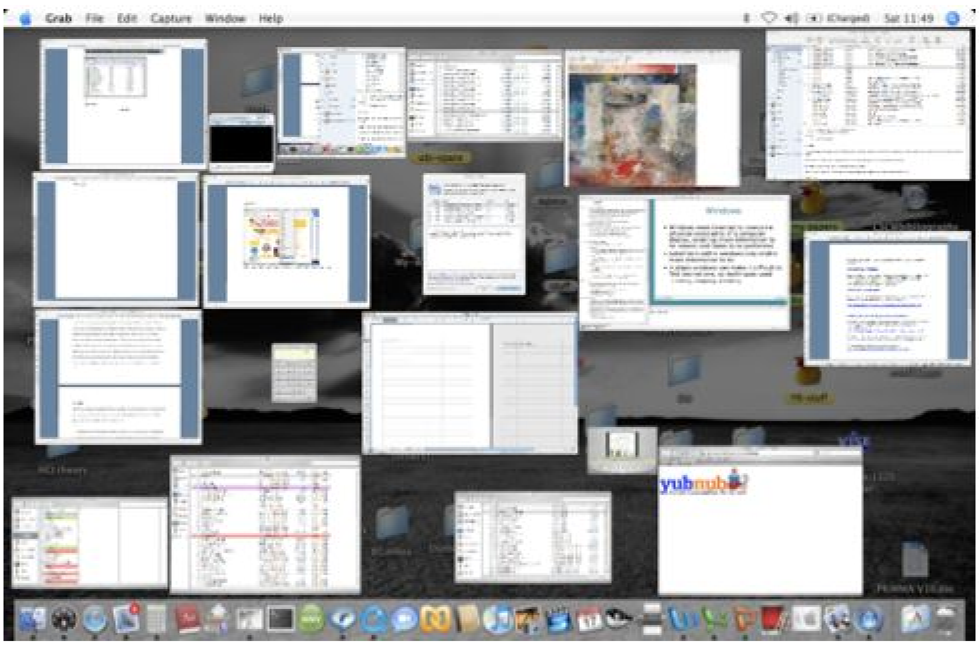

Figure 13: Apple's shrinking windows

Figure 14: Safari panorama window view

. Menus is the next important element in user computer interaction. Different menu interface styles suit better for different situations. Flat menus is good at displaying a small numbers of options at the same time, and were the size of display is small, but have to nest the lists of options within each other, requiring several steps to get the list with the desired option, and moving through previous screens can be tedious. (R. Grewal 2017)

Figure 15: IPod flat menu structure

Expanding menus on the other hand, enables more options to be shown on a single screen than is possible with a single flat menu. It is a more flexible navigation, allowing for elections of options to be done in the same window. The more popular ones are cascading ones, such as primary, secondary and event tertiary menus, downside in that they require precise mouse control, and it can result in overshooting or selecting wrong options. (R. Grewal 2017)

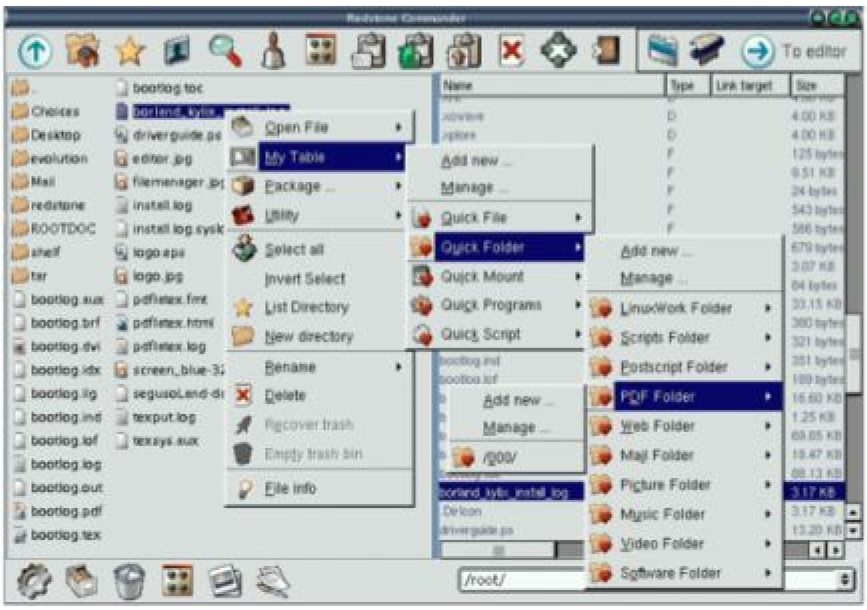

Figure 16: Cascading menu

Lastly is the Contextual menus, it provides access to often used commands, that make sense in the context of a current task, appears when user presses the control key while clicking on an interface element. It helps overcome some of the navigation problems associated with cascading menus.

Figure 17: Windows Jump List Menu

3.2.5 Icons

Since Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., & Mackey, K. (1989) the Xerox Star days, icons have changed in their looks and feel, not more black and white. Colors, shadowing, photorealistic images, 3D rendering and animation has become the newer trends. Many designed to be very detailed and animated making them both visually attractive and informative. GUIs now highly inviting, emotionally appealing and feel alive. In the lecture note form R. Grewal (2016), It stated “Icons are assumed to be easier to learn and remember than commands.” It can be designed to be compact and variable positions on a screen and it is pervasive in every interface. Icons takes in many forms. The mapping between the representation and underlying referent can be similar, such as a picture of a file to represent the object file. Analogical can another form of representations, such as a picture of a pair of scissors to represent ‘Cut’. Lastly is the arbitrary form, such as using a ‘X’ to represent ‘Delete’. From Rogers, Y., Sharp, H., and Preece, J.’s study (2011) most effective icons are similar ones. On the other hand, many operations are complicated actions, it is more difficult to represent them, by using a combination of objects and symbols that capture the salient part of an action could somehow solve this problem.

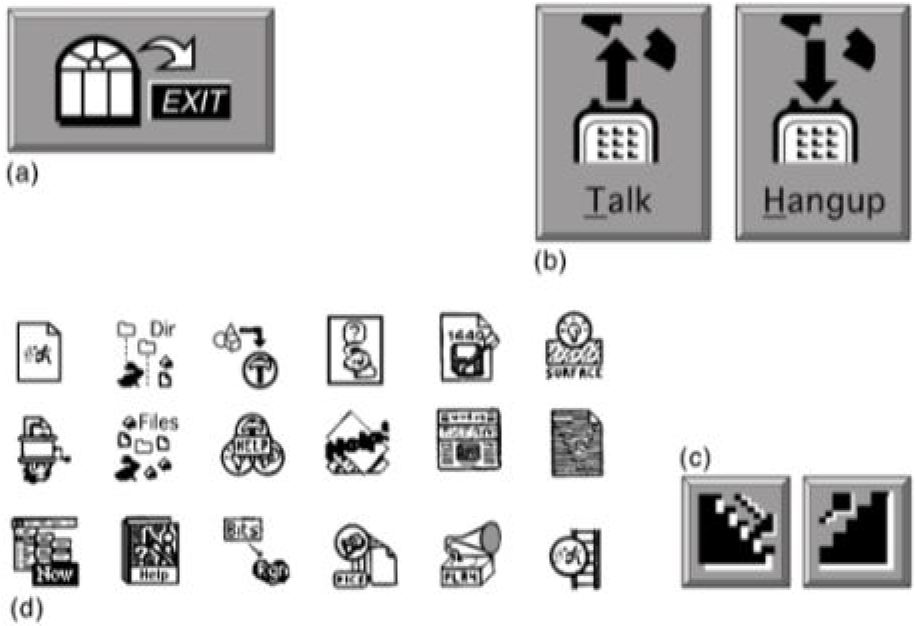

Figure 18: Early icons examples

Figure 19: Newer icons example

3.2.6 Details of UI designs

From a designer's perspective, application serves different functions. (R. Grewal 2016) Designers need to make decisions about the interface as a whole, and smaller unit within an application should serve a single important functions. If the Interface is designed to show or play a single piece of content, then the Information architecture is straightforward. It may take in the form of a long, vertically scrolled page of flowed text, or a zoomable interface for very large, fine grained artifacts, such as maps, images or information graphic. And also the form of a media representations - video or audio. As designing the interface, Designers need to consider of using patterns and techniques to support their designs. Alternative view, to show the content in more than one way, or multiple workspace, or deep linked state, or even a mobile pattern can smooth out the design. When Facilitate a single task, the interface is simply trying to get the job done, not much Information architecture needed if the user can the necessary work in a small contained area. Such as a sign-in box. However, when the task gets more complicated, then designers need to consider of breaking the task down into smaller steps or group of steps. A settings editor that gives users a way to change or modify the settings would ease the complicated tasks.

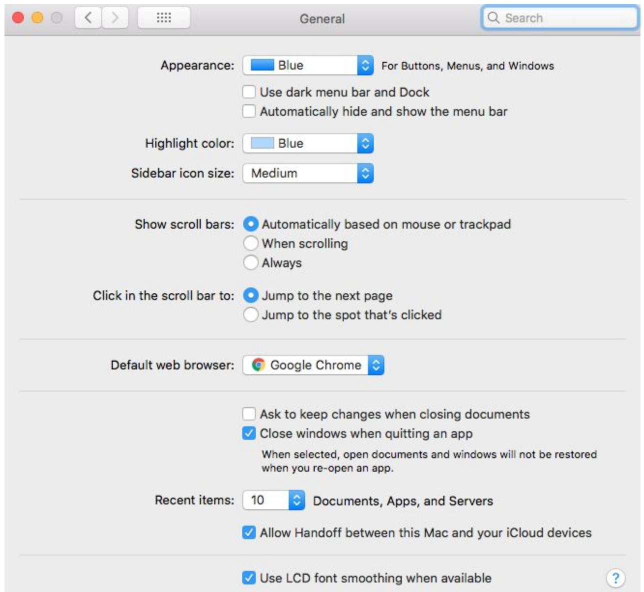

Figure 20: Setting Editor with Left Alignment

However, Lists on the other hand present rich challenges in information architecture. (R. Grewal 2016) List often shows a noticeable non visual characteristic of length, order, grouping, item types, interaction and dynamic behavior. With more complex list, graph representation is a great way to visualize data. And answers the question of how the data are organized, and what relate to what.

