UNIT 8: COMPUTER GAMES DEVELOPMENT

Learning Aim B: Design a computer game to meet client requirements

BY

ARIYA BAYAT

YEAR 12

WALWORTH ACADEMY

Unit 8: Computer Games Development

Name: Ariya Bayat
Coursework: Computer Gaming Technologies

Contents

List of Figures

Unit 8: Computer Games Development

Name: Ariya Bayat
Coursework: Computer Gaming Technologies

Chapter 1

Background and Motivation

Game development has always been my passion ever since I was 12 years old. The biggest part of game creation that I enjoy is designing an idea that you can share for the world to see, receiving feedback from the audience that would help me improve my game.

At the beginning of my game development journey, I was inspired to make games at the age of 12, which is when I started to watch Unity tutorials on how to get started with making games. My favourite YouTube channel that I liked to learn from was ‘Brackeys’ which included all different kinds of game making tutorials from 2D to 3D which intrigued me the most. After experimenting and creating different types of projects like a clone of Super Mario Bros, horror simulator and obstacle dodging game, I started created my most ambitious game yet called ‘Tap Fever’ which I then released on the Google Play Store on 12 July 2017. I later changed my programming language from JavaScript to C# since Unity no longer supports JavaScript.

The computer game that I will be making for this coursework is inspired by the dodging style gameplay of my old dodging game and its title ‘Shoot Fever’ is linked to my first ever published game ‘Tap Fever’. In the future, I plan to participate in competitions where developers have to produce a game within a week which will then be reviewed by other game developers, including Brackeys. I believe that this would have a huge impact on my games development skills as it will allow me to get used to planning my time more effectively and efficiently to being more productive in the limited time that I am given. When creating a game, you must first design it, planning out the structure and concept of the game with story boards and sketches. that would meet with the user requirements. The internal processes that would be occurring during the game is another thing to consider when designing a game. It is important for designers to learn the techniques and skills required for designing games effectively.

Name: Ariya Bayat
Coursework: Computer Gaming Technologies

Chapter 2

P3 & M2 | What is The Game?

2.1 Introduction

In this chapter, we will present our game idea and explain how it works, including brief explanation and details about its: controls, characters, hazards, collectibles, level design, etc.

2.1.1 Brief Explanation

The game is a 3D arcade-based dodging game, called 'Shoot Fever'. In this game, you get to name your own character who you will guide through an obstacle course, dodging hazards, collecting coins, and shooting down the obstacles in your way! The goal of the game is to survive as long as possible, as you try to beat your best score. The concept of the game GUI can be below shown in Figure 2.1.

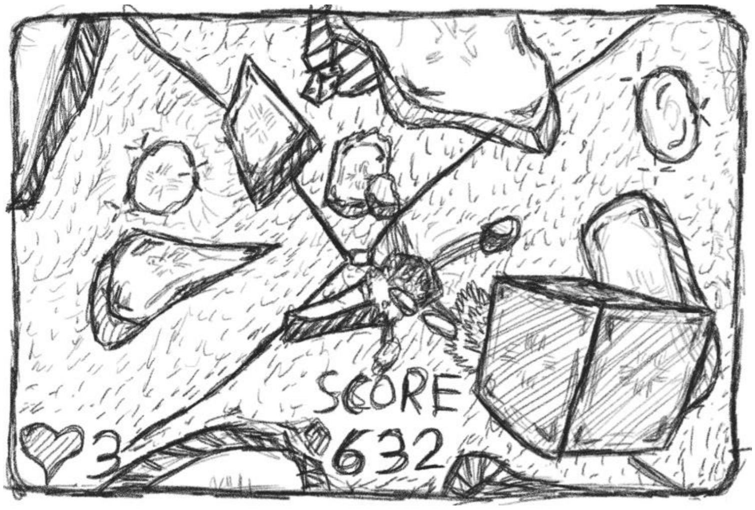


Figure 2.1: Concept of game GUI

2.1.2 Audience and Requirements

The game will be targeting users from ages 8 to 12, so there is a lot to consider when designing this game as it must meet their requirements, including: the controls, goal, challenges, collectible, and rewards.

