VIRUS SURVIVOR
Computer Game

By

SHUBHANGI KALYANI BIRAJDAR

Student ID: 2032649

A thesis submitted to
the University of Birmingham
for the degree of

MASTERS OF ADVANCED COMPUTER SCIENCE

Supervisor: Mr Martin Escardo

School of Computer Science
University of Birmingham
September 2020

ABSTRACT

The plague has harmed humanity many times. One of the most significant challenges throughout a pandemic is spreading awareness amongst individuals. Awareness helps control the pandemic. Covid-19 has brought great misery to humanity. When considering the spread and severity of this infection, it is important to educate individuals about various safety guarantees. This way of educating people about this disease often manifests itself in online advertisements or physical awareness campaigns, media advertisements, Etc. Not to mention when Covid-19 occurs, there are not that many mobile apps and games designed to draw attention to pandemics. With a global shutdown mode, the game breathes well indoors. Therefore, I propose Virus Survivor, a 2D resistant game. The aim is to educate individuals on how to protect themselves from the Covid-19 virus at home, taking into account social isolation and the use of pads and disinfectants in games.

The design behind this game 'VIRUS SURVIVOR' is motivated by all-time favorite and popular video-games like "MARIO", "DANGEROUS DAVE" and "LOST". This game is based on 2D java programming language. It has missions such as shoot an enemy, collect foods and superpower to gain ability to become more strong and many more challenges. This game has implemented by using Java AWT GUI library that handles Animation, manages controls with Event Handling, Import sprites to render assets inside the game and so on. Virus Survivor game is user friendly and play it as an entertainment purpose to spread awareness of Covid-19 virus.

ACKNOWLEDGMENTS

I would like to thank my supervisor Mr Martin Escardo who encouraged me and guided me for this project. Special thanks to the University of Birmingham, School of Computer Science department staffs and teachers who gave me the golden opportunity to create an outstanding project on gaming called VIRUS SURVIVOR. I also thank Mia, from the welfare department, who helped me and supported me when I was unwell and at a bad time.

I am really lucky and say thanks to my family, who supported me and motivated me through this journey and became a part of this project. I would also thanks to my friends, who helped me and gave me some feedback for this project.

Contents

List of Figures

Chapter One

Introduction

In many cases, some diseases pose a threat to people around the world. Some of the infections can be referred to as an epidemics and others as epidemics based on the rate and area of transmission. According to the rules defined in WHO 1, a pandemic period, usually influenza, begins when similar viruses spread via networks in several countries in different WHO regions(Sklyarova and Volkova, 2019). There are few epidemics in the world like cholera and smallpox. One of the most important exercises gained from these epidemics is the struggle for recognition. Awareness campaigns have been found to help reduce the risk of infectious diseases such as hospitalisation and transportation.

Most governments and professionals around the world have set the requirements for developing public health education and awareness campaigns, indicating improved health education for individuals(Li and Cheng, 2020). Health professionals in some countries have been asked to practice some surveillance exercises, including a basic disinfection room, to differentiate and isolate prognostic cases and to lead the public into a conscious battle. In addition to the necessary clinical considerations, it is vital to know the safety measures that everyone in the world must follow during a pandemic(Dolan, Whitfield, and Andino, 2018).

For some reason, Wuhan, China suffered a severe attack of pneumonia in December 2019. The cause was later identified for the new coronavirus, which is a very severe coronavirus with respiratory disease. This SARS-Cov-2 infection was named Covid-19 by the World Health Organisation in February 2020(Morin-Simard, 2016). Covid-19 has spread quickly and affects to some countries in the world. As of April 16, 2020, it was detailed by the World Health Organisation. Covid-19 side effects include fever, upset stomach, shortness of breath, lymphopenia, and weakness. It is believed to compromise the safety system against injury, thus increasing complications in patients with diabetes, cardiovascular disease, and persistent kidney disease, leading to cardiovascular collapse, kidney disappointment, disappointment in breathing, and even the occasional death(Radchenko, 2020).

The death volume caused by the Covid-19 virus increased from 6,600 to 130,800 in the period from March 16, 2020 to April 16, 2020. The World Health Organisation and several countries around the world are taking several steps to control the Covid-19 pandemic. These steps include post-action analysis and better communications, expanded health centres, and expanded medical equipment that can help identify drugs to treat Covid-19 and raise public awareness. Some governments around the world do not have to block policies to prevent the spread of disease through open social relationships. The Covid-19 detection program is implemented in different ways around the world. The public is constantly being guided through telephone and television commercials, physical and online fights, open space flyers, numerous web sites, and more to remind themselves of the supportive measures to be followed(Wilson et al., 2018).

Several locations are working on Covid-19 to provide data on security measures and Covid-19 status. Many countries around the world quickly have lockout policies in place that allow individuals to purchase essential daily items that they can leave. When people are in the daytime, it is very important to follow the safety precautions. Today's youth invest a lot of energy in their cell phones with different applications and different cell phone games. Similarly, it has been observed that games can affect the behaviour and behaviours of gamers. Despite the extensive public health education program So, I propose Virus Survivor, a game based on 2D single-player resistance. The plan is to inform the player about various welfare measures that have been taken against Covid-19 in his absence(Geraci, Recine, and Fox, 2016).

1.1 Related Work

Spreading an open awareness of various diseases plays an essential role in maintaining individual health. Healthcare mobile apps and websites are rarely built because the extension is used across multiple apps and web apps(Down, 2016). Wearable applications have been developed to help with a variety of illnesses such as weight, cardiovascular infections, and persistent diseases such as skin diseases. This study introduced SunFace, a wearable facial maturation application. This application should suppress the growth of malignant skin. SunFace recommends customers click selfies to see the changes caused by skin maturation for 5 and 25 years with and without sunscreen. Besides, it provides data on most cases of malignant skin growth, usually caused by UV light, and the likelihood of developing skin conditions from various practices. The SunFace application evaluation test of approximately 350 elected Brazilian students in the school added to their reflections on defensive UV practices(Katsaridou and Thibault, 2016).

The CAIDE Dementia Risk Score application was created by experts to predict the risk of developing dementia based on a person's profile. It also leads people to reduce their risk of dementia. The individual's medical history, health status, and current health behaviour are considered contributors throughout the study, and the level of risk is determined from this data(Morin-Simard, 2016). In addition, it gives clinical professionals the ability to study a person's risk of dementia and provide basic guidance. He evaluates the acceptance of food by a customer based on three images that were clicked on from three different perspectives or on the basis of a short video clip of the food used.

The types of food included in a photo or video are aggregated by adjusting the highlighting effects and how they interact with different types of food stored in the food and diet database. The food size is recognised by calculation adjustment and 3D reconstruction(Sirparanta, 2018). The customer is provided with an estimated number of calories burned based on the type of food and the size produced. This way, you can scan your calorie intake and order your diet(Sirparanta, 2018).

Research has also found that over 70% of weight loss customers have reduced mobile application usage and have effective weight loss. Therefore, these reviews show positive results for the use of various applications in the health sector. IPhysioMeter is suggested as the standard software for cell phones and heart rate. The HR and NPV values are determined by the software when the customer clicks on the iPhone main screen and unfolds the CMOS camera with the LED light and display on the left(Søndergaard, 2018). The estimated results will be sent to user's email address. Some mobile games have also been developed to improve health monitoring of various infections. MobileQuiz was designed as a mobile game to enhance the more established adult physical and intellectual experience. Send the task to the customer with the physical separation in mind to make sure the customer can address it. Upon completion, customers are rewarded centrally and receive specific surveys in the area(Ananda and Chusyairi, 2019).

This study suggests SnackBeaker, a multipurpose game designed to improve healthy eating choices amongst gamers. It offers a test that can be approached based on either of two decisions regarding the nutritional benefits of the other two options. In this way, the client will be informed of the health benefits of various anecdotes and will help in choosing high meal snacks(Ntokos, 2020). Diabetes Mario, a mobile game based on Mario's style, has been identified as a tool for understanding diabetes control. The players monitor the glucose level in diabetic Mario and specially choose the foods presented in the game, which will calm the princess in the long run(Ntokos, 2020). Given the importance of external robotic defibrillators in out-of-hospital heart failure, make general changes to automated external defibrillators and convince them to report cases of epilepsy.

A multipurpose game is being developed to be obtained. Few locations or phone lines have been set up given the importance of open minds and the exploitation of pandemics, monitoring plans and responses to combat epidemics. ISIS was created as a web framework to plan and respond to epidemics by examining different versions of potential mediation(Azevedo and Sarinho, 2019). For example: "Organisation of vaccinations for part of the population, use of antivirals as treatment, closure of schools and workplaces, establishment of permits, Etc. Also, the World Health Organisation published several welfare measures to raise awareness during the epidemic: most of the applications and websites were created to track detailed Covid-19 cases and to spread awareness about Covid-19(Wilson et al., 2018). This study created an online panel that outlines the sometimes stated views on Covid-19 cases. He reports on a case of Covid-19 pandemic at the national level.

User around the world relies heavily on 4,567 different mobile apps designed to raise awareness of risks, best practices and important warnings about Covid-19. Given the positive impact of multipurpose games and apps on healthcare and their proven capability to raise awareness of various infections through games and apps for mobile devices, I have developed these games for PC to support the COVID pandemic -19(VIRUS, 2017). However, I do not know of any mobile and PC games like Virus Survivor that promote an open awareness despite the potentially massive application of the Covid-19 regulation(Søndergaard, 2018). Following Figure1.1 estimates the infrastructure of VIRUS SURVIVOR game.

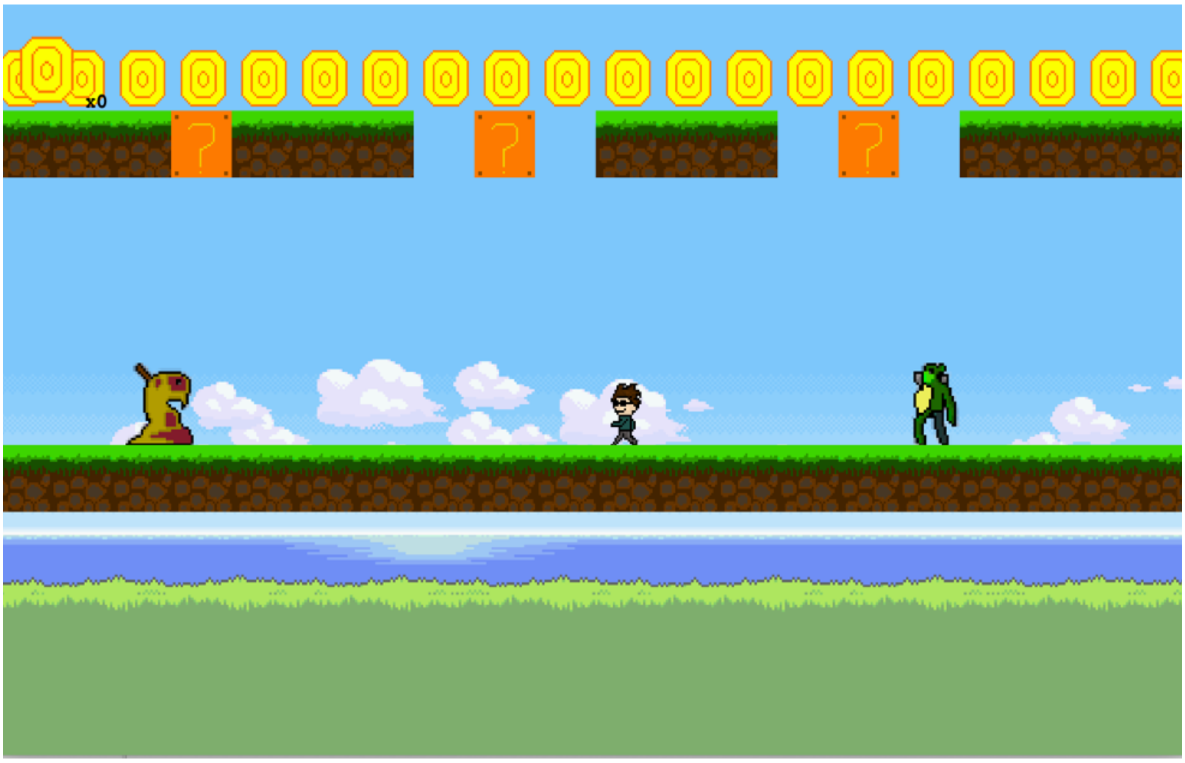


Figure 1.1: Infrastructure of VIRUS SURVIVOR game

1.2 Aim and Objectives

This study aims to evaluate the development, designing and implementation of computer game using the Agile method. The objectives of the study are:

- To explore the game development methods
- To identify critical issues whilst implementing the developed game

Chapter Two

Literature Review

Creating a game, whether it is a relaxation game or an original game, is an unpredictable step in designing, programming, creating content, and testing games. You can save some of your game development efforts, your time and your money by reusing existing game items, which you can access through various online portals such as the Unity Asset Store, Guru and Game Salad Surprisingly, most of the game items submitted have three main disadvantages(Morin-Simard, 2016). First, most articles are media objects (such as regions, audio, structures, weapons, articles, and user interface formats) that you can program whilst reduce the workload of content creation. Second, it does not separate relaxation from actual play, but items rarely show beneficial benefits(Morin-Simard, 2016).

RAGE, the European Commission's Horizon 2020 exploration and development project, addresses these issues by providing access to many original game programming components that are available in a variety of game engines, game phases, and programming languages. It offers up to 40 reusable programming components and implements a variety of functions tailored to the way the game is taught in real life(Geraci, Recine, and Fox, 2016). Components include exploration of player data, detection of emotions, assessment of confidentiality, personalisation, game modifications, procedural animations, language and multiple tests, intuitive narration, social manipulation, and a variety of skills(Søndergaard, 2018). The European Commission supports the RAGE project.

The RAGE project is building a realistic and thriving gambling industry that will fuel creative job creation and help address a wide variety of cultural challenges in education, health, social relationships, and citizenship. Similarly, the ODP methodology is in line with the European Digital Single Market strategy, which is one of the main concerns of the European Commission—promoting Europe's position as a world leader in the computer industry, open guidelines, open data and access support(Geraci, Recine, and Fox, 2016). Also, RAGE meets reasonable competitive standards as its open and versatile programming components help game engine resellers to overcome the blockade, apply business models and set prices. This study introduces the first selection of programming components and explains its structure and causes. It contains components for emotion recognition, performance statistics, personalised speech processing and data storage. Before discussing these game programming components in detail, first design the overall architecture(Radchenko, 2020).

2.1 The RAGE Component-Based System Architecture

The motivation for the RAGE component architecture is to consider easy integration of reusable programming modules on many different game engines and levels. This architecture integrates a service-oriented architecture that web administration uses for messaging with remote experts through HTTP contracts and a client-side architecture of the fully-coordinated components of the game engine(Sirparanta, 2018). This approach is expected to bypass these SOA limitations and client-side deployments. In general, services have some favourable terms. For example, the separation of implementation accuracy and severity of reuse of departments(Orji et al., 2018).