Inside of any Setting Editor or a regular menu, Buttons, menu bars, pop-up menus, drop down menus, toolbars, links, and action panels in an interface are actions made accessible, and the common commands to use of those functions are clicking, drag and drop, keyboard actions or typed commands. (R. Grewal 2016) the reason for those accessible actions are for the use of gathering input from users. Few principles list by R. Grewal (2016) when doing input and form designs are:

- Make sure the user understands what is asked for and why
- If possible, avoid asking the question at all
- Make clear instructions
- Respond sensitively to errors, and try to be forgiving when possible
- And beware a literal translation from the underlying programming mode

More importantly, different accessible actions suit better for different situation. The controls for selecting of two options are often checkbox with a click action, it is simple, and low on space consumption. Two radio buttons is an advanced checkbox, that both choices are stated, visible, externally consistent and saves development efforts. And lastly a drop down list or a combo box can be applied too, the advantages of them are all choices are stated, low and predictable space consumption, mainly is because it can be expanded later to more than two choices. Moreover, it controls for entering of text or one of N choices. As for controlling for entering numbers, R. Grewal (2016) had listed many different forms in his class lecture notes. Text field for example, it is visually elegant, and permits wide variety of formats or data types. However, it may cause temporary confusion of input formats, and requires careful backend validation. Spin box or slider are best used for integers or discrete steps. With spin box, users can arrive at a value with both keyboard-only or mouse-only accessible actions, and the values are constrained to be in range when buttons are used. Furthermore, slider is a obvious metaphor with the same reason justified in Chapter 3.2.5. The position of value in ranges is shown visually, and limiting the user from entering a number outside the range.

R. Grewal (2016) explained the reason for controlling user inputs by using the best accessible action controls is to prevent error. Errors can take in 3 forms, slips, lapses and mistakes. Human errors such as inattention or inappropriate attention are often the causes for slips and lapses. By Avoid habitual action sequences with identical prefixes and avoid action with very similar descriptions can help to avoid capture and description slips. As for avoiding lapses, designer should keep procedures short, minimize interruptions and use forcing functions. Mistakes on the other hand is just simply using the wrong procedure for the goal, and typically found in rule based behavior or problem solving behavior designs. To further error preventions, designer should disable illegal commands, use menu or forms, combo boxes and all information should be visible and kept simple at all times.

3.2.7 Design Layout

With guideline stated in chapter 3.2.2 and 3.2.3 provided by Shneiderman, B. (2010), a design layout with a good visual hierarchy manipulate the user's attention on a page to convey meaning, sequence and points of interaction. A good visual hierarchy gives instant clues about the relative importance of page elements, and the relationship among them. Moreover, Color is immediate, and it is one of the first things that perceived about the design. However, anything that makes the text or contents difficult to read or see should be ruled out at once. (R. Grewal 2016) Saturated colors should be avoid, only few colors should be used and keep consistent with the expectations.

Several layout patterns are often recommended to take advantage of how people scan or read through a design. Three of the more common are the Gutenberg diagram, the z-pattern layout, and the f-pattern layout where to place important information. R. Grewal (2016) had mentioned few common patterns in his work. Eldesouky, D. F. B. (2013) presented a more detailed report on 3 of the layout patterns.

1. The Gutenberg diagram describes a general pattern the eyes move through when looking at evenly distributed, homogeneous information. This pattern commonly applies best to text-heavy content. The pattern suggests that the eye will sweep across and down the page in a series of horizontal movements called axes of orientation. Each sweep starts a little further from the left edge and moves a little closer to the right edge. The overall movement is for the eye to travel from the primary area to the terminal area and this path is referred to as reading gravity and Important elements should be placed along the reading gravity path.

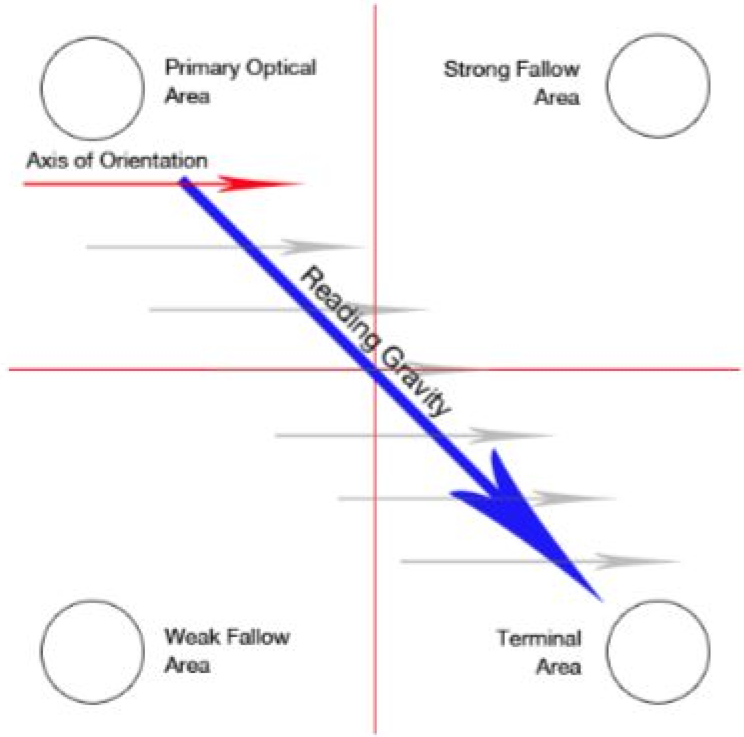

Figure 21: Gutenberg diagram

2. The z-pattern layout follows the shape of the letter z. Readers will start in the top/left, move horizontally to the top/right and then diagonally to the bottom/right before finishing with another horizontal movement to the bottom/right. The main difference with the Gutenberg diagram is that the z-pattern suggests viewers will pass through the two fallow areas. Otherwise they still start and end in the same places and still pass through the middle. As with Gutenberg a designer would place the most important information along the pattern's path. The z- pattern is good for simple designs with a few key elements that need to be seen.

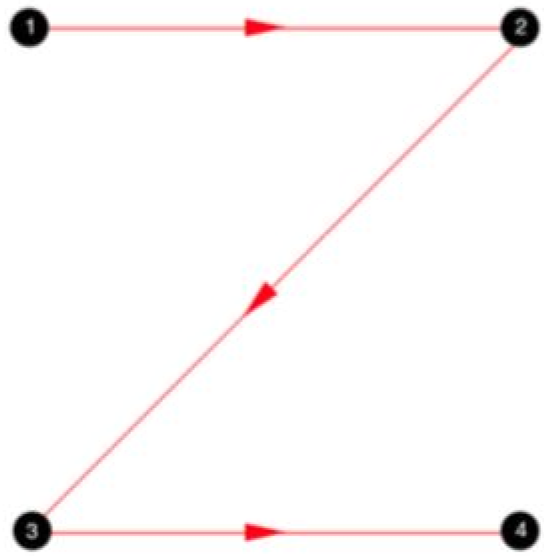

Figure 22: z-pattern layout

3. The f-pattern gets mentioned by Eldesouky, D. F. B. (2013) and as expected, it follows the shape of the letter F. As with the other patterns the eye starts in the top/left, moves horizontally to the top/right and then comes back to the left edge before making another horizontal sweep to the right. This second sweep won't extend as far as the first sweep.

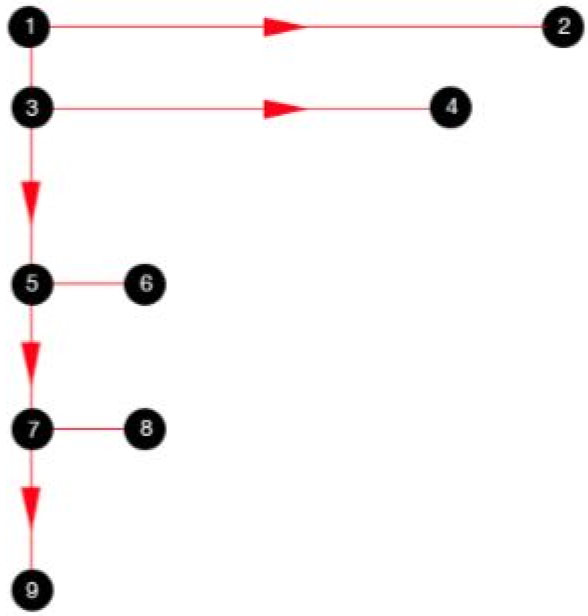

Figure 23: f-pattern layout

3.3 Game feedback

All interaction designs are a form of direct feedback, every time an action is taking place, an immediate feedback will be presented to the user, thus the task can easily been stay on focused. Experiences without feedback are frustrating and confusing. Schell, J. (2010) believed players need to see that they are making progress when solving a problem. Designers need to make sure users are getting their feedback, but sometimes, the importance of good feedback may be easily overlooked. In a game, The feedback a player gets from the game is many things: judgment, reward, instruction, encouragement, and challenge. (Schell, J. 2014) Since feedback in a game is continuous but needs to be different in different situations. It takes a lot of mental effort to use the lens of feedback (Schell, J. 2014) in every moment of the game, but it is time well spent, because it will help guarantee that the game is clear, challenging, and rewarding.