In general, we are expected to present to our gamers a fun and exciting game to play. The game must also be visually appealing. We can see successful dodge-based games like 'Subway Surfers' achieve these needs. This entertaining game with over one billion downloads features an endless gameplay of a 'surfer' collecting coins and dodging trains in a train station themed environment. The simple but fun gameplay complimented with the cartoonish style is a suitable mix for kids to enjoy the experience as their requirements are met.

2.1.2.2 Controls

Even though this is a 3D-based game, the player is only limited to move around horizontally and vertically as they will be constantly approaching towards the end of the course. However, your player is gifted with a shoot button, enabling you to fire at hazards and destroy them for extra points. These controls will meet with the user's requirements as they are very simple and quite easy for them to learn quickly during gameplay. The concept of the player can be shown below in Figure 2.2.

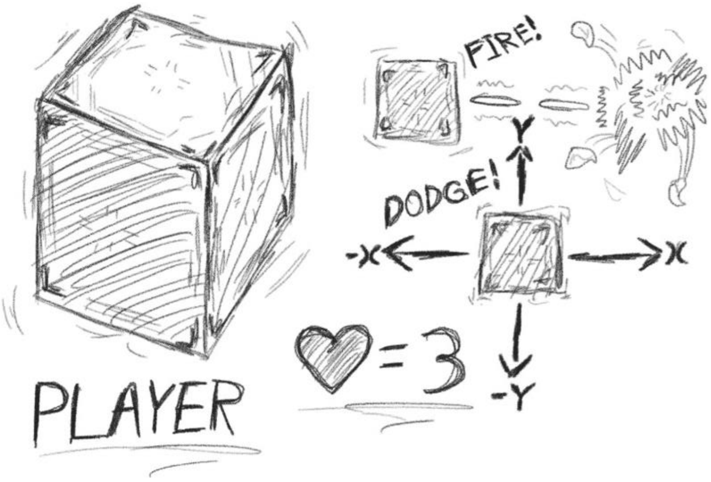


Figure 2.2: Concept of player

2.1.2.3 Goal

Every game has some type of goal, without one you would never be able to complete the game. In our game, the goal is simply to survive through and reach to the end of the obstacle course. Once you have reached the goal and finished a level, you get to see your final score based on the timer and the hazards you destroyed. You can then see if you have collected enough skin pieces to claim a new skin for your character or you can replay the level and see if you can beat your best score. This goal is acceptable for our audience as it is very simple learn.

2.1.2.4 Challenges

A game without a challenge is pointless, but a game that is too challenging would fall into the 'rage games' category. As we are targeting 8-12-year olds, we want this game to be hard but playable. The challenging aspects of this game are the obstacles (hazards). If your player collides with a hazard, you will lose a life. Harder levels will introduce you to moving hazards, which would be harder to dodge. This is challenge system is suitable for our users as they are simple to overcome, leaving you with two choices for you to fight (shoot) or flight (dodge). The concept of the obstacles can be shown below in Figure 2.3.