As a disadvantage, SOA can degrade system performance due to normal system calls and additional costs and restrict the department designation and Organisation by management buyers. Also, players need to be on the internet all the time, which is usually not the case. The RAGE client architecture relies on a limited number of static programming examples and coding exercises to separate discussion from implementation. Removes the requirement for external programming systems and limits code that should not be included in the game engine code. However, when client-side components require extensive preparation, remote departments can be an excellent alternative to maintaining a strategic distance from insufficient gaming performance on neighbouring devices. The client components are mainly based on C# and JavaScript / TypeScript and cover most of the known developments(Ananda and Chusyairi, 2019).

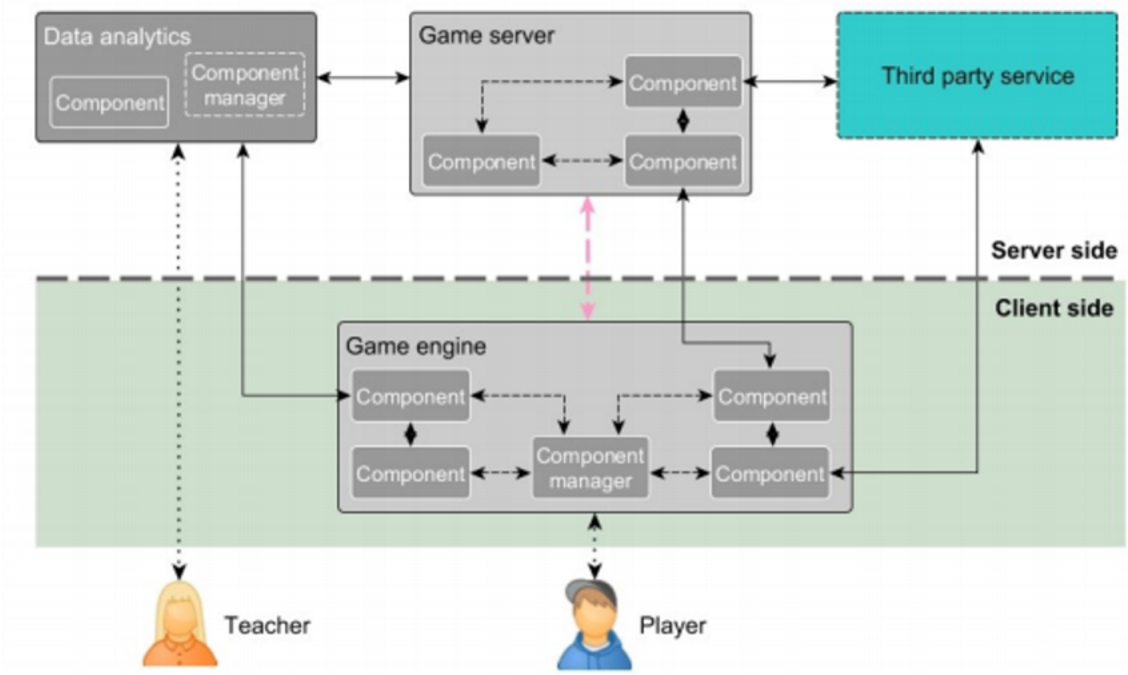


Figure 2.1: The RAGE Component-Based System Architecture

It looks like the game engine and potential game elements have faded. The blurry square shape shows a programming component of RAGE(Geraci, Recine, and Fox, 2016). The RAGE Data Investigation Factor is a fundamental elementary aspect system that collects collaborative data from a population of players for evaluation by coaches and teachers. This agent takes over the validation and approval and acts as a ticket broker for central components and departments. It contains the main components for webmaster recruitment. RAGE components are created by the customer's local component administrator(VIRUS, 2017). Both client-side and operator-side RAGE components can work together legally. The component on the factor side can either remember the game factor or use a different factor. External web departments can be accessed from client-side and worker-side components(Sirparanta, 2018).

2.2 The Emotion Detection Component

Emotions are exceptionally applicable elements in both play and study. Emotions can have a significant impact on psychological behaviour as emotions coincide with memory and activity(Radchenko, 2020). Emotions and discrimination generally speak of a deeply coordinated and intelligent behaviour, which, according to neuroscience, is compatible with the dangerous degree of availability between nerve tissue and the brain system. So far, however, the player model of the original game systematically excludes the player mood due to recognition problems and limitations. Advances in artificial intelligence and the recognition of examples have enabled practical and moderate exposure to emotion. With the RAGE Emotion Detection component, game developers can effectively remember the emotions of the players in the game(Down, 2016). This client-side component uses video broadcasts from the player's webcam to continuously detect emotions about the appearance of the player.

The emotion finding component replicates the motives for one of the six main emotions and fair emotions. Feedback estimation can be used in a number of ways to improve the quality of learning in real games(Søndergaard, 2018). First of all, following the player's mindset whilst playing the game can improve the suitability and customisation of the game by, for example, providing ideas, guidelines, comments, and other tips to aid the game. Second, proper monitoring of player mood can help evaluate and investigate customer encounters during the game, as well as identify and resolve cases and bottlenecks in the game(Down, 2016). Third, if emotions are part of the learning content (e.g. preparing a potential meeting with a co-worker, preparing an agreement, or preparing social skills), use the emotion recognition component to express emotions. To make and donate the authority of the player.

2.3 The Performance Statistics Component

Due to the regular development of teaching and preparatory data sets, many educators and researchers see learning tests as a way of improving the quality of learning. The real game provides a personalised data path that reflects the decision making, practice, and performance of the player. This concept makes the original game an excellent target for studying in college. In any case, because of their myriad of needs, educators and game engineers are unpredictable(Li and Cheng, 2020). First, the variety and the amount of the recorded data are tested with a view to the selection of the sections to be selected and the practical application. Second, proper use of realistic methods requires point-to-point information about techniques and their limitations(Radchenko, 2020).

Third, hidden assumptions and standard requirements and pitfalls associated with statistics can make it challenging to understand measurable results. The RAGE Performance Statistics component provides educators and game designers with tools to help them overcome these problems. This component requires vital data on the player's performance as a source of information and lots of realistic progress in player learning, reinforced by legality checks, significant events and translation rules. Of course, generate various indicators. The Performance Statistics component comprises two types of information metrics that are commonly used to indicate learning progress(Søndergaard, 2018).

The main metric is the time it takes to complete the task, and the second metric is the performance attribute of the task. Engagement is a very well known game exercise that is distinctive at the beginning and at the end and can be identified wherever your learning goals are. There may be inconsistent progress amongst students, as some students may complete in-game assignments, and others may require other presentations(Sklyarova and Volkova, 2019). The guide must process this data so that it can provide further assistance and comments if needed. The performance statistics component takes over the operator side and gives users access to all player data at the population level. The necessary knowledge is gained from associations at the population level in order to provide references for individual students(Radchenko, 2020).

After the game data has been sent to the operator, the performance statistics component carries out measurable surveys and presents the results via a web interface in a uniform or graphical structure, including descriptive messages. At the individual level, each student's score is in contrast to the grouping mean and standard deviation, but a large deviation triggers a warning. In addition, each student is self-diagnosed if different presentations need to be made in a similar company. The learning curve can be seen by fitting the regression curve based on student grades or time spent(Down, 2016). The second level of the exam is at the assembly level. For example, different meetings such as schools, groups and classes are taken into account. The progress of the playoff rally can be determined from the individual data of the player. The T-test is used to identify discrepancies between clusters. In addition, this component supports the third survey.

Usually, you can see the students by summarising the performance of the assignments at different stages of implementation. Deviations from the ideal or average can indicate a problem. For example, a commercial drift can cause similar problems in the work you are doing. Also, large standard deviations as the task progress can be confusing or indicate a wrong definition(Geraci, Recine, and Fox, 2016). The Performance Statistics component provides hands-on preparation for realistic alternatives that will help teachers gain knowledge of student learning performance and provide appropriate incentives to identify uneven progress Trainers and engineers do not need useful historical statistics to use this component. The performance statistics component is built into Java and JavaScript(Bosman, 2020). The first application is being tested for a long time whilst the game is being prepared for a Dutch technical school.

2.4 Language Technology Services

For a long time, computer-based learning tools have not been able to prepare printed papers for students. Instead, they had to get clear feedback on many decision questions. As a result, students were asked to spot and select the correct answer in limited order rather than conveying their thoughts in their usual language(Søndergaard, 2018). This has great implications as the dynamic use of language is strictly defined by internal and external intellectual understanding and interaction. Languages are a fundamental part of learning as they are used in the context of data transmission and collection. Recently, natural language processing strategies have found an accurate, effective, and reliable way to analyse speech and are increasingly used(Azevedo and Sarinho, 2019).

So far, however, there have only been minimal and light training applications. Advances in language innovation will allow the RAGE project to bring more NLP departments right into the original game. These departments recommend various updates to the original game design(Bosman, 2020). All of this relates to the immediate investigation and understanding of a player's verbal or structured cues in collaboration with the game and the individual player. Besides, our language department promotes a wide range of normative situations that cover, for example, computer-aided assessments of summaries of grade expectations, resumes and assessments of cover letters as broad and promising and improve style. We make individual suggestions for you, such as the authorship(Ntokos, 2020).

NLP practices focus on different parts of the language and form the basis of computer tests that have been identified in a specific task or area. The tool identifies various semantic highlights that are important in preserving literature and predicting student understanding, e.g., a study of literary relevance, printed complexity, and assumptions often referred to as meta-complementation. In doing so, abstract data is clearly removed from the message(Rita et al., 2020). The department is based on the ReaderBench architecture and offers computerised, multilingual summary forms that are closely related to dissertations and story-type news(Zan, 2018).

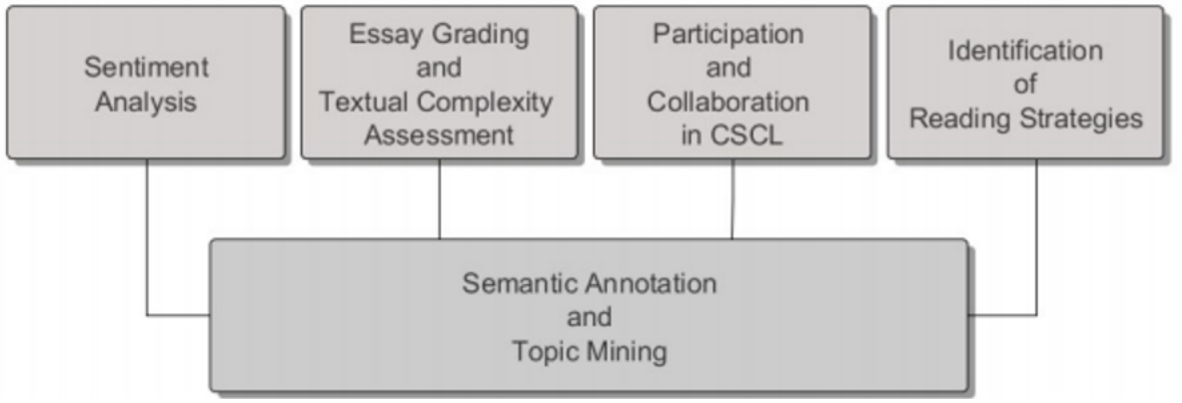

Sentiment Analysis
Essay Grading and Textual Complexity Assessment
Participation and Collaboration in CSCL
Identification of Reading Strategies
Semantic Annotation and Topic Mining

Figure 2.2: Language Technology Services

The system takes into account the prediction and evaluation of perception for both individual and group learning. Perceptual expectations measure the impressions of the material and the multidimensional nature of the problem, whilst understanding the perception implies the later preparation of the student for the product, e.g., Participation in a report, answer, or discussion. The centre of our linguistic research is a consortium(Morin-Simard, 2016). This can be considered as a collection of vocabulary, languages, and semantic links that work with the sections printed together. Attachments are of course calculated based on the semantic separation of the lexical ontology, the vector space of the latent semantic analysis and the Dirichlet attributes.

The last two semantic models should be built in ad-hoc groups based on language and spatial characteristics. In addition, a specially developed pipeline for individual language preparation offers various methods for notation, separation, selection of word references only, removal of word breaks, derivation, grammar marking, the credibility of the analysis, stemming and recognition—supported recognition of named subjects. At RAGE, we use mood tests to assess the state of mind, as evidenced by cover letters that students are required to fill out on the recruitment game recruitment form(Radchenko, 2020). Second, automated paper reviews relate to assessments of student writing in reports and responses to open inquiries. The department helps identify obvious defects on the paper expansion of the formula, use of language drivers, lack of correlation between neighbouring formulas), taking into account that personalised notes are provided to the students. Such comments have been shown to have a significant positive impact on style quality(Geraci, Recine, and Fox, 2016).

A multifaceted literature review involves many factors. For instance, ideal quality of clarity, surface measurements commonly used in other programmed paper scoring systems, punctuation, semantics and language. It is important that our model is extensible, covers many languages, is highlighted in attachments, and provides creative files such as Union Stream Record and Age of Exposure. Third, discussion polls are used to support mentors in a lengthy analysis of discussions on the web. Two interactive algorithm models were used to evaluate investments and collaboration. The model carries out longitudinal studies of the ongoing discussion and records the efforts that have been coordinated towards building social information(Li and Cheng, 2020).

The posting model is slang but uses an alternate phrase of speakers to help explore the discussion. Fourth, the use of perceptual behaviour in student self-declaration is often seen as a limiting factor in assessing classes. This RAGE component naturally characterises the systems used, including metacognition, causality, crossover, reformulation, and illustration. A recognised action profile is a reliable indicator of a person's perceived level. On the other hand, the accuracy of the estimated predictions can be improved mainly by including various printed natural records. All components are made available as REST web resources and can be easily integrated into dedicated external applications that might compromise the functions provided by web developers(Sirparanta, 2018).

2.5 Purposeful Game Development

In the product development industry, in addition to their actual purpose in relation to various areas such as education, business, and health, programming games are gaining attention as they are used for transformation. The original game was designed to influence intended interests, such as a fun game, but appears to be combined with a realistic scale. You need to speak to an attractive, broad, and objective audience. Especially with real games, the income continues to increase with their importance in various areas. In 2012, game programming generated sales several times that of any other programming component. Development plans are distinguished as a systematic approach to achieving the goal of creating business components within and on time in financial plans. Various methods are used to develop and design games. The first method is cascading, which is also widely used in traditional programming development. In contrast to game projects, the exercises in the production phase are carried out in a "cascade" after the pre-production phase has been completed.

The exercises are first separated by job and resource, then into separate groups. The requirement group put much energy into defining gains and front-end exercises and demonstrated late-level implementations and tools. However, it is difficult to reverse the movement using the progressive method. The second development method is an agile method commonly used in game development. These methods are very iterative and document independently. The production phase is divided into a small focus and a focus on the most important highlights. At the beginning of each focus, the entire group fulfils and defines clear goals. At the end of each session, the results are passed on to the customer. These methods support multiple group cycles and elements during the daily meeting.