Leaderboard is a social mechanisms enable us to compete, collaborate, and coordinate our activities. Keeping aware of what others are doing and letting others know what you are doing are important aspects of collaborative working and socialising. (Rogers, Y., Sharp, H., & Preece, J. 2011) Providing reassuring feedback that can be both informative and fun, but can also be intrusive, causing people to get annoyed and even angry, in turn this can affect the usability of an interface. For example, each time a player gets a point or loads of points it can be rewarding and informative, Same reason can be applied for a high leaderboard rankings, special achievements, or anything else that gives a player higher status in the community of players can be very desirable award, especially to competitive players. On the other hand, when HP bar drops low because of one wrong action, may cause some negative emotions.

3.4 Project Proposal

Since gamification has been used in the field of training in the past and based on the research of game and simulation elements, this project will target the field of gamify an existing simulation. The goal of this project is to apply the studies that has done before, and "gamify" an existing model of heavy machine operation training that was developed in Unity. The VR-based SAMS model is designed to train load-haul-dump (LHD) operators how to safely operate in a mine environment. We intend to implement additional gamification features in order increase the extrinsic motivation and engagement of learners. This will be done through research-driven design decisions. The current base simulation already visualizes site conditions, working behavior and construction methods or sequencing.

Since the base simulation is already in operation this work can focus on the gamification elements and play style. Each style will focus on it own individual task and experience. These different styles Include "bonus mining" which focuses on the scores at the end of the game, which gives a sense of "Last minute of work, how much more can I earn at the end of the day?" and an "efficient mining mode which allow players to put aside of the fear of limited time and the damage they might cost to the vehicle, by giving them a safe and re environment to focus on the efficiency and controls of the game play. Furthermore, leaderboards will be presented and displayed at the end of each mode, to further motivate users by giving them a sense of competition, giving them room for self-improvements. Ultimately, the goal is to get the users motivated, so they would come back to play some more, and enjoy the training that were presented to them. The following are the game mode and element that are considered in the design.

3.4.1 Bonus Mining gamification elements:

- Ore scooped and delivered will give money or scores as feedback
- Feedback on collisions - warning, and damage may affect the HUD
- Machine Health - Health bar
- Timing information - countdown / Clock / warning of remaining time (time can be set at the beginning of the test)
- Show game stats, scoreboard, money earned at the end or game

3.4.2 Efficient Mining gamification elements:

- Feedback provided on or dumping efficiency
- Feedback on collisions - warning, and damage may affect the HUD
- Machine Health - Health bar
- Timing information - unlimited time / end until goal reached / countdown Clock
- Show game stats, scoreboard, efficiency result at the end or game

3.4.3 Element for Research reasons

- Data logs for future studies

- Enable / Disable function for comparison tests

3.4.4 Element for Game Only (aside from study)

- (PRO) Random mine pile generators

The Randomness creates challenging game play, but it may take many user testing to get a good study results and too much time consumption.

Chapter 4 - Implementation

4.1 Implementation Set Up

By Following the guidelines of chapter 3, and understanding the difference between games and simulations, the first step is to understand as much as possible about users, task, and context. As presented in the project proposal, the goal of this project is to apply the studies that has done before, and “gamify” an existing model of heavy machine operation training that was developed in Unity. The VR-based SAMS model is designed to train load-haul-dump (LHD) operators how to safely operate in a mine environment. The users would be the mine workers that are in training. And by implementing additional gamification features to increase the extrinsic motivation and engagement of those trainees. Since the existing Simulator is already functional, so the trainees are assumed to be familiar with the existing mechanics and VR environment. Thus the system level will be set as expert. Therefore, there will not be an in game tutorial about the basic mechanics. The tutorial will be in a separate environment as the instructor required. Bonus Mining mode and efficient Mining mode are two main mandatory context requested by the instructors. The required game elements are also listed in the project proposal. Both model shares the element of a real time feedback on the scoreboard but with different scoring mechanisms. Both model Also share a timing mechanism, Collison feedbacks and a leaderboard function at the end. As for bonus, some of the vehicle attributes such as current gear level, reaction timer, gas and brake pedals will be re-implemented with the new elements to the HUD for the reason state in Chapter 1.4 and also for a better user experience.

4.2 Menu display

To separate the two game mode, a sequence of Instructing and Navigating the interface will be implemented to existing menu. A stepwise structure interface is implemented following the steps of chapter 3.2.1 Navigating the interface. A stepwise structure leads the user step by step through the screens in a prescribed sequence, with back or next links to each page.