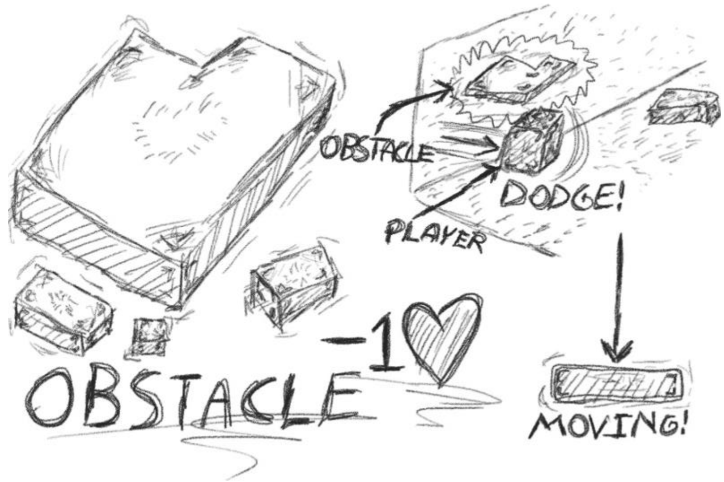


Figure 2.3: Concept of obstacle

2.1.2.5 Collectables and Rewards

Collectable items and rewards are the satisfying aspect of the game: the collectables are the satisfaction factor during gameplay, and the rewards are usually the satisfaction factor after gameplay. As you play the game, the course is riddled with items: coins are commonly found and collecting them will add to your score counter, power up orbs are a lot more uncommon and collection them will increase the fire rate of your shooting, skin pieces are rarely found and collecting a certain amount of the same skin pieces can unlock you a new skin for you to customize your character with. These collectables and rewards meet with the user's requirements as they are fun and easily obtainable. The concept of the items can be shown below in Figure 2.3.

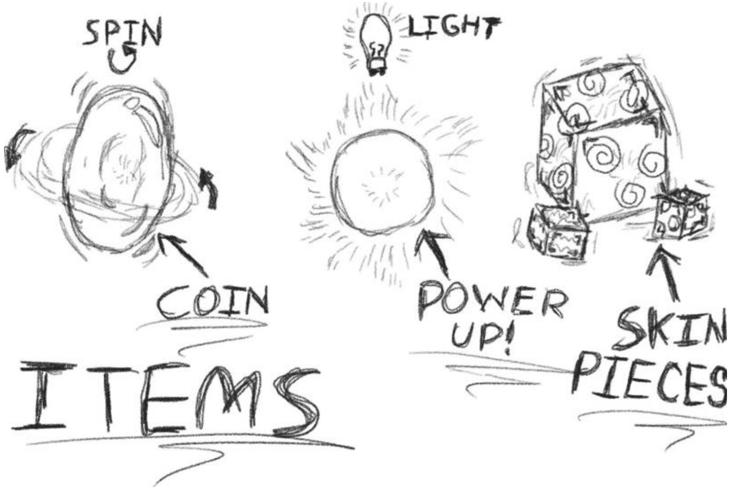
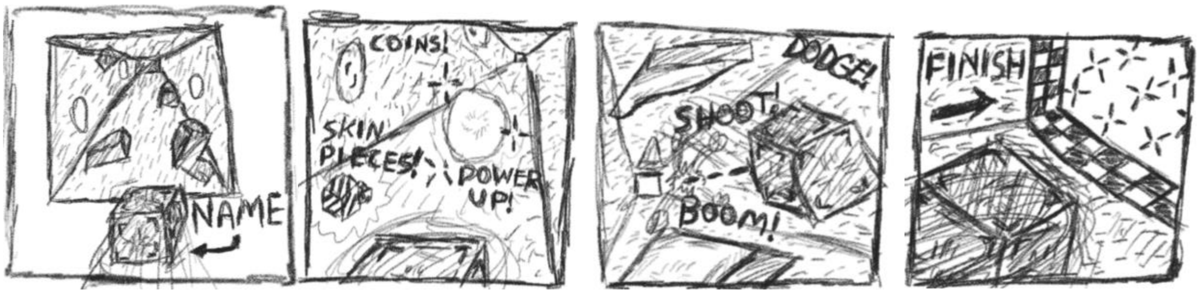


Figure 2.4: Concept of items

2.1.3 Storyboard

A storyboard is a visual representation of a story sequence; designers use this method to illustrate their games in action. This is a storyboard for ‘Shoot Fever’, which includes the beginning of the game, the collectables like coins that you may encounter, as well as the hazards, and the end of the game where the goal is.



This is the beginning of the level, after you have named your character, you will be approaching the tunnel.