The most commonly used agile methods in game development are Extraordinary Programming (XP), Rapid Prototyping and Scrum. The linked development process is another well-known approach for SE that focuses on requirements research and turns requirements into programming components for utility companies. The demand survey report includes definitions of game ideas, use cases, and resources. This method includes five assignments for requirements, inspection, design, implementation, and testing. Grouped actions are based on the idea of four main components. Iterative and static, case-based, engineer-based, risk-based.

Although far from the standard, Canudi and Haddad made an important assumption that the GDSE follows the inheritance scheme. Analysts recently agreed that they need to follow a static model in order to link the cascade method to an iterative process. A major concern was a very useless development method that engineers regularly use to create programs in the games industry(Ntokos, 2020). The GDSE appears as an overview of some structures trying to know the nature of the practice being used. In any circumstance, there is no particular answer to this question. Few analysts have examined GDSE practices and looked at headings, such as GDSE process lifecycle times(Morin-Simard, 2016).

The Game development proposes six phases of the GDSE action lifecycle. Pitch, pre-production, main production, alpha, master stage. They proposed a five-step GDSE process lifecycle consisting of prototype, pre-production, production, beta and finally stage separation. The life cycle of the GDSE process includes design, development/remediation, runoff assessment, testing and verification. Another life cycle of the GDSE process proposed by Chandler consists of four phases(Dolan, Whitfield, and Andino, 2018). Pre-production, production and finally post-production. The last GDSE operating life cycle proposed by Ramadan and Widani in 2013 is based on the four recently recorded GDSE operating life cycles. They suggested six phases: launch, pre-production, production, test and beta(VIRUS, 2017).

When creating traditional programming, the development phase usually includes exercises such as designing and implementing the application. The production phase is the phase in which the product is produced and ready for use. In any case, the production phase in the life cycle of a GDSE process includes the development process(Ntokos, 2020). This is a pre-production period of standard programming design procedures and a production period of programming construction. Traditionally, this is the post-production period of the life cycle of the GDSE process. In this sense, the life cycle of the GDSE process is not the same as the normal programming design process, and many scientists have studied the difficulties encountered in this area(Azevedo and Sarinho, 2019). The most evident finding from these studies is that broader programming design techniques must be used to address the difficulties considered in the life cycle of the GDSE process. Most scientists explicitly compare the product formation process to the GDSE process, but researchers who delve into the life cycle and exploration topics of common GDSE processes in this area(Sirparanta, 2018).

This survey provides evidence of these issues and changes from the normal programming design process. In this article, we have broken the GDSE process phase into three phases to understand the basics of pre-production, production, and post-production. Efforts have been made to address further inquiries based on written reviews. The main work of this article is that it will be the first SLR camera to address these research topics in the life cycle of the GDSE process and to introduce topics that need further research by scientists(Rita et al., 2020).

2.6 Pre-production phase

2.6.1 Management

Prior to production, most tests are categorised according to administrative issues at the target site during the life cycle of GDSE processes. The overall control of the game development process integrates design steps into the creation of creative resources. Ramadan and Widiani looked at different game development techniques from a management perspective(Lucas, 2019). Most of the surveys are similar to those that suggested a system for developing games. You can monitor the life cycle of the GDSE process according to your game development rules. Besides, the proximity of agile practices to game development forms is characterised by specific tests. Tschang showed problems in the game development process and differed from traditional programming development exercises(Zan, 2018).

From this perspective, the development and management of peer communication are fundamental. Some surveys show data surveys, detailed surveys on game development practices, and issues of interdisciplinary group participation. The best management exercises in the game development process should consider specific components. For example, keep spending plans and plans and deliver superior returns. Five friendly and quality criteria can be used to rate the quality of the game (comfortable, complete from the inside, change, fun, open). So far, models have been expected to develop exact process steps—game development measures these actions to improve control and superiority(Ntokos, 2020).

2.6.2 Requirements specification

One of the discrepancies between traditional program development processes and the life cycle of the GDSE process is the requirements phase. Many factors need to be considered in the game development process, including emotions, gameplay, emotions, and life factors. In the four surveys, developers talked about design requirements from one perspective to highlight the importance of game programming throughout the development process. They examined emotional factors, ontologies, provocations, reactions and developments(Lucas, 2019). In particular, game developers need to understand these impractical basic requirements and game requirements and take them into account when creating games. The main challenges in defining requirements are: a) support between several founding partners; b) a mix of unnecessary and gaming requirements (such as media and technology); c) non-working needs (like fun) consent, Etc Considering that it is absolutely subject to the intended interest groups. The study also created a set of requirements based on emotional patterns, game criteria, and tactile requirements. In the requirements definition phase, you need to consider the beneficial and useless requirements of game development(Pereira and Lelis, 2019).

2.6.3 Game system description language

Many visual languages are currently used by designers, including UML models, operator-based methods, and soft-system methods. They investigated and announced that it is a decent model for game development on the Android operating system. Learn about the different levels of Super Mario Bros highlights, repeating layer components, and realistic design layers. At that time, they used the players' understanding of research and attentional research presentations to dissolve the relationship between the boundaries of level design for theatre games(Katsaridou and Thibault, 2016). This work proposes a rigorous systems methodology to distinguish game reviews in the development process. The suggested styles can be used in place of the famous photo language to give an overview of the game. Chan and Yuen proposed an ontology information architecture for advanced game development and presentation of original games using the AOSE methodology. The language of the game system images must be in a human-understandable format that supports the mapping and monitoring of player and system practices. In addition, it should be independent of production, correctly explain the overall action of the game and give clear rules to the designer(Ondarza, 2018).

2.6.4 Reusability

Some experts have taken into account the existence of reprogramming and development stages in the development of games, but in order to pick the perfect point of interest, pre-production studies with common features and inconsistencies have been carried out. This category deals with game development programming reuse(Katsaridou and Thibault, 2016). This study looked at ways to reuse game development programming and conducted a study that compared them to programming product offerings. Reusing strategies in game development can reduce costs and time, improve quality and profitability. Examining common features and variations, such as: product presentations, is essential when reusing. Szegletes and Forstner proposed a reusable architecture for developing multipurpose games. The proposed system architecture consisted of loosely coupled components to improve adaptability. They tested the system by creating a how-to game. The requirements of the new game should match the reusable components of the newly developed game(Azevedo and Sarinho, 2019).

2.6.5 Game design document

The game design document is an important milestone before production. It consists of basic components, their interrelationships, orientations and a clear description of general, detailed development terms. This study tends to doubt the unpredictability of the design in the original game and suggests a design scheme(Sklyarova and Volkova, 2019). In addition, the study emphasised the importance of game design reports in game development and examined several game design records that were accessible in writing.

Similarly, they compare the results and specifications to typical programming assumptions, and the helpless game design files can lead to poor quality problems, mods, and budget issues during the production and post-production phases(Li and Cheng, 2020). This study raised the issue of ensuring the standards of the game and left out the options involved. They proposed a way to address the problem of choosing a medium based on the nervous system method and using genetic calculations to improve the design. Finally, Ibrahim and Jaafar worked on the game material model in-game design files. From now on, GDD is faced with morphology and imperfection. Proper development of the GDD is important to solve this problem. Comprehensive GDD brings excellent game quality(Cao et al., 2020a).

2.7 Game prototyping

Prototyping the game before production allows engineers to explain the basic mechanics of the final game. Pre-production prototypes are considered important as they serve to convey the mechanics and gameplay of the game and to evaluate the player's understanding of the game(Sklyarova and Volkova, 2019). Reino and Cubel proposed a prototype that will develop games based on a model-based approach. The programmatic change creates the product model code in C++. De Study also proposed a prototype for network-based games. Engineers can move towards the underlying grid to highlight niche objects rather than starting without preparation(Azevedo and Sarinho, 2019). They used this method to develop a great multiplayer online game. A proposed study that separately examines rapid game prototyping, historical plans, intuitive 3D computer games, and game development structures. Prototyping can also help identify missing benefits. Engineers can then make quick design changes without significant expansion. You can prototype your game using fast or model-based prototyping techniques(Rita et al., 2020).

2.7.1 Design tools

Game design tools allow game developers to visualise game effects and events in detail without the need for advanced programming capabilities. Cho and Lee suggested an event device for fast game development and confirmed that no programming skills were required.

These tools allow you to reuse existing components and shorten the absolute season for creating games(Cao et al., 2020b).

2.7.2 Risk management

In the field of game development, risk management agents are not generally commented on by experts. Risk management is important from a project management perspective. It is also essential to know the risk factors in the game development process(Bosman, 2020). In game development, the project manager is the creator of the game and must combine the perspectives of management, professionals, and hobbies to create an effective game. You recognise two risk factors during the development process: disappointing development systems and the lack of soft workers. Important risk factors in game development are development measures, ease or degree of innovation, plans, plans, Etc. However, the game engineers have significantly reduced the need for a formal investigation of risk factors(VIRUS, 2017).

2.8 Computer game structure

Our method of creating computer games is to break the game down into about six basic components and assign each of those parts to alternate students(Zan, 2018). The student's job is to implement these components during the third semester and then incorporate them into the rest of the game. In this segment, we present a summary of the degradation used in our group. (Note that this decomposition, used in business game design and development, clearly highlights the reason for the separation.) Before we proceed, first consider the programming problems that apply. Before you can start a game, you need to set a language that describes the game(Sirparanta, 2018).

This study allows you to introduce some of the most useful parts of the language. However, some issues are also expected to be fixed, such as: manual memory management. We use source control software to process code for large projects like computer games. The hardware (PerForce Etc) is perfect, but the tuition is limited to programs the university can buy or get for free. So we use CVS to take care of storing category codes. This storage is entered as a single exchange point for tokens and monitors synchronisation as it appears during development(Ananda and Chusyairi, 2019).

Similarly, the control of the source code provides a documented tool designed to restore the code to a stable state if an actual failure occurs during the encryption process(Morin-Simard, 2016). The class also applies strict criteria for adding symbols to the safe. You should at least provide your last check-in code. "Breaking the Industry" is a real crime and is discouraged due to the pressures involved (culprits are emailed to the rest of the class). In the ideal world, the code has to go through some cycles to test its authenticity before it is registered. In practice, this test method is challenging to implement. Many factors affect the game and its condition(Sirparanta, 2018).

Two things that are difficult to control, such as the exact timing of information and the transmission of packets through the system. As a result, many of our tests are performed with autonomous adaptation to the disabled system management, which is generally insufficient to ensure that the software is in its final operational state. In any case, there are many trivial flaws in this system that regularly cause incredible headache to other members of the development group(Ondarza, 2018).

2.8.1 Modelling

As we customise the various game components for students, we try to provide them with core components that have the experience and inspiration to try the game out. Pretending is probably the most important part of the game(Bosman, 2020). The decision to display the system determines the type of math displayed in the game, the usefulness of that computation during the game, and how artisans control and generate the operations in the game. In order for the game to be fruitful, you need to complete the demo system in the development steps and configure it right from the bat. A burden on this methodology is that developers need to build a transformation model in order to create this verifiable image of the polygon model(Azevedo and Sarinho, 2019). In addition, the game cannot talk to accounts that are smaller than the square of the net, which limits the options for level designers.

Additionally, Level Designer's activities are confused by the fact that the calculations displayed in the game depending on how the imported polygons are arranged in the grid(Ntokos, 2020). This method offers the highest degree of adaptability because complex calculations can be placed away from the underlying network. In addition, polygons and vertices can be explicitly marked with data such as orientation and surface values. Famous subtitle packs like Maya are also ready to sell these polygon models in a standard layout and can be introduced into the game. The downside, however, is that you have given up on the productive / destructive math that makes understandable structures attractive. Another area of adaptation to current practice is the ability to preserve the current mathematical design of a business game for which the data set is known(Pereira and Lelis, 2019).

In our last game, Time Bomber, for example, we decided to use the Quake III level rating, which gives us instant access to test levels. This decision eases the bottleneck in the coding process because the student in charge of the presentation is likely to make changes to the layer created by the designer(Ananda and Chusyairi, 2019). Additionally, students assigned to companies like Delivery and Impact Discovery at these levels can pass their tests and resolve their own code before completing conversion actions(Rita et al., 2020). However, if you use the existing layer placement, you will see design options and highlights in this record. This may or may not be desirable.

2.8.2 Animation

In our game, animations control how various player activities develop. For instance, when a player shoots a weapon, the animation system changes the player's visual appearance to indicate a predefined movement. This component may not be required to play, but it is considered a prerequisite as it will significantly reduce the playability of the game if the animation does not display a clear warning. Students recently implemented two different options for animating characters. A skeleton animation and a master edge animation. Each approach has its own characteristics and limitations, which we will explain in more detail(Ntokos¸ 2020). At this point, the game switches between different animations using a status table that is listed according to the events of the game and the animation currently being created. Skeleton animation is an animated procedure that uses schematic structure to talk to a character's skeleton, with each edge of the skeleton talking to a living org chart.

The neighbourhood eases certain "bone" edges, so the character is activated by speaking to the model's head. At this point, the painter distorts the structure of the skeleton, deforms its head and gives life to the character(Wilson¸ et al., 2018). This system creates a small animation log because the diagram structure must be removed from the animations, and Illustrator provides a reorganised interface for creating those animations. This method is also suitable for dynamic animation of games. For example, you can move your character backwards in different ways by capturing the character in different areas without using preprogrammed animations(Dolan, et al., 2018). Developing the skeleton for modelling and assigning the head to the bone is an easy task. Our answer was to create an animation using free or affordable tools. One such device we use is Poser which has an extensive database of previous animations. However, these animation tools do not send the animation to a base placement that contains all of the base data for structural animation(Cao¸et al., 2020).

For example, these tools typically include considerations for allocating loads to the head due to limited orthopaedic effects but do not pass distributed loads on to software developers. Because of this lack of data, developers have to interpret the missing data, so animators typically do not show exactly how animators wanted to add complaints. Leading animations, however, do not use skeletons in their animation(Wilson¸ et al., 2018). The main data is the state of the character's head with a clear focus over time. To reproduce the animation, the software mixes the headbox between two different levels. The good part of this methodology is that the data is clear, and the animation program can always send the model in clear cases. In addition, this system frees the artist from specific structural requirements and allows greater customisation of the animation performed. In either case, large frame animations are unsuitable for dynamic animation performance and produce large animation data sets because document size is a multilayered nature of the model(Radchenko¸ 2020).

Chapter Three

Methodology

This study collects data according to epistemological criteria. Examination with other test methods makes it clear why this was the most plausible method for this theory.

3.1 Agile Method

The agile development allows for exceptional upgrade notifications for product assignments. Since computer games are developed using information technology, similar standards that lead to product development currently also regulate the development of computer games(Li and Cheng, 2020). As a result, agile development plays an important role in the production of computer games. A rapid evolution was reported in the 1990s that gave rise to the mainstream paradigm of program development(Li and Cheng, 2020). Previously it was defined as the moderate pace of other traditional methods of storing many records at the end of development. Valued preferences for agile development include proactive adaptation to changes, fast component transfer and excellent meetings with everyone involved in product development through continuous agile collaboration.