Figure 24 to Figure 26 follows a stepwise structure

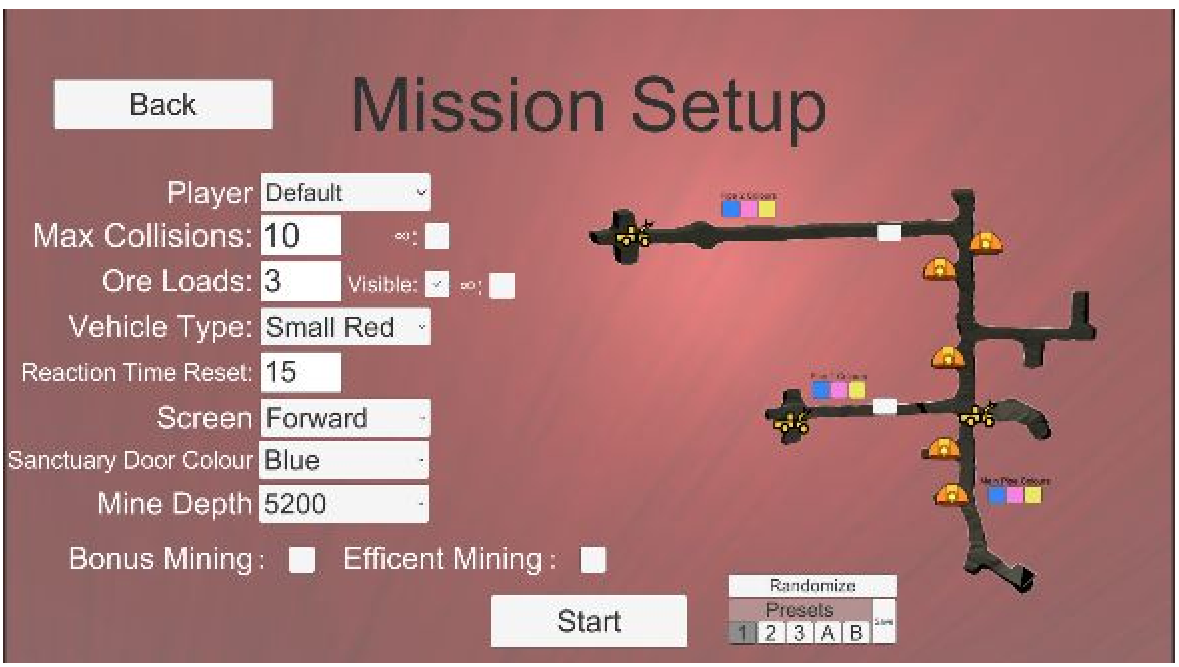

Figure 24: Mission Setup Menu

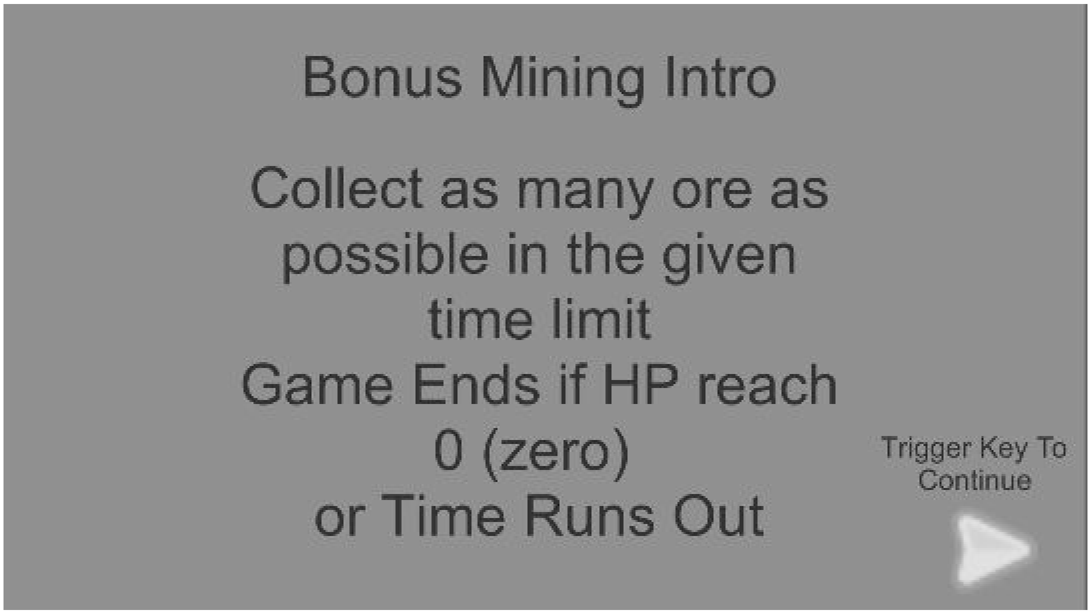

Figure 25: Instruction Page

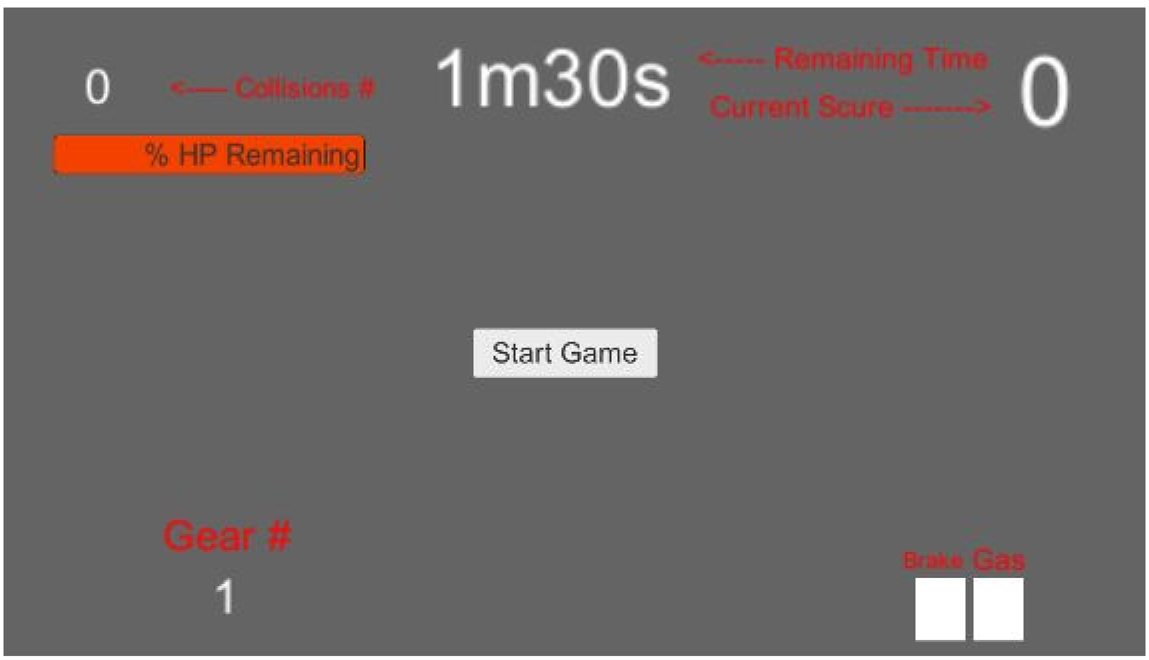

Figure 26: Game UI instruction page

Moreover, from the understanding of chapter 3.2.4, and due to window management that enables users to move fluidly between different windows and monitors. Therefore, in this design, there won't me multiple windows displaying at the same time as the Menus. Menus and windows will be transitions with a fade away effect. By using the techniques mentioned in chapter 3.2.4, multiple panels will be shrinked in a way that can be easily spotted by the user with The VR headset.

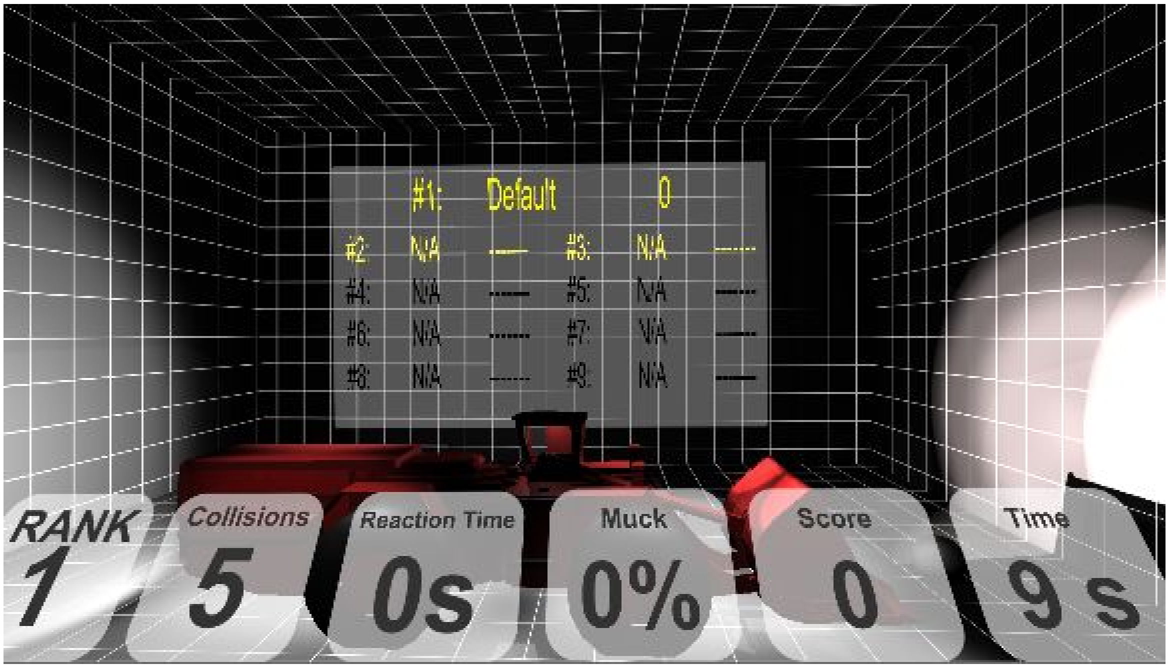

Figure 27: Multiple panels

4.3 New menu elements

New functions have been added and modified the existing menu interface through the designing process, to satisfy the functions of enable and disable the new added game modes. Furthermore, some of the original menu elements will be modified, and adjusted according to different game modes. The design layout is kept constant with original design for the reason of learnability and usability in UI designs presented by R. Grewal. (2016)

Figure 28: Bonus Mining Game mode mission Setup page

Max Collisions, Ore Loads, Reaction Time Reset and AI miner option is disabled for user and a new timer option is added in the Bonus Mining Game mode.

Figure 29: Efficient Mining Game mode mission Setup page

Max Collisions, Ore Loads, Reaction and Time Reset is disabled, the miner AI is automatically enabled and a default 20 minus time is added for the Bonus Mining Game mode.

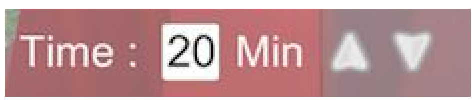

Figure 30: In game timer modifier

Timer can be adjusted with up and down arrows. This design to minimize the error might cause with text field, With spin box, users can arrive at a value with both keyboard-only or mouse-only accessible actions, and the values are constrained to be in range when buttons are used. (Chapter 3.2.6)

Figure 31: Game mode selections

With results shown in the study stated in chapter 3.2.6, Checkbox with a click action is a great controller for selecting of two options. A check mark icon will be shown in the box and uses as a clear feedback for the of enabling and disabling action. More importantly, it is simple, and low on space consumption. With study shown in chapter 3, Interface should be kept clean and organized, and all instructions should be clear and on point.

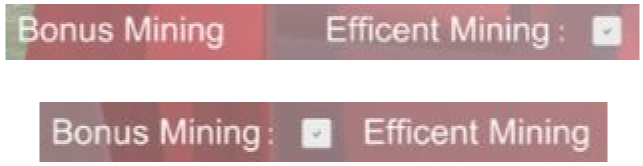

Figure 32: Selecting Efficient Mining Mode & Selecting Bonus Mining Mode

When one Game mode has been selected, the other one will be automatically disabled for the reason of error control stated through Chapter 3.2.5 to chapter 3.26.

Figure 33: Modified Menu

For the purpose of game play, few options are disabled from user. Collisions are presented as impact damage to the vehicle in both game mode, and it is limited with a Max vehicle health in the bonus mining mode, and as for Efficient game mode, due to the instructor's request, players are rather not overtly punished for collisions. and rewarded for safe driving, therefore max collisions will be set to unlimited because collision does not affect the game play. Both game mode requires unlimited ore loads, and reaction time Reset will set at a fix value of 15 second, also for the reason of error control stated through Chapter 3.2.5 to chapter 3.26.

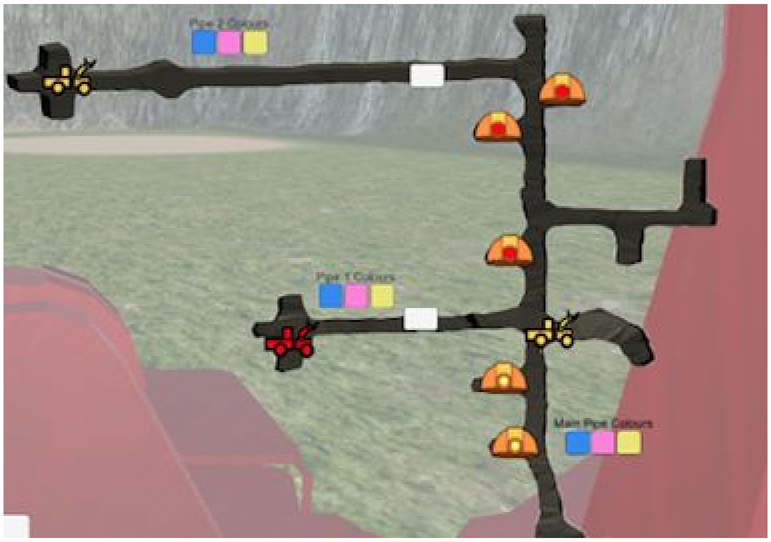

Figure 34: Game Map layout

A clear map layout is shown at the beginning on the mission set menu to give users a preview of the environment layout. Moreover, with the use of metaphor icons of a safety helmet, and a miniature LHD to represent the position of pedestrian AI miners, and AI LHDs for more visually attractive and informative effect. (chapter 3.2.5) Both icon will turn red as a feedback when it is enabled.