As you enter, you can look for and collect ‘coins’ for boosting your score, ‘power up orbs’ for boosting your shooting speed, and ‘skin pieces’ for unlocking new skins for your avatar.

But be careful, as you may come across obstacles that float in your way! You can choose to fight or flight; shoot your way through the course or dodge your way past the hazards. You may also find moving hazards!

This is the end of the level, once you finally reach here, you can hope to beat your best score. If you have collected enough skin pieces, you can go to the shop to unlock and use your new skin for future levels.

The overall concept of the game can be seen below in Figure 2.5, including the player, hazards, collectables, game view at two different perspectives, collectables and the GUI system.

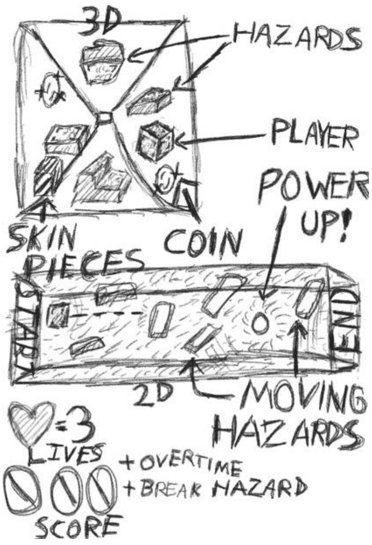


Figure 2.5: Concept of game

Chapter 3

Game Functionality

3.1 Introduction

In this chapter, we will dive deeper into the game to have a clear understanding of how it functions. We can achieve this with two skills: a pseudocode can help us understand how the game functions with simplified programming languages, while a flowchart can help us understand how the game functions visually with shapes and arrows. We will be using both skills to help us understand the internal processes of the game.

3.1.1 Pseudocode

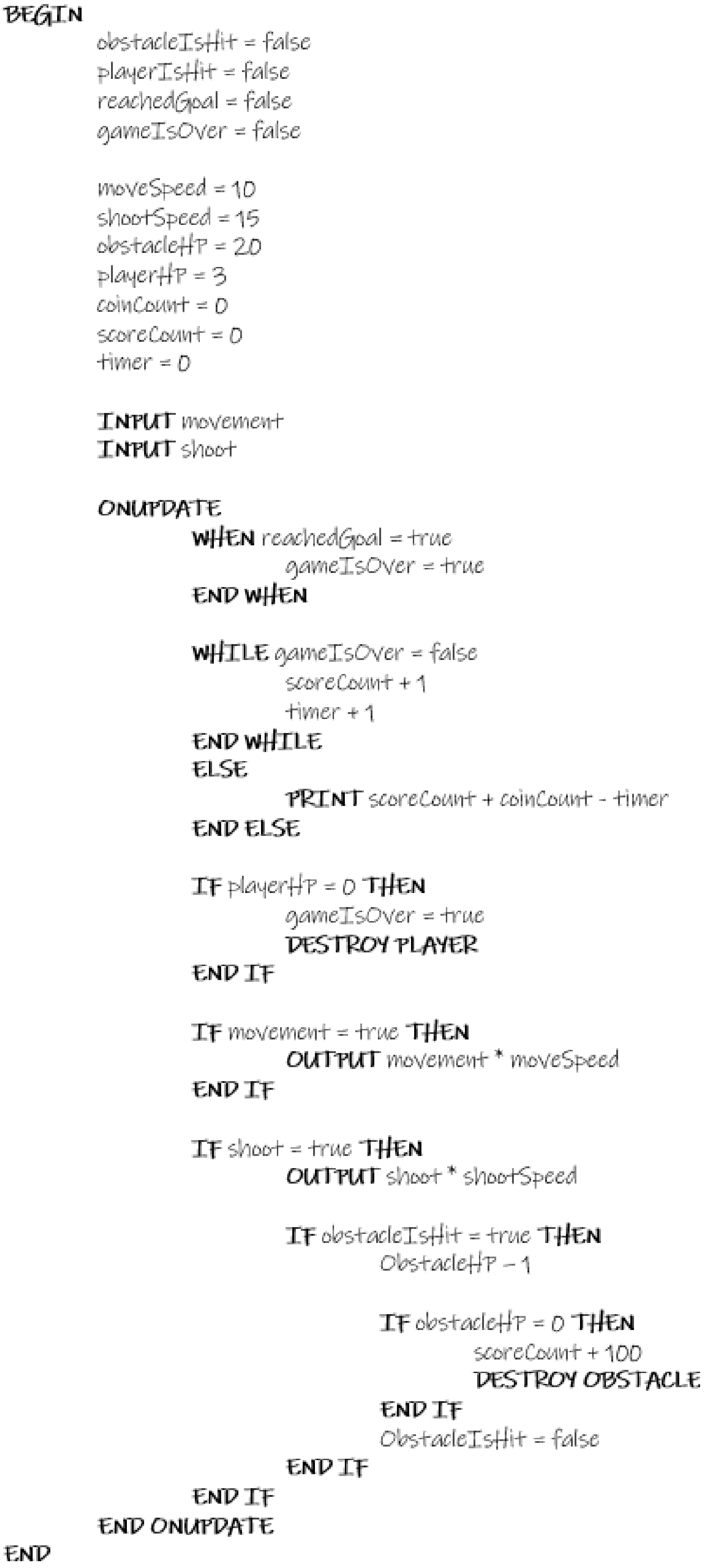
A pseudocode is a skill used by programmers to structure their program codes when developing high-quality software for more effective and efficient approaches to programming. Pseudocode is like a rough sketch of code, with the use of keywords that are categorised in three different types: operations, decisions, and repetitions.