Another popular place is code development and does not require extensive documentation. Besides, agile development is viewed as optimal for use in small groups of 10 people or less(Radchenko, 2020). This item can be a little slow or annoying, depending on the size of the individual group. This method is easy for outside engineers, but it can be difficult for AAA designers. One of the undeniable weaknesses of Agile is its over-reliance on a particular workforce. This places a heavy burden and obligation on some members of the group and can slow the pace of development. Another problem is the relationship between agile practices and project management. Agile does not advocate a specific approach to project management. Nor does it provide a framework for how the results will be achieved in the project. Instead, Agile relies heavily on contextual approaches to approach projects. For more accurate data on the expectations and goals of agile development, announcements have been made to move agile position points and engines(VIRUS, 2017).

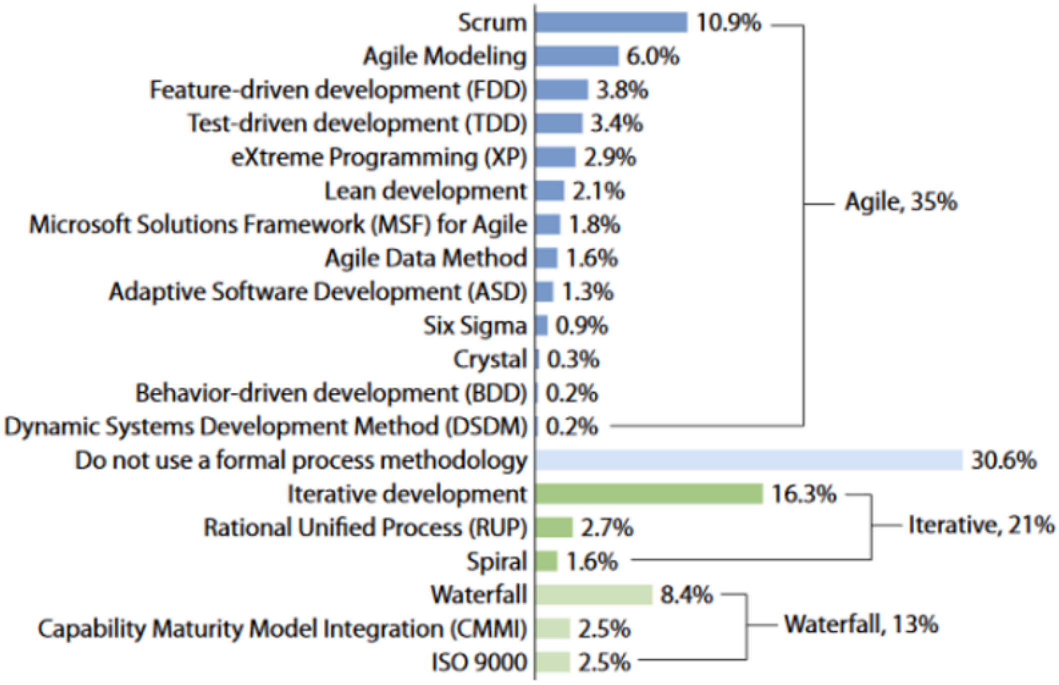


Figure 3.1: Graph of Methodologies in Computer Game

Computer game development is generally a complex process based on standards of agile development methods for programming. In these methods, courses occupy a predominant position and are seen as an important element in achieving high standards and as an important component at various stages of development(Ntokos, 2020). These development phases are the foundation (advanced ideas, suggestions, ideas, game design files, models), structure (design, level creation, programming, staff production, sound production), as shown in Figure, finer details and improvements (can be in the alpha stage before filtering and are in the code freeze stage)(Morin-Simard, 2016).

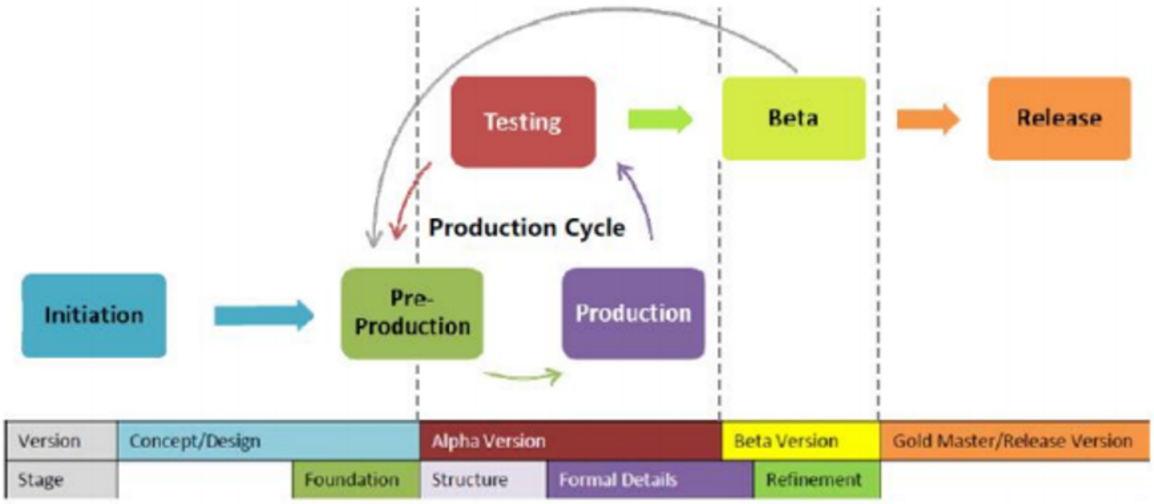


Figure 3.2: Stages of Agile Development Method

3.1.1 Ontology

According to Aristotle, ontology implies as much as it currently exists or at the end of reality. Ontologies can find out about correct and general ontologies. The former realises things in an explicit and individual way, whilst the latter realises the lack of protest and yet the nature of everything. Based on these definitions, it is clear why ontology is not the right way to convey this assumption(Bosman, 2020). Ontologies and agile methodologies, which are generally focused, are unique ideas and their relevance in the computer game industry is openly overlooked. Furthermore, in the course of the formal ontology, the formal ontology is not based on precise facts and statements that are carefully examined in order to make, estimate and evaluate further decisions(Ntokos, 2020).

3.1.2 Axiology

Axiology is a search for value. These characteristics can broadly be explained as the link between the article and the enthusiasm of the topic for it. These properties are assessed and analysed through an objective focus set out in axiom standards. Objectivity is based on perception and perception of instinct. In particular, scientists who identify as proponents of axioms strive for the agency's objectivity in their research. Given what is said in the archived idea of the agile cycle, it turns out that axiology is not the right way to express this suggestion. Of course, the originator of this hypothesis has his own preferences for the subject(Sirparanta, 2018).

In particular, the founder defies Agile's claim, his character starts with the topic and ends with a series of test questions. In either case, the Creator tries to find reality by combining different ratings from different experts based on recorded facts, not based on their personal encounters or other individual ratings. The results of this proposal are not intended to emerge from the truthfulness of the facts, but rather take into account the subjectivity of the originator or each member in order to compromise the legitimacy and scope of the data collected. We will enter the further room to examine this current proposal from an individual perspective that is done(Ondarza, 2018).

3.1.3 Epistemology

Epistemology is a dynamic step in finding the right activity to understand a person's personality, moral compass, encounter, insight, and information. Epistemology follows scientific standards. It will never end, but it will keep moving forward and pushing its limits. In other words, epistemology is part of the thought of evaluating conflicting views about the quality, nature, definition, principles, sources, and elements of information(Azevedo and Sarinho, 2019). In every way, epistemology is legitimate and accurate. It is an inspection method that is used as a virtual backbone in gathering various information and sharing information. Tests show that the ethical implications of this methodology show that advertising and management research is regularly linked to epistemology. This includes a valid rationale for obtaining that particular methodology for this proposal. In particular, epistemology examines various structures and phenomena of information.

The epistemological criteria recognise the two types of information or data used in this theory as "authoritarian" and "empirical information"(Pereira and Lelis, 2019). The first contains data from people surveyed about the respondent's ability, and the second contains additional data with targeted evidence and a range of facts. In summary, "logical" information is used in the investigation of data and, for the sake of reason, the results of the survey are collected for useful purposes(Ntokos, 2020).

3.2 Data collection

3.2.1 Quantitative vs qualitative research

Quantitative methods mainly use numerical data from statistical models, studies, research, Etc. To support scientific models and conclusions. This method usually has low syllable levels that appeal to everyone(Li and Cheng, 2020). The purpose of this methodology is to achieve objective and fair results. Often used in neuroscience, merchandising, and finance research. Quantitative research relies heavily on scientific models to produce results(Bosman, 2020). This is not a great approach to this proposal as it is based on a general test of the data which will decode that data based on the data they provide.

Qualitative surveys are used for the exploratory investigation of suggestion points. Some of the knowledge and data is gathered based on the individual meetings that are mailed by the creators of this proposal and that are later used to form the final and answer the survey questions. Qualitative research seeks to obtain factual and channel data about their survey members(Sirparanta, 2018). In general, qualitative research reveals key issues emerging from the data and the data collected during the interview. Qualitative methods depend on the relationship between the data acquisition and subsequent hypotheses. Epistemological methods, which can then be called leaders, try to decipher the world through their members (in the theoretical case members, are called speakers)(Pereira and Lelis, 2019).

3.2.2 Secondary data

Extensive communication data such as television and film projects, as well as proposals, magazines and editorial articles can be used for qualitative research. A guarantee of the source's credibility can be found, taking into account the niche and size of the journal to which this theory relates. These sources were considered important to describe the subject of the proposal and to increase knowledge of exploration research from different perspectives(Morin-Simard, 2016).

3.3 Planning and Execution

Game development was generally project-based. As shown in the beginning, the game is created in four phases. These phases generally include the idea phase, the pre-production phase, the production phase, and the post-production phase. A secondary sales phase can also be included. Many modern games, especially web-based multiplayer games and multiplayer games with constantly updated content, require regular maintenance after delivery(Sirparanta, 2018). The traditional view of this process in many industry manuals is somewhat continuous. Game ideas are generated in the ideas phase. Game title activity may come from the distributor(Orji, et al., 2018). In this case, it would be a regular selection of measures.

The dealer has a creative kit to get ideas for ideas or can advise an outside game designer. With some new titles, especially independently published titles, this step is very different, and the degree of freedom in the whole development process is much clearer. In the sequence model, the game design is optimised in pre-production, and the game design is registered. GDD is made up of game designers and provides detailed explanations of gameplay depending on the type of game, character, Etc. (Bosman, 2020). There is no layout for game design files as it is clear that every game is unique. The study warns against using GDD as a record of production requirements. The GDD is intended to continue to exceed the game's potential and the requirements are listed in a different register specifically included for production purposes(Pereira and Lelis, 2019). Demonstrations and operational prototypes are another alternatives for testing ideas. For example, Keith suggests starting production as soon as the vulnerability to mechanisms and technologies decreases. With this agile way of dealing with game development, it is ideal and necessary to continue pre-production until you find the game interesting(Cao¸et al., 2020).

This is not possible without a prototype that you can actually play, as you can see in Massive Entertainment for example. Various partners such as distributors have placed their own orders as a track record and downtime in this promotion. In a survey to document game development, Winget and Sampson found that operational prototypes are essential today. In addition, their meeting showed that "assembly is important" to create "fun" and "elements of the game" (Li and Cheng, 2020). This seems to compare well with agile development. Pre-production was usually a scheduled period for game development. Special design files are provided in pre-production and contain the architecture deployments and unique parts of the game. Production begins in an ideal world where pre-production ideas are properly embodied. In a more traditional methodology, where pre-production is in the planning phase, production begins with the game engine making, and the main phase of production may not have an actual game to display(Down¸2016).

The dealer determines the successes and the progress is regularly checked. With Keith's agile game development model, the core mechanics are ideally excluded from pre-production, making production an ideal opportunity to develop gameplay and incorporate content(Geraci¸ et al., 2016). A more flexible methodology results in less separation between game development periods. This can also include pre-production and production exercises for some time. Post-production of sequence models is an ideal opportunity for final testing, troubleshooting, and cleanup. In the Lean model, the tests are consistent and are carried out before and during production. For example, some tests on specific devices should be done by the end of the project(Rita, et al., 2020).

Usability tests are the most widely used test method. It is important to check that the game mechanics are working and that the player is loading the game. Running tests on the client can also help ensure that the game has the correct problem. In any state, it is important to preserve the project from anomalous changes at this stage to avoid implementing schedules and spending plans in both cascade and agility models. Game development was more open to slow changes than general program development(Pereira and Lelis, 2019).

Chapter Four

Result

This section describes the tracking of the found papers and the results of the survey. Articles are initially assessed for context analysis and research. A small picture of the survey is provided, and the paper describing the attempt is recorded(Sirparanta, 2018). In this phase, exploratory problems are analysed. The Idea Network helped us focus our exam on the ideas in question. For example, some ideas regarding the quality of the game itself and the quality of the code should be separated. On the one hand, the results change in relation to them.

The article contained general overview studies and context research. Both numbers have been fundamentally revised. Seven public polls were published in 11 articles and eight contextual polls in 12 articles. Similar reporting material is occasionally used in several articles, sometimes focusing on different topics. The survey was an online check for death tests (4). The meeting was combined with an online survey in two surveys(Bosman, 2020). This includes an experience report that is a mixture of two encounters and scenarios (1). The standalone games category includes specific methods, initial shooting, online poker, slot machines, and virtual worlds. The game is designed for a versatile and portable computer and console(Zan, 2018).

4.1 Use of agile methods

This section seeks answers to basic exam questions. How widespread are agile game development techniques? Discussions about quantitative studies provide some clues. Agile techniques seem to be used on a regular basis, and Scrum is the most popular technique referred to in testing. In Austria, a study with 13 participants used flexible or lean methods in 77% of the organisations. The review was carried out in 2009(Katsaridou and Thibault, 2016). In a 2011 Finnish survey, practically all respondents used agile methods at certain points in time in certain parts of the development process. Scrum is well known for pre-production, production, and post-production, and more than half of the studios have used it on these stages(Bosman, 2020). Thus the Austrian and Finnish overviews really provide comparable results. The 2012 Finnish conference survey was complemented by a review in 2016, which included game engineers from around the world.

The outcomes of this study differ slightly from the overview for Finnish studios only. Only 26% use Scrum as a development method, and 13% use other flexible methods. However, almost all organisations use iterative methods. The dangerously high rate of 61% felt that there was no systematic method of development. This is in contrast to an Austrian study in which only 23% did not use a specific procedure(Rita et al., 2020). Differences in public ratings can explain this difference. The number of respondents to the public offer of Kaslinen was large and not concentrated in a single country(Katsaridou and Thibault, 2016). In another article, most of the Finnish organisations have met at length with agile techniques. However, this is a different route as it depends on planning. Three of the seven organisations used Scrum, one the agile process and two the so-called agile development approach. Even those who have disclosed the use of Scrum and other agile techniques operate in what is called a "pipeline" process(Sirparanta, 2018).