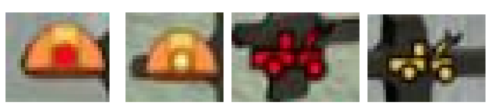

Figure 35: Far left: AI Miner enabled, Center left: AI Miner Disabled, Center right: AI LHD enabled, Far right AI LHD Disabled

4.4 Game elements implemented

The in game VR view interface uses a combination of Gutenberg diagram and z-pattern layout presented in chapter 3.2.7, to establish the effect of grabbing user's attention, and game elements are placed along point 1, and point 2, and on point 3, and 4. And the level of importance drops along the reading gravity path. More importantly, this design does not conflict with the purpose of HUD design.

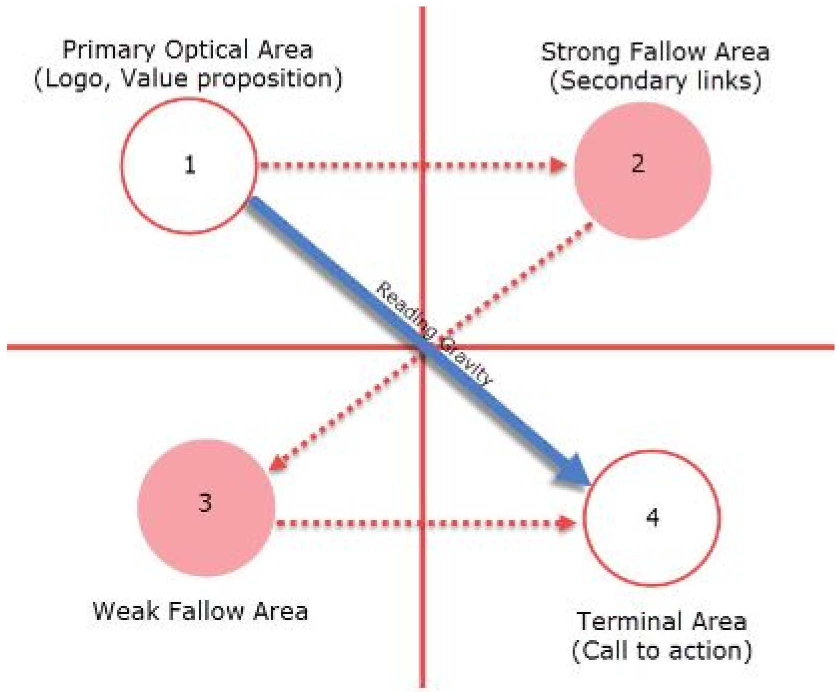

Figure 36: Combination of Gutenberg diagram and z-pattern layout

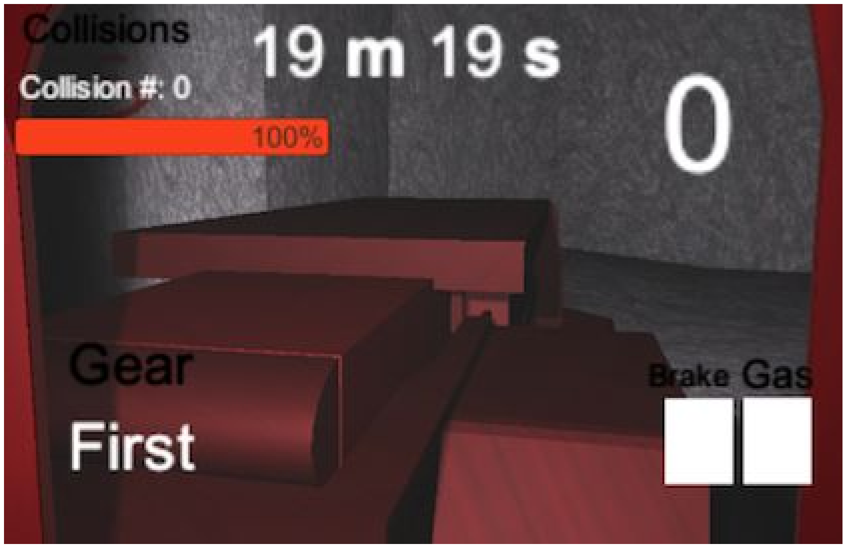

Figure 37: Applying To game UI

This UI is fixed on the VR camera. It will alway in the user's line of sight to apply the HUD affect onto VR game environment. Furthermore, by creating a seperate camera for the HUD, and putting it at a different camera depth, this solves the problem of HUDs clipping into the walls and surroundings.

Figure 38: Multiple camera for different HUD

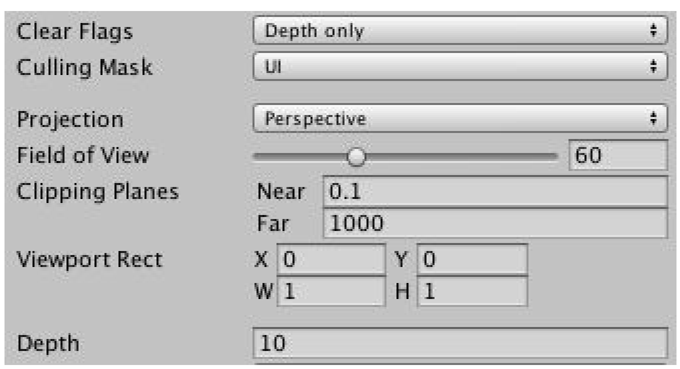

Figure 39: Camera only randers in Depth, and Set at Depth 10

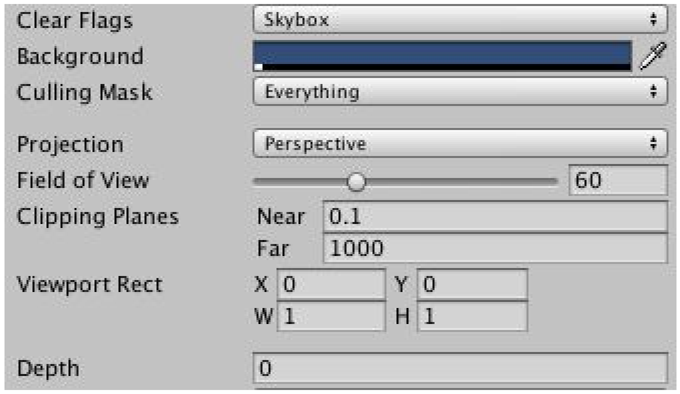

Figure 40: Camara randers everything with skybox, and Set at Depth 0

Camera at depth 0 will always randers first, and the camera with the highest depth will alway rander last and on top of everything.

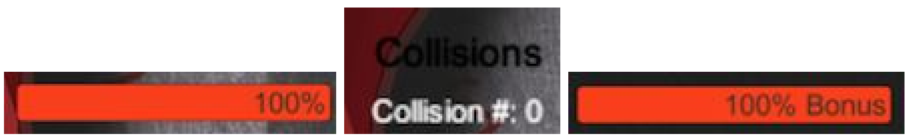

Figure 41: HP bars and Collision Feedback 1.0

feedbacks on collision - Two different HP for each game mode bar and Collision counter

Collision counter increment by 1, each time a collision occurs. A damage will have calculated based on the current velocity of the vehicle, and the decreased HP bar will give users a feedback on the damage that the vehicle has taken.

Figure 42: HP bars and Collision 2.0

Figure 43: Reaction Timer

Reaction Timer only displays in Efficiency Mining mode, as a timer feedback on how quickly a player can tag a pedestrian AI miner.

Figure 44: Time Feedback - Remaining timer

Both game mode has a 20 minus Game timer. Both timers can be modified by the user with a fixed range 60 minus to 1 minus.

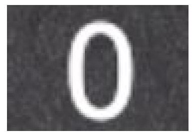

Figure 45: Reward feedback - Score board

Different game mode has different scoring mechanisms. Bonus Mining mode only counts the ore collected as part of the scoring mechanism and each ore counts a certain amount of points based on the magnitude of the ore. On the other hand, Efficiency Mining mode not only take count of the ore collection, and also takes the pedestrian reaction timer into count as part of the scoring mechanism, given a score based on how quickly the player tags a pedestrian, the faster he/she tags the pedestrian AI miner, the higher the point it with a fixed point range of 0 to 20 points. At the end of the Efficiency Mining mode, the remaining health bar will count as bonus points, and will be added to the scoreboard.

Figure 46: Vehicle states feedback - Brake/Gas Pedal and Current Gear level

Both feedback was originally place inside the vehicle with a HDD model. With the research stated in chapter 1.4, both feedbacks were moved on screen, with a HUD model along with other game elements.

4.5 End Game Feedback

At the end of each game mode, all information respected to the current played game mode will be presented to the user as a summary report. “Rank” is a representation of ranking respect to the end game scores. Users are given a rank number as a feedback for the wellness they done through the game, and it is compared with all existing runs under the same game mode. Rank 1 as the highest rank and restaurants the highest scores in all existing game play under the same game mode. Rank numbers increase as score decrease. Total number of collisions, average reaction time, much percent, end game scores and total game time used will be displayed as overall game feedback.