As part of research into whether the gaming industry can use systems and programming to design ISO / IEC 29110 for some materials, the Organisation we spoke with said the patterns are not repeatable enough Game development is often an iterative process in a sense, and even seems to be a normal agile process. In the Finnish overview, there were some obvious inconsistencies in adopting the elegant style. The Finnish review concludes that, unlike the Finnish programming industry, agile techniques are not well known in the games industry(Orji et al., 2018). However, most organisations use agile practices. However, discussions and analyses showed that flexibility was used more in game development than in other software developments. The author reminds us that the Organisation can report the use of agile methodologies (like Scrum) but that it has a pipeline aimed at planning the game's later development. It is, therefore, uncertain whether the subjective details of the development measures are robust.

Of course, many organisations are enthusiastic about their rapid development. In a post-mortem case study, 11 of the 20 projects were reproducible, six were consecutive, and 2 were each half(Bosman, 2020). Although the four repeating actions were not recognised as agile, both hybrids were just as agile as the seven repeating motifs. Therefore, 9 out of 20 steps are considered accelerated. Obviously, not all post-mortem cases have equivalent data scales, so there may be problems with decoding after processing. The exciting thing about the half-action model is that the initial production was done in an agile manner, but the production was changed to a series action when the item was nearing completion(Wilson et al., 2018). A 20-day post-mortem survey did not consider the agile method itself but looked for evidence of the agile method.

4.2 Degree of agility

A Finnish study found that most organisations were flexible and even officially accepted to use agile techniques. This does not mean that all institutions were pacing at the same rate. They found that the overall course of the process was inconsistent, with some organisations starting with a more traditional conventional pipeline approach whilst others highlighted with consistent and cyclical responses(Li and Cheng, 2020). An online analysis by the

Finnish game studios showed that agile technology was used to varying degrees depending on the development period(Morin-Simard, 2016). For example, Scrum exercises are mainly used for prototyping and production. The pre-production and post-production execution and overflows are known, so not all methods are applied in a similar way. Scrum is also used daily in parallel by organisations in all development phases. Burner is mainly used in manufacturing when used imaginatively(Cao et al., 2020a).

Generic groups, indicative workspaces, and ongoing participation are most commonly used in built-in XP exercises(Morin-Simard, 2016). The comparison is part of the form for the post-mortem examination procedure. Halftone distortion is used in the production phase, but at some point, it becomes more cascaded. The figure shows an example of the agile evolutionary steps used in Massive Entertainment(Morin-Simard, 2016). In this figure, the actions are constantly changing, there are focal points to consider, and of course, changes are to be expected. Hence, the degree of flexibility appears to vary between organisations and in relation to development processes within the Organisation (Down¸2016). Contextual research also made it clear that there were disagreements at the authority level. Some studios only have one elegant group, whilst the rest of the larger groups vary depending on the arrangement. In other cases, the entire association effectively uses agile methods. There is little data on agile practices that defy the imaginary situation. All use cycles or courses as each one repeats itself(VIRUS, 2017).

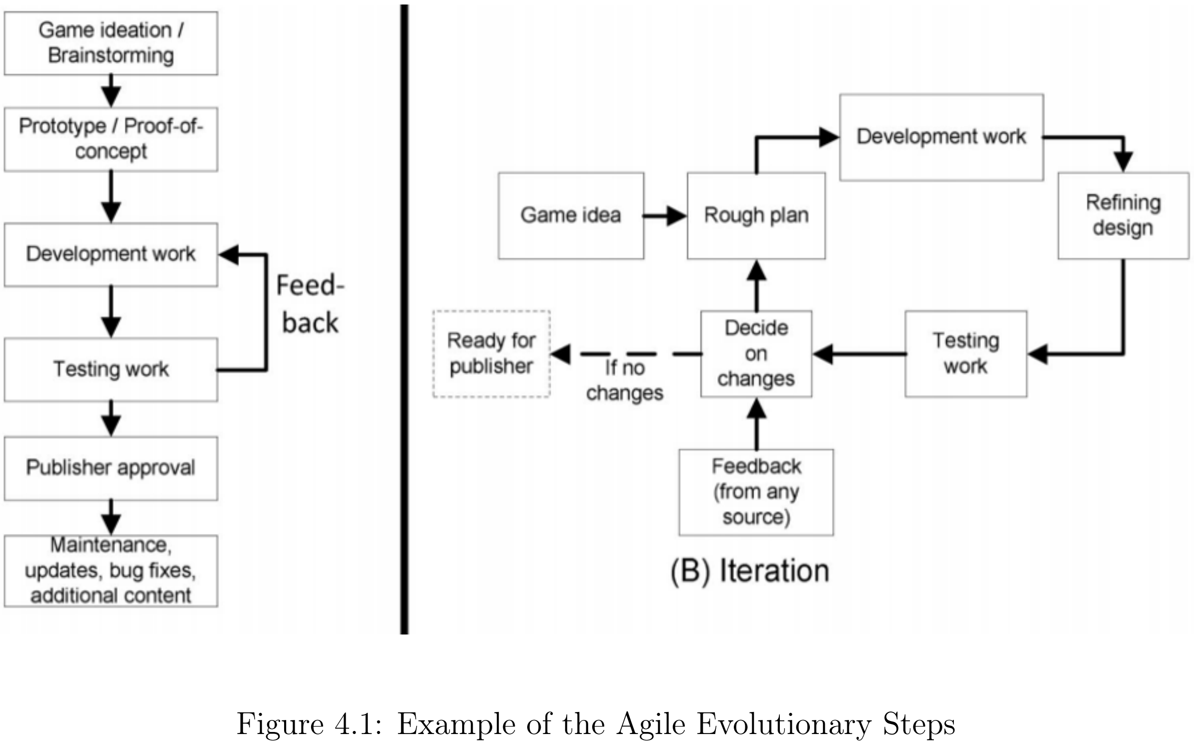


Figure 4.1: Example of the Agile Evolutionary Steps

In either case, a daily scrum or backup meeting has five states, one of which is from a group that is not giving the game direct shots. Multiple sessions will also be held to plan, review and reflect the context of the case study organisation. However, their names can vary and are appropriately organised in unexpected ways. Overuse was reported in four contextual surveys. In further investigations, a comparison will be carried out under the name "feature plan" (Li and Cheng, 2020). Flare diagrams are used by the corresponding four planned organisations. The current agreement is shown in two status surveys. Two quantitative studies that are being considered for continued participation produce conflicting results. A post-mortem study by Petrillo and Pimenta found a stable combination that was only shown in 2 out of 20. However, a Finnish review found that continuous adjustment is one of the most common XP exercises(Dolan, Whitfield, and Andino, 2018).

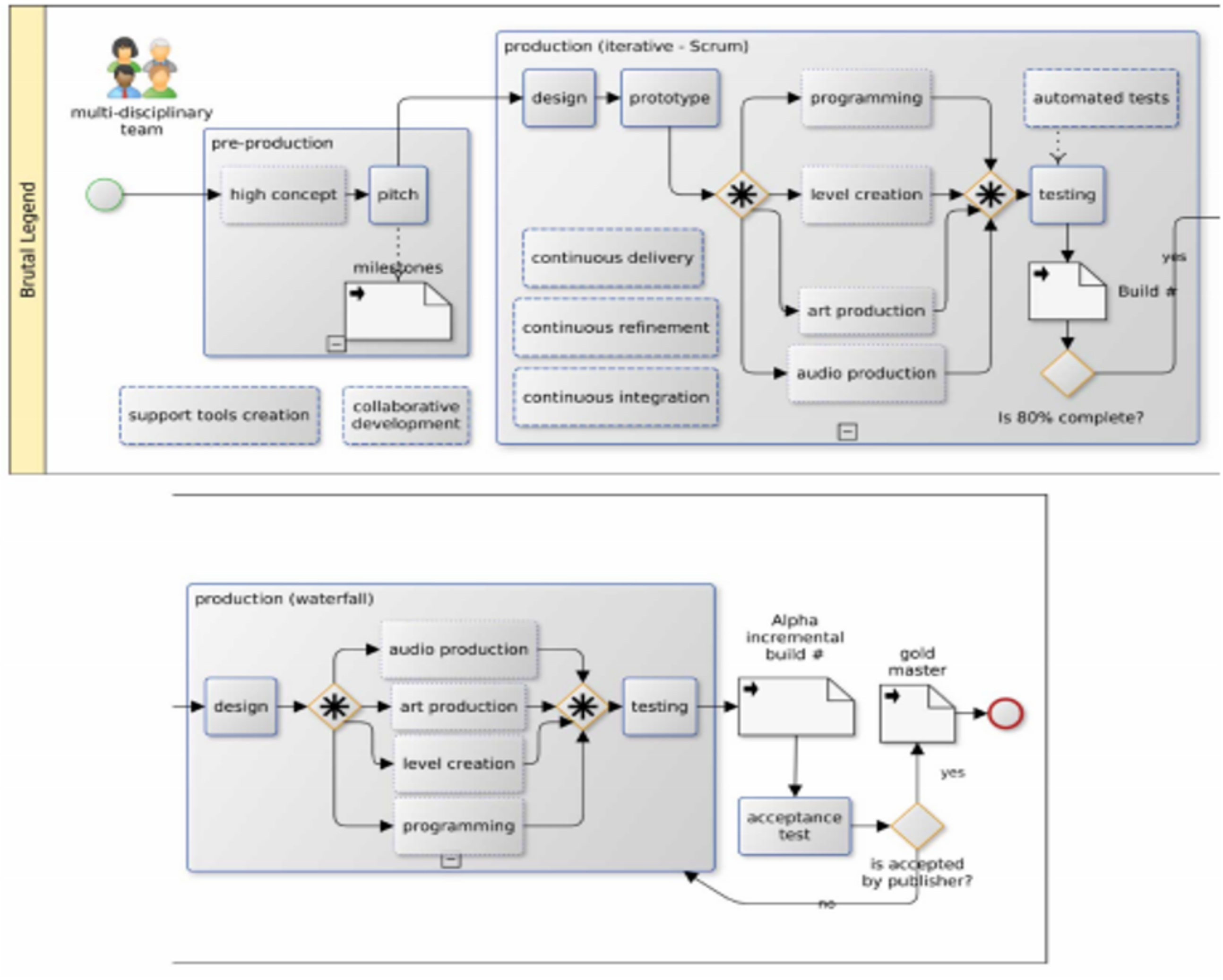


Figure 4.2: Feature Plan(Li and Cheng, 2020)

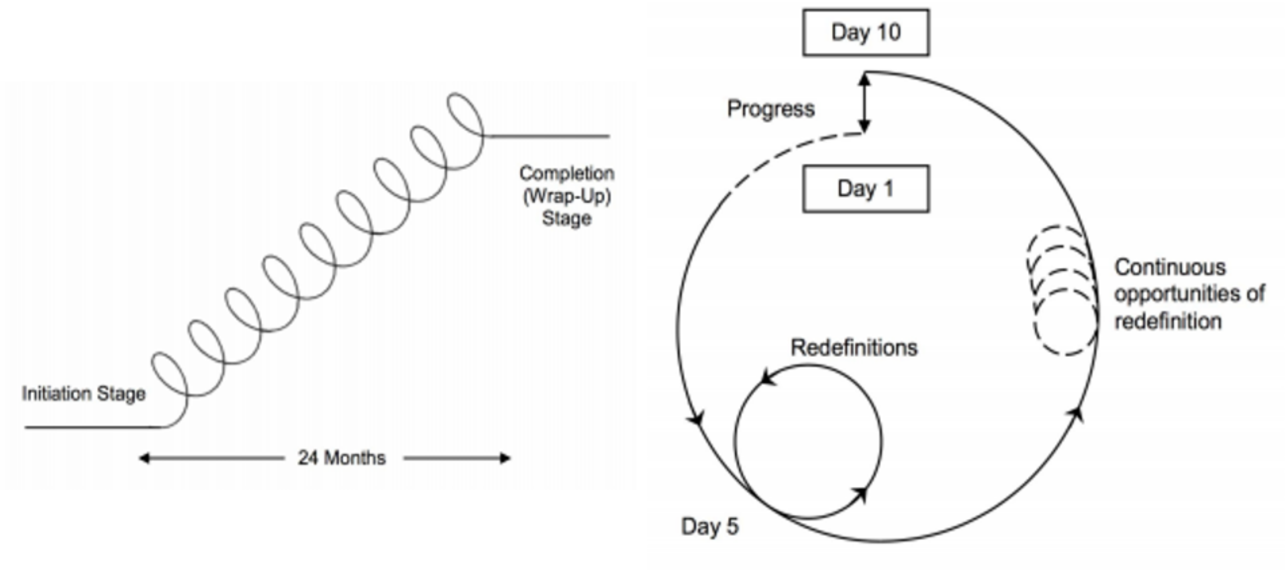

Initiation Stage
Completion (Wrap-Up) Stage
24 Months
Day 10
Progress
Day 1
Continuous opportunities of redefinition
Redefinitions
Day 5

Figure 4.3: Example of A Post-Mortem Study(Dolan, Whitfield, and Andino, 2018)

4.3 Prototypes, playtesting and feedback

Despite the fact that prototyping is not inherently a core agile practice, early prototyping and demos are viewed as a crucial part of the game development process, especially in the early stages. Prototypes are used to test game ideas before production before the focus is on production. The prototype also provides an attractive demonstration of potential distributors and sources of funding for components(Li and Cheng, 2020). One engineer said in a study at a conference, "Why to spend time making difficult decisions when you can build and test a model in two hours?" The developers argue that ease of use, gameplay, and "fun elements" determine the decisions that are made through prototype testing. Prototyping is an important part of ensuring these impractical theoretical requirements. The study considered prototyping, the proper pre-building of various parts of the project, and agile practices as mitigating factors(Zan, 2018).

The test banknote switching is also a central part of the work process at Singapore Studios. The feedback from gameplay tests can have an impact on significant changes to the element and in any case on the redevelopment of the idea phase. This type of change does not mean that the entire element has returned to the beginning(Wilson et al., 2018). For example, in a risk management meeting with one of the respondents, the level design could now be in production whilst some game mechanics could be in pre-production for more fun(Wilson et al., 2018). The demonstration at the end of the race is also used by associations and partners to keep their projects up to date. A playable variant of the game was also delivered at the end of every major entertainment tournament. The Organisation's feedback on these tasks at the beginning of the development steps helped improve the quality of the game(Orji et al., 2018).

Given the importance of these comments, Massive Entertainment decided to explore the player also to have an engineer's perspective on the engineer. For comparison, a playable copy of the game was distributed to all staff at the London studio for each outcome. In an organisation of 200 delegates, this generated many comments. In this case, the feedback came from all employees, and not all were protagonists. However, unlike the ongoing action titles from Massive Entertainment, the Organisation has created a versatile game that is more open to a wider public(Ondarza, 2018). The gameplay or testing of clients inspires new features, as an overview of situations in Singapore's studio shows, in which mistakes have become interesting elements(Ntokos, 2020). This was also found in a survey of Finnish conferences. There the Organisation showed a change in the element after inadvertently discovering a new interest profile through customer testing. The feedback gathered from the tests was legally conducted in another form of the game and not systematically recorded. This type of feedback loop is not used by all organisations that have declared it agile(Azevedo and Sarinho, 2019).

4.4 Agile Adoption Factors

Useful prototypes help send requests to the development group. Iterative methods are useful when the association needs information on another topic(Pereira and Lelis, 2019). The association can take on the task of developing a spirit of synergy and providing the necessary material tools. The XP method lends itself well to bottom-up development because of its tendency to change requirements and production stability. The model of the winding process is suitable for mega-projects(Sirparanta, 2018). For example, the existence of a suitable group of working people, such as those who have usage or game resources. Pay attention to the highlights. Closes free control over the entire project. Quality and creep protection is very important for certification(Ananda and Chusyairi, 2019).

When requirements are difficult to explain in advance and cannot be predicted, when there is a need first to confirm the introduction of an element and its beliefs; There is a desire to improve the quality and profitability of game development. Indeed, Scrum can pose problems of revision; for example, Kanban can alleviate these problems in production. The Scrum project management process requires a flexible schedule between the designer and the various implementation partners(Ananda and Chusyairi, 2019). The president must be ready to move at the same rate as or faster than the group. To support his efforts to move to Scrum, the Organisation called together an external expert and led the Scrum. The Organisation should not have left the specialist before the information was transferred, resulting in several development steps. The unwillingness of contractors and the lack of workable tools to accelerate the rapid development process caused problems with the introduction of Scrum(Orji et al., 2018).

4.5 Use of independent learning

At the beginning of the course, students emphasised their ability to build games. After understanding nature and finding the data they need, they will quickly understand how to create their own game. Students discover most of the data themselves and change it in their projects. They found some web resources and gave them to a colleague(Li and Cheng, 2020). The way they sifted through the data shows how they learned. For example, at the beginning of the course, the students showed that I was feeling confused. Anyway, after a whilst when I decided what to do, what data to need and where to look, it all became clear(Sirparanta, 2018). I received many websites on the internet. I took source code from these goals and adapted it to my project. When I found a way to learn without learning from others, I found my game to be easy. It should be emphasised that for me, the internet is a special access point to learn without others. Everything I asked for was there. I tracked and processed the data. Different students came up with great ideas for the game.

The teacher did not comment on most of these ideas during the class. For example, a game of "Steel Pong" created by someone else has two poles that a customer controls, two poles attached to the edge of the table (jump when the steel circle hits the edge row of the table), and one. It consists of a bar that blocks when hit by a steel plate suitable for two-way movement with the player(Pereira and Lelis, 2019). The variant that this game created had some issues during development. For example, he had to find his own strategy. For example, frustum separates the strategies and implements them in the game(Cao et al., 2020a).

During the interview, the teacher did not provide any specific data on the SE model. In either case, the teacher responds to the game design based on some of these factors, e.g., the needs identification process, the documentation of the requirements survey, the methodology used in the product development process and the tools used in the product development process, evaluated the product development process. The session results have been improved. Thirteen students who have not yet taken the SE course scored high in the game design report. The results show that these students chose relevant data to produce high-quality documents(Ananda and Chusyairi, 2019).

Chapter Five

Implementation

The video game industry is very young. It started a little and over the past 30 years has grown into one of the most profitable companies in the world. Without a doubt, the history of classic video games began in 1972 with Ralph H. Bear, best known as the "father of computer games". It belongs to the realm of transformation and crafting, and its evolution and new items are traits that everyone, even non-gamers, will notice. Computer games mean intelligence and diving. There are four areas that contribute to this goal(Morin-Simard, 2016). Respect for sensation and joy, work value and depth of the game structure. To advance in these areas, computer game developers must oversee a large group of people and assets. There is no doubt that the production of computer games has to cost 1 million euros in a standard production cycle of 2-3 years(Dolan, Whitfield, and Andino, 2018). The building is not an easy process. Due to their innate creative nature and the customer's need for something to enliven them, computer games are constantly being re-evaluated, and fundamental decisions are made to keep the customer happy.

These options typically require a development rollback(Azevedo and Sarinho, 2019). It is not impossible for the substances so far invented to be rejected by others of greater value. There will also be events to discover other creative ideas that designers think are better than the basic ideas or to find ways to implement developments that are more effective than the methods previously used. The most popular engineering and management device that allows for customisation during development is an agile method that encourages the use of a large focus in every development phase(Geraci, Recine, and Fox, 2016). Agile is an exemplary method of manufacturing products that require multiple levels of collaboration between self-organising groups (made up of people with different roles and responsibilities) and their intended customers Basically; it is a method that recognises and promotes adaptability, ongoing planning and review in order to achieve a continuous realisation process and achieve first-class results. In any case, agile focus comes with some important consequences during manufacturing. An important consequence of resuming development is the risk of losing significant development results. This means that companies are going over budget and facing financial challenges(Wilson et al., 2018).

To pay for this, the Organisation removes materials to get the item ready on time, investigates the relationship with the seller, uses more assets than originally budgeted, or creates a crisis. To explain the bottom line, the Organisation makes representatives work longer than usual. Wonder of Crisis has received so many visits to the industry that it has been standardised as something special overall(Geraci, Recine, and Fox, 2016). It is a small vote for any partner in the PC game industry, and controversy is difficult. This assumption confirms the idea of whether the end product components generated by consequent re-testing can be achieved during early development(Ondarza, 2018).

5.1 Design

Covid-19 has been rapidly and seriously affecting various countries around the world lately. It has been observed to spread through physical contact and granules in the airways. Later some countries issued closing guidelines to avoid open meetings and to encourage arching(Sirparanta, 2018). To maintain physical separation, you need to wear a cover over your nose and mouth, regularly use disinfectant to keep your palms clean, and see a specialist if you observe the side effects of piracy and fever. By reducing the contract risk, viruses(Ntokos, 2020). It is necessary to instruct the public to follow the above measures. Virus Survivor is a hands-on game that will help you understand the importance of scarves and disinfectants to protect yourself from COVID-19. The game is based on a basic top-down, pixel-based design where individuals explore the place using health and safety measures(Pereira and Lelis, 2019). The game starts from the access screen.

When the game begins, players get information from the main menu to tell the basic idea like how to play the game. A screen will then appear with instructions on how to quit the game. The player's goal is to collect a distributed number of foods and superpowers exploring the place(Radchenko, 2020). Each food and superpower gives an ability to the character to trounce monsters and other enemies. Players have to use it on purpose in order to win. The character comes with a face mask at the beginning of the game. The game ends when the character reaches the checkpoint and enters in next level. Also, this game has a hidden room that the character can collect more coins after defeating the enemy.

5.2 Implementation

Before implementation, it is necessary to know what functional requirements should be required- java framework(Eclipse/NetBeans) and memory compatibility to play the game. Implementation has done using Java AWT GUI library(Fan, 1996).

- **Event Handling:** An input device plays an important role to move the character from left to right, by clicking buttons from mouse or keyboard. Java AWT handles these two input devices. It triggers an event by clicking buttons(Fan, 1996).

  1) Keyboard:

i) **java.awt.event.KeyEvent** is an AWT keyboard event which is used to give character to move by pressing buttons. Some features are as follows,

– id designates typed, pressed, or released methods.

– keyChar includes the ASCII code of the key pressed method.

– keyCode contains a constant recognising the key pressed method.

ii) **java.awt.event.KeyListener** is used to get notified if there is a change in key state. There are three abstract class methods such as,

– keyPressed, keyReleased, keyTyped

2) Mouse:

i) **java.awt.event.MouseEvent** is an AWT mouse event which is used for mouse movements. Some specialities are as follows,

– id designates clicked, pressed, released, moved, entered, exited, or dragged methods.

– x,y hold position of the cursor of mouse

ii) **java.awt.event.MouseListener** is used to get notified if there is a change in mouse state. There are five abstract class methods such as,

– mouseEntered, mouseExited, mousePressed, mouseReleased, mouseClicked

iii) **java.awt.event.MouseMotionListener** is used to get notified if the player drags and moves the mouse. There are two abstract class methods such as,

– mouseDragged, mouseMoved

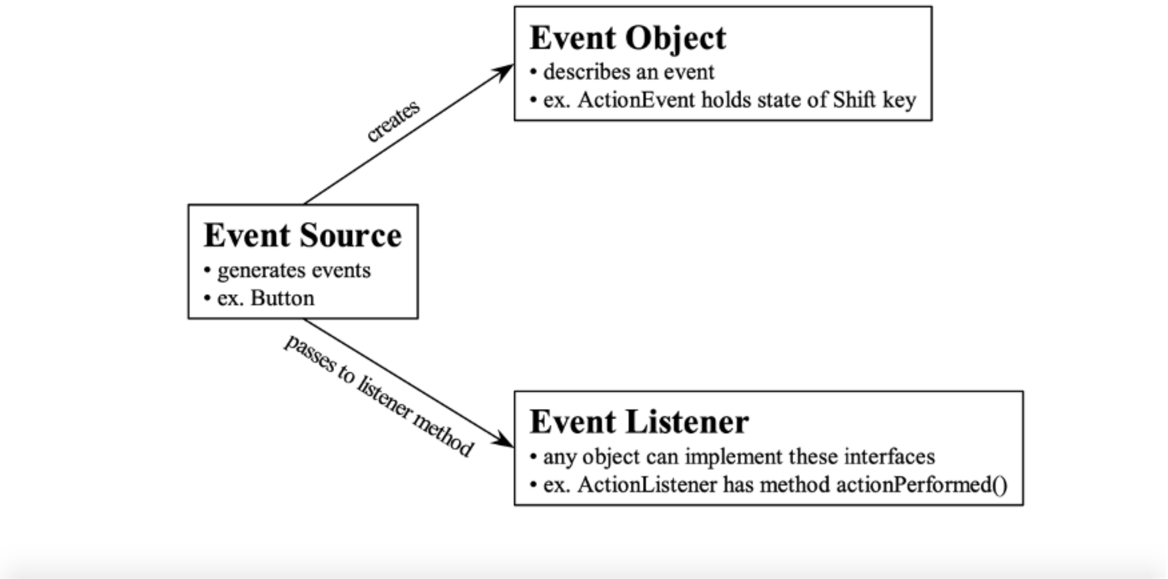
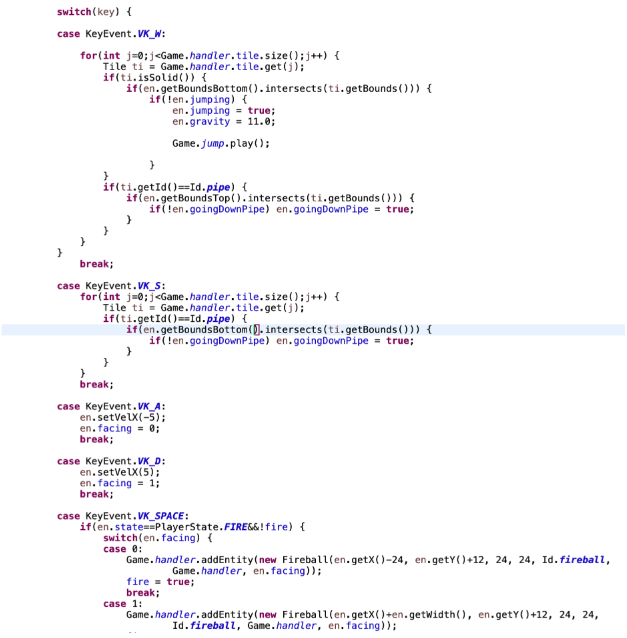
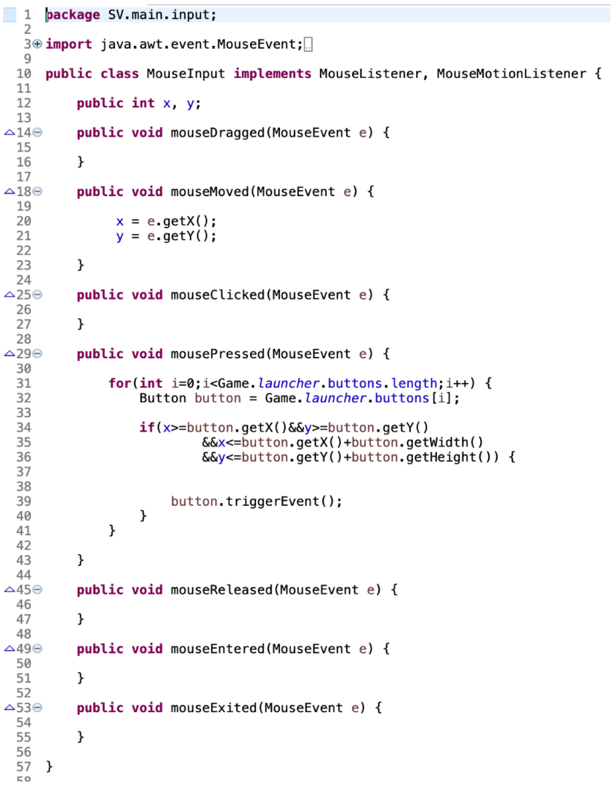

Event Object
• describes an event
• ex. ActionEvent holds state of Shift key
creates
Event Source
• generates events
• ex. Button
passes to listener method
Event Listener
• any object can implement these interfaces
• ex. ActionListener has method actionPerformed()

Figure 5.1: Event Handling



```
switch(key) {

case KeyEvent.VK_W:

    for(int j=0;j<Game.handler.tile.size();j++) {
        Tile ti = Game.handler.tile.get(j);
        if(ti.isSolid()) {
            if(en.getBoundsBottom().intersects(ti.getBounds())) {
                if(!en.jumping) {
                    en.jumping = true;
                    en.gravity = 11.0;

                    Game.jump.play();

                }
        }
        if(ti.getId()==Id.pipe) {
            if(en.getBoundsTop().intersects(ti.getBounds())) {
                if(!en.goingDownPipe) en.goingDownPipe = true;
            }
        }
    }
}
    break;

case KeyEvent.VK_S:
    for(int j=0;j<Game.handler.tile.size();j++) {
        Tile ti = Game.handler.tile.get(j);
        if(ti.getId()==Id.pipe) {
            if(en.getBoundsBottom().intersects(ti.getBounds())) {
                if(!en.goingDownPipe) en.goingDownPipe = true;
            }
        }
    }
    break;

case KeyEvent.VK_A:
    en.setVelX(-5);
    en.facing = 0;
    break;

case KeyEvent.VK_D:
    en.setVelX(5);
    en.facing = 1;
    break;

case KeyEvent.VK_SPACE:
    if(en.state==PlayerState.FIRE&&!fire) {
        switch(en.facing) {
        case 0:
            Game.handler.addEntity(new Fireball(en.getX()-24, en.getY()+12, 24, 24, Id.fireball,
                    Game.handler, en.facing));
            fire = true;
            break;
        case 1:
            Game.handler.addEntity(new Fireball(en.getX()+en.getWidth(), en.getY()+12, 24, 24,
                    Id.fireball, Game.handler, en.facing));
```