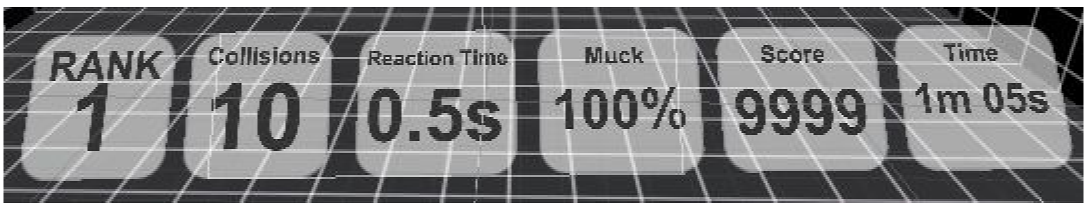

Figure 47: overall game feedback

From the studies done in chapter 1.2 and 3.1 shows competition improves motivation. Leaderboard is a visual extension of the Rank level. It show the top 9 existing players in the current game mode with their respected scores with the player ID displayed on the left and scores on the right. This further motivate users by giving them a sense of competition, showing them there are rooms for some self-improvements.

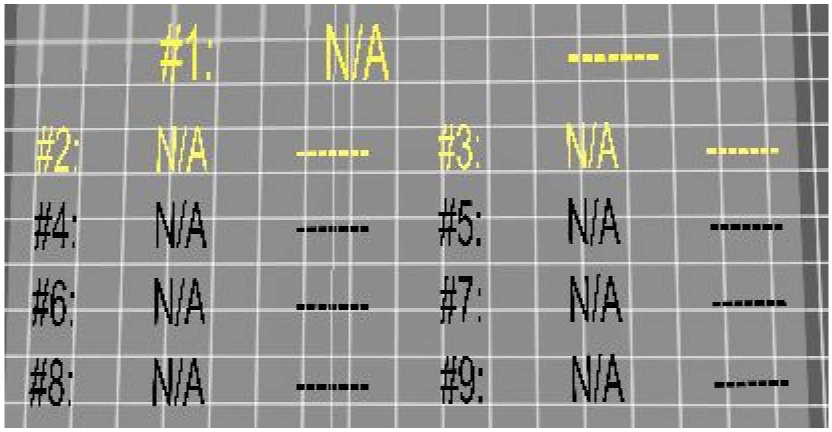

Figure 48: Blank Leaderboard

Furthermore, with the study shown in chapter 3, by using a z-pattern layout to manipulate the user's attention on a page to convey meaning of importance. moreover, the uses of 2 different colors can better grab user's attention and making the top 3 players stands out from the rest.

4.6 Game Balance (skill vs time, punishment and rewards)

Game balance is ecentral for every game based on the research done in Chapter 3. However, there will not be any user testing for this project, therefore it is very challenging for balancing all the implemented game elements. Especially for efficiency game mode, due to a complicated scoring system. There are still rooms for balancing between the reaction score, the ore collection score, and the end game bonus score. The bonus score alone may even worth more than the scores gained from collecting ores for 20 minus. Another alternative balancing issue may be the scores gained from tagging a pedestrian be too high, may cause the user to camp at a pedestrian spawn point for the unbalanced points. Therefore, even one score mechanics is out of balance, it may cause a shift in the soul purposes for the game mode.

.

4.7 Element for Research purpose only & Element for game only

In this Project, both game mode are designed for research purposes. Researchers want to study and compare the training result between the pure simulation model and the new gamefied model, and the better model may ba applied to real life. In order to better study both game mode, there is a build in file IO function. At the end each mode, there is a save function.

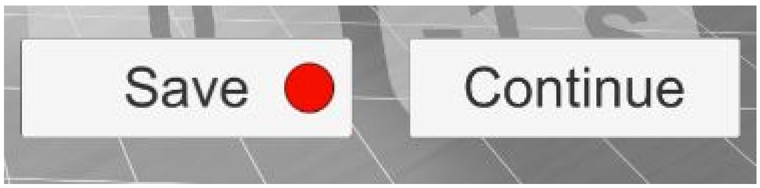

Figure 49: Save and Continue function

Player will given a choice of saving there current game states or continue. If save funcuntone was chosen, a feedback will shown to player with a greed sign.

Figure 50: File saved feedback

If player chose “Continue” instead of “Save”, a second confirmation will shown to the player, as a prevention of any slips and lapses errors explained in Chapter 3.2.6.

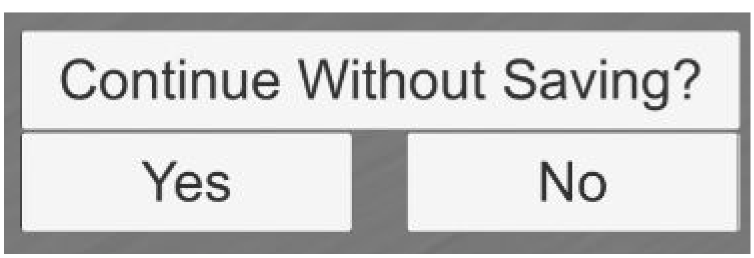

Figure 51: Second confirmation

If player has decided to save all date, only then, there scores will be add and updated to the leaderboard. Furthermore, a copy of MXL file will be saved with all the game logs. Game logs include everything that has happened to the player, such as game start time, end time, and each time a score is gained and each collision occurred and its corresponding damage to the vehicle will be logged and added to that file. This MXL file can be exported for future studies on the game mode.

In contrast, both game mode can be modified to establish a more challenging game play, by bringing in the concept of randomness and chance. Chance is an essential part of a fun game because chance means uncertainty, and uncertainty means surprises. And as discussed earlier in chapter 3, surprises are an important source of human pleasure and the secret ingredient of fun. To enable the concept of chance is just simply allow the fixed ore pile to randomly appear on the map. This makes the game more challenging and interesting, allowing the player to be further motivated and engaged with the game play. However, this idea was not implemented in this project, because the randomness factor may cause massive time consumption user testing for the researchers. And this tends to deviate from the studying purpose, thus the idea was terminated at the phase of implementation.

4.8 comparison between new and old model (Josh's UI and Current UI && older UI)

There two major improvement done in this project compared with the older model. First of all, this new model added a new HUD system to better present important information and improves safe driving and some other reason presented in Chapter 1.4. Compare the figure 41 with figure 56 we can see the difference and state, the new implemented UI is better than the older model.

Figure 52: HDD from older model

Figure 53: VR set with no HDD system

Additionally, some feedbacks was received from previous user testing studies on the old model, the flashing warning sign used in the older model shown in figure 57 is very distracting and annoying to look at. This effect was replaced with a HP bar. The HP bar is neat and clean to look at. More importantly, with a numerical expression on the collision damage provides a clear and better feedbacks.

Figure 54: distracting and annoying Warning Sign from older model

Chapter 5 - Conclusion

This project has established the function of all the requested elements, and the implementation followed a clear guideline done through researches, all the research is analyzed and explained through Chapter 1 to Chapter 3. More importantly, The new model is a competitive exercises in which the objective is to obtain a high score and players must apply subject matter or other relevant knowledge in an effort to advance in the exercise and top all other player, i.e Rank 1 on the leaderboard.

Furthermore, this new gamified mode suits the purpose of practice and/or refine already-acquired knowledge and skills, i.e all the skills from the original simulation. and it could potentially used to identify gaps or weaknesses in knowledge or skills, thus when a player received a low score in this new model, then this player must have a weakness in the acquired knowledge or skills. And with the help of game log, it the weakness can be identified through log analysis. Next, a summation of scores and review will be shown at the end of the game. And lastly, this game could possibly help players to develop new relationships among concepts and principles through the game play. Furthermore, if the random ore pile mechanism was implemented, games could be somewhat challenging and interesting for the players too.

This design not only follows the four purposes of academic games, but also meets the five important design criteria in Game design. First scores and ranking is based only on the demonstration of knowledge or skills. Second, important contents are addressed at the beginning of each game modes. Third, the dynamics of the game is easy to understand and somewhat interesting for the players but not obstruct or distort learning. it provides a practical, yet challenging exercise with the scoring mechanisms and collision damage system. Next, players will not lose points for wrong actions, instead, they just don't get any points, however, there are more works need to be done at balancing rewards and punishments. Lastly, both models are not zero-sum exercises. All players will receive some point trough the game, and no matter what score they got at the end, as long as they want, that can all be part of the leaderboard.

With reasonable defence shown above, this project has successfully completed the requirement of gamifying the unity based simulator.

However, there are no user testing done on this project at this point. Thus, at this moment, this paper can not conclut on the result of impact on learning. But through the researches done on this model, the result may possibly be positive and pertantirally could be useful for real life training purposes.

REFERENCES

Alessi, S.M., & Troltip, S.R. (1991). Computer-based instruction: Methods and development. (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Brathwaite, B., & Schreiber, I. (2009). *Challenges for game designers*. Nelson Education.

Breda, L. (2008) “Invisible Walls,” in Game Career Guide, feature, Aug 19.

Crytek (2007) Crysis (PC) EA Games.

De Freitas, S. (2006). Learning in immersive worlds: A review of game-based learning. JISC e-Learning Programme.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining ‘‘Gamification’’. In Proceedings of MindTrek.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments* (pp. 9-15). ACM.

Dondlinger, M. J. (2007). Educational video game design: A review of the literature. *Journal of applied educational technology*, *4*(1), 21-31.

Dreifuerst, K. T. (2009). The essentials of debriefing in simulation learning: A concept analysis. *Nursing education perspectives*, *30*(2), 109-114.

Egenfeldt-Nielsen, S., Smith, J. H., & Tosca, S. P. (2015). *Understanding video games: The essential introduction*. Routledge.

Eldesouky, D. F. B. (2013). Visual hierarchy and mind motion in advertising design. *Journal of Arts and Humanities*, *2*(2), 148.

Fagerholt, E. & Lorentzon, M. (2009) Beyond the HUD. User Interfaces for Increased Player Immersion in FPS Games. Master thesis. Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg.

Gee, J. P. (2003). What vídeo games have to teach US about learning and literacy. Palgrave Macmillan.

Gredler, M. E. (2004). Games and simulations and their relationships to learning. *Handbook of research on educational communications and technology*, *2*, 571-581.

Habgood, M. P. J. (2007). The effective integration of digital games and learning content. PhD Thesis, University of Nottingham.

Hays, R. T. (2005). *The effectiveness of instructional games: A literature review and discussion* (No. NAWCTSD-TR-2005-004). NAVAL AIR WARFARE CENTER TRAINING SYSTEMS DIV ORLANDO FL.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., & Mackey, K. (1989). The xerox star: A retrospective. *Computer*, *22*(9), 11-26.

Keller, J. M. (1987). Development and use of the ARCS model of instructional design. Journal of instructional development, 10(3), 2.

Kent, S. L. (2010). *The Ultimate History of Video Games: from Pong to Pokemon and beyond... the story behind the craze that touched our lives and changed the world*. Three Rivers Press.

Klopfer, E., Osterweil, S., & Salen, K. (2009). Moving learning games forward: obstacules, opportunities and openness, the education arcade. Massachusetts Institute of Technology.

Kriz, W. C., & Hense, J. U. (2006). Theory-oriented evaluation for the design of and research in gaming and simulation.*Simulation & Gaming*, *37*(2), 268-283.

Leemkuil, H., de Jong, T., & Ootes, S. (2000). Review of educational use of games and simulations (Knowledge management Interactive Training System Project No. IST-1999-13078).

Li, H., Chan, G., & Skitmore, M. (2012). Visualizing safety assessment by integrating the use of game technology. Automation in construction, 22, 498-505.

Norman, D. (2014). *Things that make us smart: Defending human attributes in the age of the machine*. Diversion Books.

Pan, Z., Cheok, A. D., Yang, H., Zhu, J., & Shi, J. (2006). Virtual reality and mixed reality for virtual learning environments. *Computers & Graphics, 30*(1), 20-28.

Paras, B. (2005). Game, motivation, and effective learning: An integrated model for educational game design

Prensky, M. (2001). Digital-game based learning. McGraw-Hill.

Reigeluth, C., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16(1), 1-10.

Retro Studios (2007) Metroid Prime 3: Corruption (Wii) Nintendo.

R. Grewal (2016). COSC-2026E: User Interface Design, In class lecture notes [PowerPoint slides].

R. Grewal (2017). COSC 4926E: Human Computer Interaction, In class lecture notes [PowerPoint slides].

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational technology research and development*, *44*(2), 43-58.

Rogers, Y., Sharp, H., & Preece, J. (2011). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

Schell, J. (2014). *The Art of Game Design: A book of lenses*. CRC Press.

Shneiderman, B. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India.

Simões, J., Redondo, R. D., & Vilas, A. F. (2013). A social gamification framework for a K-6 learning platform. *Computers in Human Behavior*, *29*(2), 345-353.

Thurman, R.A. (1993). Instructional simulation from a cognitive psychology viewpoint. Educational Technology Research & Development, 41(4), 75-79.

Ubisoft Montpellier (2005) Peter Jackson's King Kong (Xbox 360) Ubisoft.

Ubisoft Montreal (2007) Assassin's Creed (Xbox 360, PS3) Ubisoft.

Van Wyk, E., & De Villiers, R. (2009, February). Virtual reality training applications for the mining industry. In *Proceedings of the 6th international conference on computer graphics, virtual reality, visualisation and interaction in Africa* (pp. 53-63). ACM.

Wastiau, P., Kearney, C., & Van den Berghe, W. (2009). How are digital games used in schools?: Complete results of the study, Brusel: European Schoolnet.

Weinberg, G., Harsham, B., & Medenica, Z. (2011, November). Evaluating the usability of a head-up display for selection from choice lists in cars. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (pp. 39-46). ACM

Wilson, G. (2006) “Off With Their HUDs! Rethinking the Heads-Up Display in Console Game Design,” in Gamasutra, feature, Feb 3.

Wilson, K. A., Bedwell, W. L., Lazzara, E. H., Salas, E., Burke, C. S., Estock, J. L., ... & Conkey, C. (2009). Relationships between game attributes and learning outcomes: Review and research proposals. *Simulation & Gaming*, *40*(2), 217-266.

Zichermann, G. (2011) 7 Winning Examples of Game Mechanics in Action. Retrieved 21.03.12 (Web log message).

Appendix - Game element & changes

Two functions separating UI settings between two game modes

```
public class Game_panel2 : MonoBehaviour {

    // Use this for initialization
    void Start() {
        //Debug.Log(globalVariables.currentRun.efficentMiming);
        if (globalVariables.currentRun.efficentMiming) {
            gameObject.SetActive(true);

        } else {
            gameObject.SetActive(false);
        }
    }

    // Update is called once per frame
    void Update() {

    }
}
```

```
public class Game_panel : MonoBehaviour {

    // Use this for initialization
    void Start () {

        //Debug.Log(globalVariables.currentRun.bonusMiming);

        if (globalVariables.currentRun.bonusMiming) {
            gameObject.SetActive(true);

        } else {
            gameObject.SetActive(false);
        }
    }

    // Update is called once per frame
    void Update () {

    }
}
```

Simple Countdown Timer

```
public class TimeCountDown : MonoBehaviour {
    public Text text;
    public static bool EndTimeTrigger = false;

    void Start(){
        EndTimeTrigger = false;
    }

    void Update() {
        float temp = (globalVariables.currentRun.MaxTime
            - globalVariables.currentRun.runtime());
//        Debug.Log("Time: " + temp);
        if (temp < 0) {
            temp = 0;
        }

        text.text = convertTime (temp );
        globalVariables.currentRun.checkTime();

    }
    public static string convertTime(float time) {
        int t = (int)time;
        int sec = (t % 60);
        int min = (t / 60);
        return ((min == 0) ? "" : (min.ToString("####.") + " <b>m</b> "))
            + ((sec == 0) ? "0" : sec.ToString("####.")) + " <b>s</b>";
    }
}
```

Game variables

```
public static float MaxHealth;
public static float CurrentHealth;
public float Score;
public float MaxTime;
public bool bonusMiming = false;
public bool efficentMiming = false;

[XmlArray("GameLog"), XmlArrayItem("Log")]
public List<String> GameLog = new List<string>();

[XmlArray("BGameLog"), XmlA public List ()
public List<String> BGameLo                g>();

[XmlArray("EGameLog"), XmlArrayItem("ELog")]
public List<String> EGameLog = new List<string>();
```

Default variables setting

```
MaxHealth = 1000f;
CurrentHealth = MaxHealth;
Score = 0f;
MaxTime = 1200f;
```

Checking Time on every update()

```
public void checkTime() {
    if (runtime() >= MaxTime) {


    if (!TimeCountDown.EndTimeTrigger) {

        if (efficentMiming) {
            Score += CurrentHealth;
        }
        endLevel ();
    }

    }
}
```

Applying Score on Ore collection

```
GameObject score = GameObject.Find("Score"); ///warning
if (score != null) {
    this.Score += val;
    score.GetComponent<Text>().text = ((int) Score).ToString();

    String log = "Score From Ore " + val;

    GameLog.Add (log);

}

//Debug.Log (Score);
```

Calculate Hit and apply Hid Damage

```
public void addHit(HitInformation HI, float Damage)
{
    //make sure that the level has started.
    if(startTime != 0)
    {

        HitInfo.Add(HI);
        //Hit damage = 20 per hit  + current speed
        float temp = 20 + HI.speed;
        CurrentHealth -= temp;
        // this.Damage += Damage;
        GameObject.FindObjectOfType<Computer>().refreshDamage();

        String log = "Damage From Collison " + temp;

        GameLog.Add (log);

        if (!infinitCollisions)
        {

            if (this.Damage >= this.MaxDamage) {
                endLevel();

            } else if (HitInfo.Count >= this.MaxDamage) {
                endLevel();
            }
        }

        if (bonusMiming) {
            if (CurrentHealth <= 0f) {

                endLevel();

            }
        }

    }
}
```

End Level function and append score to leaderboard

```
public void endLevel()
{
    TimeCountDown.EndTimeTrigger = true;

    endTime = Time.time;
    globalVariables.globalVar.switchToScene("Feedback");

    //End Leve - save end time
    GameLog.Add ("End Time " + endTime);

    // read learboard data , display and update
    fullLeaderData.readLeader ();

    if (globalVariables.currentRun.efficentMiming) {

        fullLeaderData.Eappend(new leaderInfo (globalVariables.player.playerName, (int) Score));

    }

    if (globalVariables.currentRun.bonusMiming) {

        fullLeaderData.Bappend( new leaderInfo (globalVariables.player.playerName, (int) Score));

    }


    Debug.Log (fullLeaderData.Eleaderboard.Count);

    fullLeaderData.writeLeader ();   //(moved to ResulrHandler)
}
```

Applying Score to Reaction timer

```
public void addReaction(ReactionTimeInformation info){
    reactionTime.Add (info);
    GameObject score = GameObject.Find("Score"); ///warning
    if (score != null) {

        float temp = 20 - (info.tagTime - info.viewTime);
        this.Score += temp; /// balance score!!!!!!!
        score.GetComponent<Text>().text = ((int) Score).ToString();

        String log = "Score From reactionTime " + temp;

        GameLog.Add (log);
    }
}
```