Figure 5.2: Keyboard Event

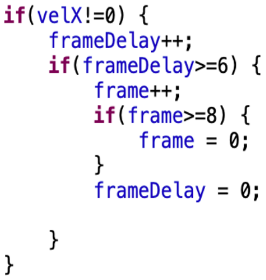


```java
package SV.main.input;

import java.awt.event.MouseEvent;

public class MouseInput implements MouseListener, MouseMotionListener {

    public int x, y;

    public void mouseDragged(MouseEvent e) {

    }

    public void mouseMoved(MouseEvent e) {

         x = e.getX();
         y = e.getY();

    }

    public void mouseClicked(MouseEvent e) {

    }

    public void mousePressed(MouseEvent e) {

        for(int i=0;i<Game.launcher.buttons.length;i++) {
            Button button = Game.launcher.buttons[i];

            if(x>=button.getX()&&y>=button.getY()
                    &&x<=button.getX()+button.getWidth()
                    &&y<=button.getY()+button.getHeight()) {


                button.triggerEvent();
            }
        }

    }

    public void mouseReleased(MouseEvent e) {

    }

    public void mouseEntered(MouseEvent e) {

    }

    public void mouseExited(MouseEvent e) {

    }

}
```

Figure 5.3: Mouse Event

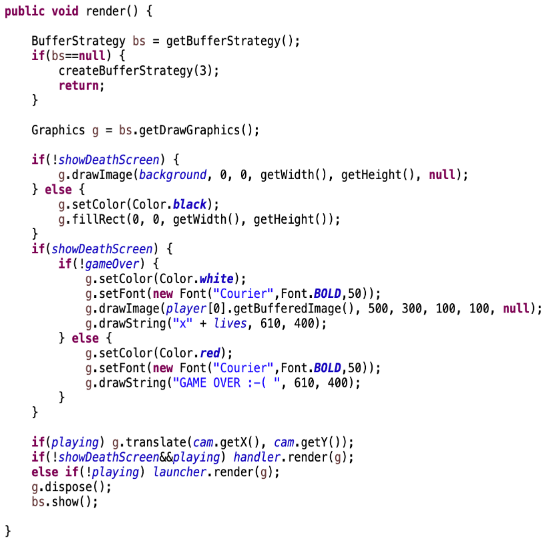
- **Animation:** It included animations, for instance, walking animation, enemy walking animation, Etc. in the game so that it looks more dynamic and attractive to the player(Fan, 1996).

  Animation loop: Animation loop is applied to updates the animation of assets and displays through the current frame on the screen(Fan, 1996).



```java
if(velX!=0) {
    frameDelay++;
    if(frameDelay>=6) {
        frame++;
        if(frame>=8) {
            frame = 0;
        }
        frameDelay = 0;

    }
}
```

Figure 5.4: Animation w.r.t Frames



```java
public void render() {

    BufferStrategy bs = getBufferStrategy();
    if(bs==null) {
        createBufferStrategy(3);
        return;
    }

    Graphics g = bs.getDrawGraphics();

    if(!showDeathScreen) {
        g.drawImage(background, 0, 0, getWidth(), getHeight(), null);
    } else {
        g.setColor(Color.black);
        g.fillRect(0, 0, getWidth(), getHeight());
    }
    if(showDeathScreen) {
        if(!gameOver) {
            g.setColor(Color.white);
            g.setFont(new Font("Courier",Font.BOLD,50));
            g.drawImage(player[0].getBufferedImage(), 500, 300, 100, 100, null);
            g.drawString("x" + lives, 610, 400);
        } else {
            g.setColor(Color.red);
            g.setFont(new Font("Courier",Font.BOLD,50));
            g.drawString("GAME OVER :-( ", 610, 400);
        }
    }

    if(playing) g.translate(cam.getX(), cam.getY());
    if(!showDeathScreen&&playing) handler.render(g);
    else if(!playing) launcher.render(g);
    g.dispose();
    bs.show();

}
```

Figure 5.5: Animation Loop in Render Method

- **Painting Graphics:** By using the concept of method overriding, it overrides the paint method to paint the graphics(Fan, 1996).

  i) **java.awt.Graphics** class is usually used for painting customisation. It handles the graphics context (such as color, font, size of text, Etc.) and implements methods for the rendering of two basic types of graphical objects:

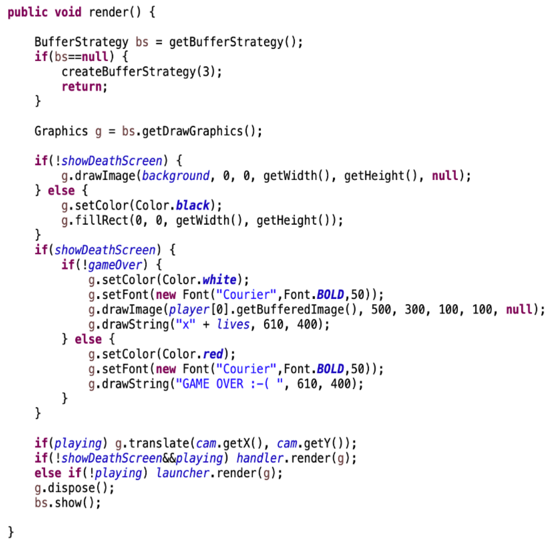
  – Texts: drawString().

  – Bitmap images: drawImage().

  ii) The Graphics class enables the programmer to get or set the attributes of the context of graphics:

  – Font such as- setFont(), getFont()

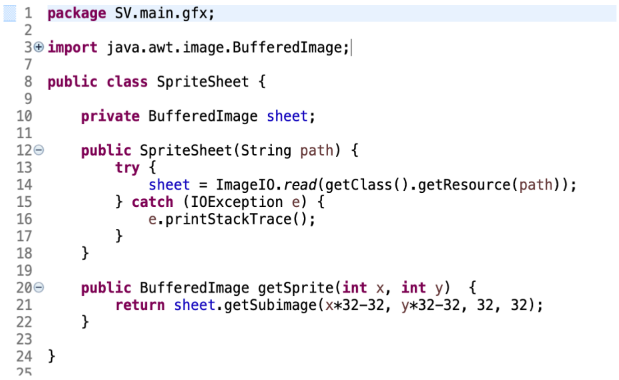
  – Color such as- setColor(), getColor()



```
public void render() {

    BufferStrategy bs = getBufferStrategy();
    if(bs==null) {
        createBufferStrategy(3);
        return;
    }

    Graphics g = bs.getDrawGraphics();

    if(!showDeathScreen) {
        g.drawImage(background, 0, 0, getWidth(), getHeight(), null);
    } else {
        g.setColor(Color.black);
        g.fillRect(0, 0, getWidth(), getHeight());
    }
    if(showDeathScreen) {
        if(!gameOver) {
            g.setColor(Color.white);
            g.setFont(new Font("Courier",Font.BOLD,50));
            g.drawImage(player[0].getBufferedImage(), 500, 300, 100, 100, null);
            g.drawString("x" + lives, 610, 400);
        } else {
            g.setColor(Color.red);
            g.setFont(new Font("Courier",Font.BOLD,50));
            g.drawString("GAME OVER :-( ", 610, 400);
        }
    }

    if(playing) g.translate(cam.getX(), cam.getY());
    if(!showDeathScreen&&playing) handler.render(g);
    else if(!playing) launcher.render(g);
    g.dispose();
    bs.show();

}
```

Figure 5.6: Image Rendering in Main File

- **Sprites:** Sprites are the representation of the image to render animation and update the image. There are several types of sprites such as text, graphics, bitmaps Etc. We can create multiple sprites in a single sheet called SpriteSheet. To render single sprite from the given SpriteSheet, create a method to allow the location of the image and BufferedImage get that single sprite to perform rendering(Fan, 1996).



```
package SV.main.gfx;

import java.awt.image.BufferedImage;

public class SpriteSheet {

    private BufferedImage sheet;

    public SpriteSheet(String path) {
        try {
            sheet = ImageIO.read(getClass().getResource(path));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public BufferedImage getSprite(int x, int y)  {
        return sheet.getSubimage(x*32-32, y*32-32, 32, 32);
    }

}
```

Figure 5.7: Sprite Implementation

i) A sprite consist of x and y coordinates. It is defined by the existing location as well as movement in space.

To reserve sprite in constant movement, it can be achieved by(Fan, 1996):

- dx=vx*dt in x-position

- dy=vy*dt in y-position

where dt = elapsed time

ii) There are two behaviours of sprites:

- Painting: It is used to paint the section on the screen.

- Updating: Update the sprite to perform animation or movement.

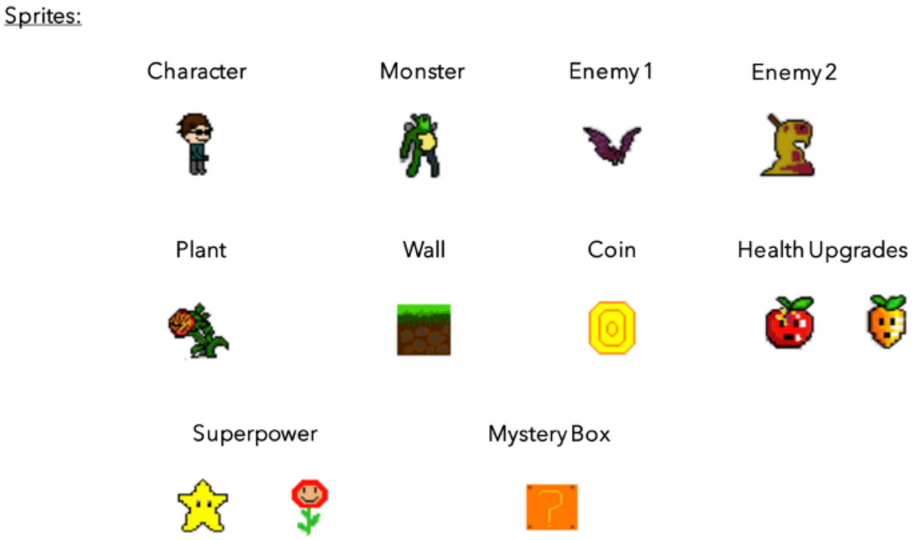
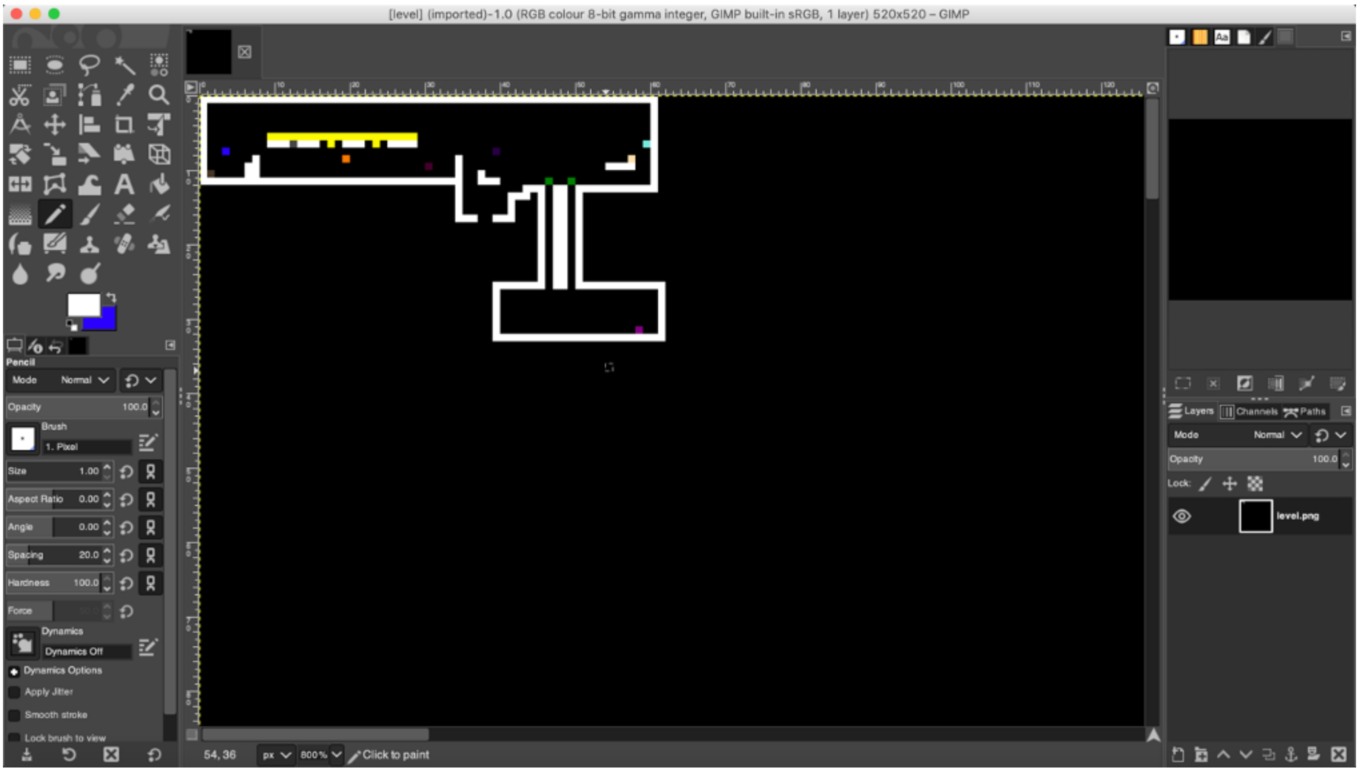


Figure 5.8: Sprites of Assets

- **Loading Map:** Using GIMP 2.10 software, It supports to design levels and set RGB value for each entity that holds sprites.



[level] (imported)-1.0 (RGB colour 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 520x520 – GIMP

Pencil

Mode Normal

Opacity 100.0

Brush 1. Pixel

Size 1.00

Aspect Ratio 0.00

Angle 0.00

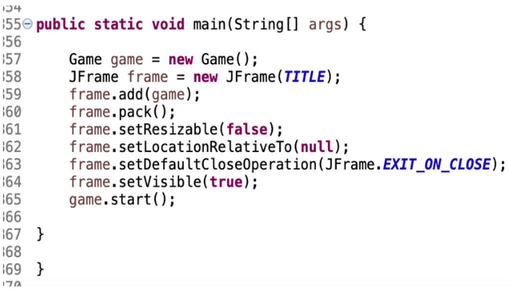
Spacing 20.0

Hardness 100.0

Dynamics Off

Figure 5.9: Level Design Using GIMP 2.10

- **Main Method:** It has a separate thread for the game by implementing the Runnable Interface. It started with the thread in a separate start() method and called it within the main() method(Fan, 1996).



```java
public static void main(String[] args) {

    Game game = new Game();
    JFrame frame = new JFrame(TITLE);
    frame.add(game);
    frame.pack();
    frame.setResizable(false);
    frame.setLocationRelativeTo(null);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
    game.start();

}

}
```

Figure 5.10: Main Method

Java Threads: To perform a game loop without exiting the game, a thread is used to continue the game.

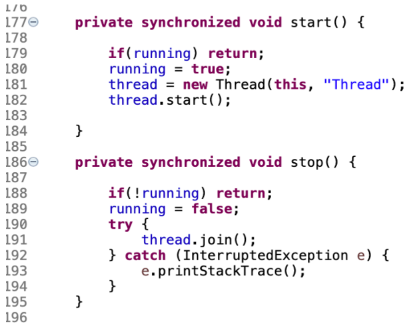
The **java.lang.Thread** class is the base class for all thread objects. To generate a separate thread of implementation, you need to build an instance of the Thread class or one of its subclasses(Fan, 1996). Below, estimates the initialisation of Multiple Threads.

Thread t = new Thread(); ....//create Thread

t.start(); ....// start running Thread t

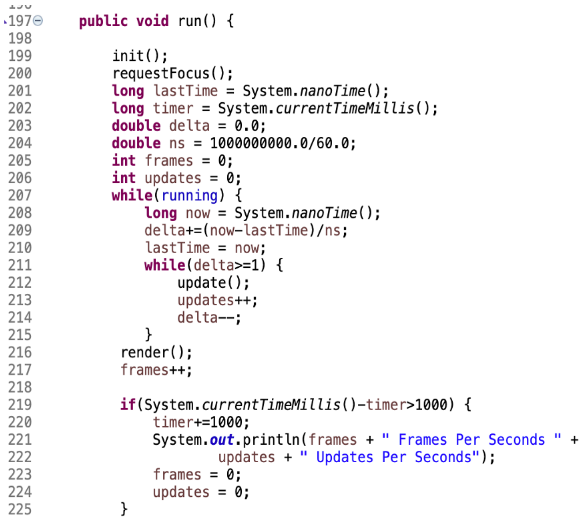
t.stop(); ....// stop running Thread t

A class that executes the Runnable interface which performs the definition of a method called run().



```
private synchronized void start() {

    if(running) return;
    running = true;
    thread = new Thread(this, "Thread");
    thread.start();

}

private synchronized void stop() {

    if(!running) return;
    running = false;
    try {
        thread.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

Figure 5.11: Start and Stop Methods



```
public void run() {

    init();
    requestFocus();
    long lastTime = System.nanoTime();
    long timer = System.currentTimeMillis();
    double delta = 0.0;
    double ns = 1000000000.0/60.0;
    int frames = 0;
    int updates = 0;
    while(running) {
        long now = System.nanoTime();
        delta+=(now-lastTime)/ns;
        lastTime = now;
        while(delta>=1) {
            update();
            updates++;
            delta--;
        }
     render();
     frames++;

     if(System.currentTimeMillis()-timer>1000) {
         timer+=1000;
         System.out.println(frames + " Frames Per Seconds " +
                 updates + " Updates Per Seconds");
         frames = 0;
         updates = 0;
     }
```

Figure 5.12: Run Method

Chapter Six

Evaluation

Virus Survivor is designed to provide open guidance on safety measures to combat Covid-19 pandemics such as PC-based Java programming project. Even though we cannot find games designed to raise general health awareness about an epidemic, some health-related games are based on similar factors such as learning outcomes, usability, and customer experience(Katsaridou and Thibault, 2016). Similarly, some educational games are rated based on usability and gaming experience. This is why I evaluated Virus Survivor to understand the ease of use and gaming experience.

Firstly, I watched tutorials to learn what procedures are required to build the game. Secondly, Game representation should be required so that players can attract towards this game and play it as fun. Thirdly, creating sprites using GIMP software, made it easy to construct assets. Lastly, with the concept of GUI library, i.e. AWT with their features has helped me to build the VIRUS SURVIVOR game successfully. To play this game, users need to download the Java framework (Eclipse) and download a zip file of source code in order to run this game. You need 4GB RAM compatibility and 2GB graphics card in the system.

This game has run successfully from four users who played the game. This game runs completly without any glitches and game crash.

Chapter Seven

Comparison

7.1 Comparison between AWT and SWING

- The main difference of AWT with other GUI libraries is, it is platform-dependent.

- It is more flexible than other GUI libraries, and it has more components such as checkbox, textbox, button Etc., that helps to customize controls manually.

- AWT GUI library is more useful if the user is relying on only one platform and in one device. It gives good outcomes while running the project.

- Java SWING is more advanced and optimized the latest version of the AWT GUI library, and it is platform-independent. However, some AWT classes are used in SWING. AWT is the older version yet more reliable and flexible than SWING or JavaFX.

- Java AWT supports a wide range for creating graphics for a gaming product.

- If the programmer is creating a Java 2D game, this AWT libraries support Graphics2D to run the game successfully and gives derived options of components to build the game.

7.2 Comparison with Other Games

7.2.1 LOST Game:

- This game is motivated by other game such as "Metal Slug". This project has made by three members Muhammad Saad Hussaini, Takreem Masood and Muhammad Ubaid Ullah. This is java 2D game has meant to be as an entertainment purpose that character shoots enemies and met with different challenges.

- This game starts with a dynamic background with menu logics. Therefore, the background image continuously rotates in a circular motion, and the menu has static logos and buttons.

- It is similar to my project like it follows Event Handling, Animations, Painting Graphics, Main Character Movement, Map Loading and the Main Method.

- This game runs in NetBeans Java Framework, and GUI used in this project are SWING and AWT.

About this game, especially Map Loading, they built the whole level by mapping with a text file and then rendering it into the game. The text file contains some digits which are associated with an individual tile, and each tile has a unique digit. Also, they did work on credits option and information about how to play the game. So that player can easily understand.

Additionally, project members have created separate .png files for every movement of character and enemy (like jump, left, right, Etc.). In the source code, they have created separate mouse and keyboard event class for both character and enemy for better performance.

More additional information about source code, go to https://github.com/SHEHANhasintha/LOST--Java-2D-Game

7.2.2 Super Mario Game:

- This game is Mario based Java 2D game with same concept of creatures and maps. This project has created by Mohamed Talaat. This game runs in NetBeans Java Framework and also, he uploaded JAR file. The GUI used in this project are SWING and AWT.

- When player runs this game, it looks similar to main MARIO game with same character, enemies and rewards. It splashes menu option to start or exit the game. It has same movements and controls like classic MARIO to hit hidden wall, jump on enemy to kill them, superpowers like invinsibility, size growth and so on.

More additional information about source code, go to https://github.com/mtala3t/Super-Mario-Java-2D-Game

Chapter Eight

Conclusion

This study illustrates the relationship between agile development and organizations in the computer game industry. With rapid evolution widely used by IT organizations, it is no surprise that it is also the most popular method amongst managers in the computer game industry. The most obvious component of agile that is considered in the hypothesis is the focus. One particular analysis was the impact of cycles on some levels of the PC gaming industry and the ability to replace agile focus. In particular, these layers are the means by which cycles affect the quality of the worker's attendance, affect the assets of the Organisation, and hinder the project's imagination and vision. An actor's quality of life is affected by the presence of breakouts. Because they regularly have to remain unpaid after 40 hours.

VIRUS SURVIVOR is based on the theme of resistance. This game is designed to allow players to follow welfare measures in order to stay protected from Covid-19 when they are not around. Therefore, the adjustments to the model created for VIRUS SURVIVOR hope to inform the player about the safety precautions to be taken in connection with Covid-19. Although this game was designed for younger age in society, it is usually played by a wide audience, regardless of social or educational background. Given the time young people spend today on mobile apps and PC games, and the impact games have on player behaviour, it is beneficial to introduce multipurpose games that help raise public health awareness.

VIRUS SURVIVOR aims to educate individuals about the social measures they should follow during Covid-19 and encourage them to protect their safety measures gradually. The requirement for the game is to equip the player with a PC Game with Java framework(Eclipse) and memory compatibility. VIRUS SURVIVOR is ranked based on quality, usability, value, and gaming experience. The assessment was conducted by 4 people, with all of them receiving positive reviews willing to explain VIRUS SURVIVOR to their peers. This game can also run in other Java framework too. I will improve the game by increasing the number of levels and level problems in the game. You can also change sprites for future web games that differ from the individual single-player variants of the current game. I will also be changing the controls of the game by improving the features that affect the controls, as suggested by the members of the review. In addition, I will add story-line mode, so that player can understand the game.

Appendix One

First Appendix

Git Repository: https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2019/sxb1494

Project File Structure:

In this project, the source code has written in Java files and Java Frameworks(Eclipse). Each file has java classes with the following description.

- SV.main: It has the main class and other java and enum classes. To run the game, make sure Game.java class should exist.
- SV.main.entity: It has source code for entity classes.
- SV.main.entity.mob: It has source code for character and enemies with animation, rendering and updating method.
- SV.main.entity.powerUp: It contains source code for superpower entities to provide extra features to character.
- SV.main.gfx: It contains Sprites and SpriteSheet source code to perform the BufferedImage method.
- SV.main.gfx.gui: It has source code for menu launcher and button to splash screen of the main menu before entering the game.
- SV.main.input: This folder handles source code of Event Handling to perform movements of character by pressing keyboard and mouse button.
- SV.main.states: It contains enum classes with constant variables to identify each entity separately.
- SV.main.tile: It has source code for tile sprites.

Running the code:

- Firstly, To run the game, the user needs to download the Eclipse Java Framework. Then download the zip file from GitLab and import it through the framework. After that, open the source code of Game.java from SV.main folder.

- Second main step is, link the location of the res folder inside the game. Follow the procedure as follows,

  Click right button on SV Game Project -> Properties -> Java Build Path -> Libraries -> Click classpath -> Add class folder -> Select res folder -> Click OK -> Click Apply and Close.

- Lastly, click run option to run the project and play the game successfully.

For more information, it can be found in README.md file in the GitLab Repository.

References

Ananda, Bagus Fikri and Ahmad Chusyairi (2019). "Perancangan Game Virus Survivor Untuk Pendidikan Kesehatan Dengan Metode Game Development Life Cycle". In: *JTIM: Jurnal Teknologi Informasi dan Multimedia* 1.2, pp. 78–84.

Azevedo, Gabriel and Victor Sarinho (2019). ""Aedes Vs. Repellents": Applying Tower Defense Dynamics in a Digital Game to Combat the Mosquito". In: *Joint International Conference on Entertainment Computing and Serious Games*. Springer, pp. 393–397.

Bosman, Frank (2020). "To Dream a Game. Dreams as Interactive-Narrative Devices in Digital Games". In: *gamevironments*.

Cao, Xiaoling et al. (2020a). "Cholesterol: A new game player accelerating endothelial injuries caused by SARS-CoV-2?" In: *American Journal of Physiology-Endocrinology and Metabolism*.

— (2020b). "Immunometabolic Cross-Talk and Regulation of Endocrine and Metabolic Functions: Cholesterol: A new game player accelerating vasculopathy caused by SARS-CoV-2?" In: *American Journal of Physiology-Endocrinology and Metabolism* 319.1, E197.

Dolan, Patrick T, Zachary J Whitfield, and Raul Andino (2018). "Mechanisms and concepts in RNA virus population dynamics and evolution". In: *Annual Review of Virology* 5, pp. 69–92.

Down, Nathalie Claire (2016). "SPECULATIVE GAME DESIGN FOR NUCLEAR DISARMAMENT". In:

Fan, Joel (1996). "Black art of Java game programming". In:

Geraci, Robert M, Nat Recine, and Samantha Fox (2016). "Grotesque gaming: the monstrous in online worlds". In: *Preternature: Critical and Historical Studies on the Preternatural* 5.2, pp. 213–236.

Katsaridou, Maria and Mattia Thibault (2016). "Architextuality and Video Games: A Semiotic Approach". In: *Contemporary Research on Intertextuality in Video Games*. IGI Global, pp. 253–272.

Lee, Dong-Hun et al. (2018). "Transmission dynamics of highly pathogenic avian influenza virus A (H5Nx) clade 2.3. 4.4, North America, 2014–2015". In: *Emerging infectious diseases* 24.10, p. 1840.

Li, Chunfeng and Genhong Cheng (2020). "Will Hydroxychloroquine Still Be a Game-Changer for COVID-19 by Combining Azithromycin?" In: *Frontiers in Immunology* 11, p. 1969.

Lucas, Rowan (2019). “Out of the Margins: Evolving Narrative Representation of Women in Video Games”. In:

Morin-Simard, Andréane (2016). “Gamers (Don’t) Fear the Reaper: Musical Intertextuality and Interference in Video Games”. In: *Contemporary Research on Intertextuality in Video Games*. IGI Global, pp. 71–91.

Ntokos, Konstantinos (2020). “CodePlay: A Tabletop Role-Playing Game System used in Teaching Game Programming Using Content Gamification”. In: *The Computer Games Journal*, pp. 1–16.

Ondarza, José de (2018). “A Game-Based Approach to Teaching Concepts of Infectious Disease.” In: *Bioscene: Journal of College Biology Teaching* 44.2, pp. 10–16.

Orji, Rita et al. (2018). “Personalizing health theories in persuasive game interventions to gamer types: an African perspective”. In: *Proceedings of the Second African Conference for Human Computer Interaction: Thriving Communities*, pp. 1–12.

Pereira, Igor Muzetti and Vinıcius Victor Lelis (2019). “Developing a open-source serious game for control and education on HPV and Cervical Cancer”. In: *Anais do XVI Congresso Latino-Americano de Software Livre e Tecnologias Abertas*. SBC, pp. 124–127.

Radchenko, Simon (2020). “Metamodern Gaming: Literary Analysis of The Last of Us”. In: *Interlitteraria* 25.1, pp. 246–259.

Rita, WL et al. (2020). “Acceptance Level of Older Chinese People Towards Video Shooting Games”. In: *International Conference on Human-Computer Interaction*. Springer, pp. 707–718.

Sirparanta, Niina (2018). “Female game characters of the future”. In:

Sklyarova, AB and A Yu Volkova (2019). “CARRYING OUT TOURS-QUESTS IN RUSSIA”. In: *Индустрия туризма: возможности, приоритеты, проблемы и перспективы* 14.3, pp. 298–302.

Søndergaard, Dorte Marie (2018). “The multible, volatile and ambiguous effects of children’s and young people’s digital play”. In: *Cultures of Computer Game Concern*. Transcript Verlag, pp. 165–184.

VIRUS, ZIKA (2017). “Clinical Rounds”. In:

Wilson, Jo Ellen et al. (2018). “Computerized Cognitive Rehabilitation in Intensive Care Unit Survivors: Returning to Everyday Tasks Using Rehabilitation Networks–Computerized Cognitive Rehabilitation Pilot Investigation”. In: *Annals of the American Thoracic Society* 15.7, pp. 887–891.

Zan, Arman (2018). “Domestication and Foreignization in The Turkish Translation of Video Games”. MA thesis. Sosyal Bilimler Enstitüsü.