Saving User information and both game logs

```
public void Save()
{

    try
    {
        globalVariables.settings.save(exportDataPlaceHolder.text,
            globalVariables.player.playerName
            + "-" + globalVariables.currentRun.date.Day.ToString("00")
            + globalVariables.currentRun.date.Month.ToString("00")
            + globalVariables.currentRun.date.Year);
        globalVariables.currentRun.saved = true;
        savedVisual.SetActive(true);

    }catch
    {

    }

    if (globalVariables.currentRun.bonusMiming) {
        Log_Read_Write.writeXML<List<String>> (globalVariables.currentRun.GameLog,
            "BonusGameLog" + globalVariables.player.playerName + "-"
            + globalVariables.currentRun.date.Day.ToString ("00")
            + globalVariables.currentRun.date.Month.ToString ("00")
            + globalVariables.currentRun.date.Year + ".xml");

    }

    if (globalVariables.currentRun.efficentMiming) {

        Log_Read_Write.writeXML<List<String>> (globalVariables.currentRun.GameLog,
            "EfficentGameLog" + globalVariables.player.playerName
            + "-" + globalVariables.currentRun.date.Day.ToString ("00")
            + globalVariables.currentRun.date.Month.ToString ("00")
            + globalVariables.currentRun.date.Year + ".xml");


    }
```

Applying Leaderboard Information to the Leaderboard UI

```
void Start () {

    if (globalVariables.currentRun.efficentMiming) {

        int c = MaxCount < fullLeaderData.Eleaderboard.Count
            ? MaxCount : fullLeaderData.Eleaderboard.Count;

        for (int i = 0; i < c; i++) {
            PlayerID [i].text = fullLeaderData.Eleaderboard[i].playerID;
            PlayerScore [i].text = fullLeaderData.Eleaderboard[i].playerScore.ToString();
        }
    }

    if (globalVariables.currentRun.bonusMiming) {

        int c = MaxCount < fullLeaderData.Bleaderboard.Count
            ? MaxCount : fullLeaderData.Bleaderboard.Count;

        for (int i = 0; i < c; i++) {
            PlayerID [i].text = fullLeaderData.Bleaderboard[i].playerID;
            PlayerScore [i].text = fullLeaderData.Bleaderboard[i].playerScore.ToString();
        }
    }

    Rank.text = RankNumber.ToString ();
}
```

Learboard class

```
public static List<leaderInfo> Bleaderboard = new List<leaderInfo>();
public static List<leaderInfo> Eleaderboard = new List<leaderInfo>();
```

Append Score to the Score list

```
public static void Bappend (leaderInfo a){

    for (int i = 0; i < Bleaderboard.Count; i++) {

        if (a > Bleaderboard [i]) {

            Bleaderboard.Insert (i, a);

            Leader.RankNumber = i + 1;

            //Debug.Log ("ADDED");


            return;
        }

    }

    Bleaderboard.Add (a);
    Leader.RankNumber = Bleaderboard.Count;

    //Debug.Log ("ADDED");

}

public static void Eappend (leaderInfo a){

    for (int i = 0; i < Eleaderboard.Count; i++) {

        if (a > Eleaderboard [i]) {

            Eleaderboard.Insert (i, a);

            Leader.RankNumber = i + 1;
            return;
        }

    }

    Eleaderboard.Add (a);
    Leader.RankNumber = Eleaderboard.Count;
}
```

Read and Write to leaderboard

```
public static void readLeader(){

    Eleaderboard = Log_Read_Write.ReadXML<List<leaderInfo>> ("EfiicentLeaderBoard.xml");


    Bleaderboard = Log_Read_Write.ReadXML<List<leaderInfo>> ("BonusLeaderBoard.xml");

}

public static void writeLeader(){


    Log_Read_Write.writeXML<List<leaderInfo>> (Eleaderboard, "EfiicentLeaderBoard.xml");


    Log_Read_Write.writeXML<List<leaderInfo>> (Bleaderboard, "BonusLeaderBoard.xml");
}
```

HP bar Display

```
public class HP_BAR_DISPLAY : MonoBehaviour {

    public Text PercentHP_Text;
    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        float percentHP = runData.CurrentHealth / runData.MaxHealth;
        GetComponent<Slider>().value =  percentHP * 100;

        int HP = (int)(percentHP * 100);

        if (HP <= 0) {
            HP = 0;
        }

        if (globalVariables.currentRun.bonusMiming) {

            PercentHP_Text.text = HP  + "%";
        }
        if (globalVariables.currentRun.efficentMiming) {

            PercentHP_Text.text = HP + "% Bonus";
        }


    }
}
```

New game object added on the Mission setup UI

```
public Toggle bonusMiming;
public Toggle efficentMiming;
public InputField game_Time;

public GameObject Time;
public GameObject minute;
public GameObject inc;
public GameObject dec;
```

Bonus Mining Mode Mission setup

```
public void BonusMiming() {

    bool on = bonusMiming.isOn;
    globalVariables.currentRun.bonusMiming = on;



    if (on) {
        globalVariables.currentRun.infinitOre = true;
        globalVariables.currentRun.infinitCollisions = true;
        collissionField.gameObject.SetActive(false);
        OreField.gameObject.SetActive(false);
        reactionReset.gameObject.SetActive(false);
        infinitColission.gameObject.SetActive(false);
        infinitOre.gameObject.SetActive(false);

        game_Time.gameObject.SetActive(true);
        Time.gameObject.SetActive(true);
        minute.gameObject.SetActive(true);
        inc.gameObject.SetActive(true);
        dec.gameObject.SetActive(true);

        efficentMiming.gameObject.SetActive(false);

        for(int i = 0; i < minnerToggles.Length; i++)
        {

            minnerToggles[i].gameObject.SetActive(false);
            minnerToggles [i].isOn = false;


        }


    }
```

```
    else {
        on = infinitOre.isOn;
        globalVariables.currentRun.infinitOre = on;

        on = infinitColission.isOn;
        globalVariables.currentRun.infinitCollisions = on;

        collissionField.gameObject.SetActive(true);
        OreField.gameObject.SetActive(true);
        reactionReset.gameObject.SetActive(true);
        infinitColission.gameObject.SetActive(true);
        infinitOre.gameObject.SetActive(true);


        game_Time.gameObject.SetActive(false);
        Time.gameObject.SetActive(false);
        minute.gameObject.SetActive(false);
        inc.gameObject.SetActive(false);
        dec.gameObject.SetActive(false);



        efficentMiming.gameObject.SetActive(true);

        for(int i = 0; i < minnerToggles.Length; i++)
        {


            minnerToggles[i].gameObject.SetActive(true);

        }
    }
}
```

Efficient Mining Mode Mission setup

```
public void EfficentMiming() {

    bool on = efficentMiming.isOn;

    globalVariables.currentRun.efficentMiming = on;
    if (on) {

        globalVariables.currentRun.infinitOre = true;
        globalVariables.currentRun.infinitCollisions = true;
        collissionField.gameObject.SetActive(false);
        OreField.gameObject.SetActive(false);
        reactionReset.gameObject.SetActive(false);
        infinitColission.gameObject.SetActive(false);
        infinitOre.gameObject.SetActive(false);

        bonusMiming.gameObject.SetActive(false);

        game_Time.gameObject.SetActive(true);
        Time.gameObject.SetActive(true);
        minute.gameObject.SetActive(true);
        inc.gameObject.SetActive(true);
        dec.gameObject.SetActive(true);

        for(int i = 0; i < minnerToggles.Length; i++)
        {


            minnerToggles [i].isOn = true;

        }



       for(int i = 0; i < LHDToggles.Length; i++)
       {


           LHDToggles [i].isOn = true;

       }
```

```
} else {
    on = infinitOre.isOn;
    globalVariables.currentRun.infinitOre = on;

    on = infinitColission.isOn;
    globalVariables.currentRun.infinitCollisions = on;


    collissionField.gameObject.SetActive(true);
    OreField.gameObject.SetActive(true);
    reactionReset.gameObject.SetActive(true);
    infinitColission.gameObject.SetActive(true);
    infinitOre.gameObject.SetActive(true);

    bonusMiming.gameObject.SetActive(true);

    game_Time.gameObject.SetActive(false);
    Time.gameObject.SetActive(false);
    minute.gameObject.SetActive(false);
    inc.gameObject.SetActive(false);
    dec.gameObject.SetActive(false);

    for(int i = 0; i < minnerToggles.Length; i++)
    {


        minnerToggles [i].isOn = false;

    }



        for(int i = 0; i < LHDToggles.Length; i++)
        {


            LHDToggles [i].isOn = false;

        }
    }
}
```

Max time setup

```
public void setMaxTime() {
    int time = Int32.Parse(game_Time.text);
    if (time < 1) { time = 60; game_Time.text = time.ToString(); }
    if (time > 60) { time = 1; game_Time.text = time.ToString(); }
    globalVariables.currentRun.MaxTime = time*60f;

}
```

Incrementing and decrementing function for time setup

```
public void incTime() {
    int time = Int32.Parse(game_Time.text);
    time++;
    game_Time.text = time.ToString();
}

public void decTime() {
    int time = Int32.Parse(game_Time.text);
    time--;
    game_Time.text = time.ToString();
}
```

File IO - Read & Write

```
public class Log_Read_Write {

    public static T ReadXML<T> (string filePath) {
        XmlSerializer serializer = new XmlSerializer (typeof(T));
        FileStream stream = new FileStream (Application.dataPath + "/Game_Log/"
            + filePath, FileMode.Open);
        T Result = (T)serializer.Deserialize (stream);
        stream.Close ();
        return Result;
    }

    public static void writeXML<T> (T file, string filePath) {
        XmlSerializer serializer = new XmlSerializer (typeof(T));
        FileStream stream = new FileStream (Application.dataPath + "/Game_Log/"
            + filePath, FileMode.Create);
        serializer.Serialize(stream, file);
        stream.Close ();

    }




}
```