Operation keywords are used to declare actions like when blocks of code 'Begin' and 'End', asking for the user's 'Input' then displaying them an 'Output' like with 'Print', and to 'Read' and 'Write' data.

Decision keywords are used for decision making like checking 'If' and 'When' a statement is true, 'Then' declaring actions, or using 'Else' and 'Else If' to declare other actions.

Repetition keywords are used for repeating processes like looping blocks of code 'For' a specific number of times, repeating them 'While' or 'While something is Not' happening, and setting blocks of code on 'Repeat Until' a statement is true.

Figure 3.1 shows a pseudocode for the game functionality in 'Shoot Fever', which includes the player's controls such as movement and shooting, calculations of the score counter, obstacles, game state, etc.



```
BEGIN
          obstacleIsHit = false
          playerIsHit = false
          reachedGoal = false
          gameIsOver = false

          moveSpeed = 10
          shootSpeed = 15
          obstacleHP = 20
          playerHP = 3
          coinCount = 0
          scoreCount = 0
          timer = 0

          INPUT movement
          INPUT shoot

          ONUPDATE
                    WHEN reachedGoal = true
                              gameIsOver = true
                    END WHEN

                    WHILE gameIsOver = false
                              scoreCount + 1
                              timer + 1
                    END WHILE
                    ELSE
                              PRINT scoreCount + coinCount - timer
                    END ELSE

                    IF playerHP = 0 THEN
                              gameIsOver = true
                              DESTROY PLAYER
                    END IF

                    IF movement = true THEN
                              OUTPUT movement * moveSpeed
                    END IF

                    IF shoot = true THEN
                              OUTPUT shoot * shootSpeed

                              IF obstacleIsHit = true THEN
                                        ObstacleHP – 1

                                        IF obstacleHP = 0 THEN
                                                  scoreCount + 100
                                                  DESTROY OBSTACLE
                                        END IF
                                        ObstacleIsHit = false
                              END IF
                    END IF
          END ONUPDATE
END
```

Figure 3.1: Pseudocode of game functionality

3.1.2 Flowchart

A Flowchart is a type of diagram made up of shapes and arrows used by designers to structure internal processes and calculations when designing their games. Figure 3.2 shows a flowchart for the game functionality in 'Shoot Fever', which includes the functionality of the actual gameplay and the shop.

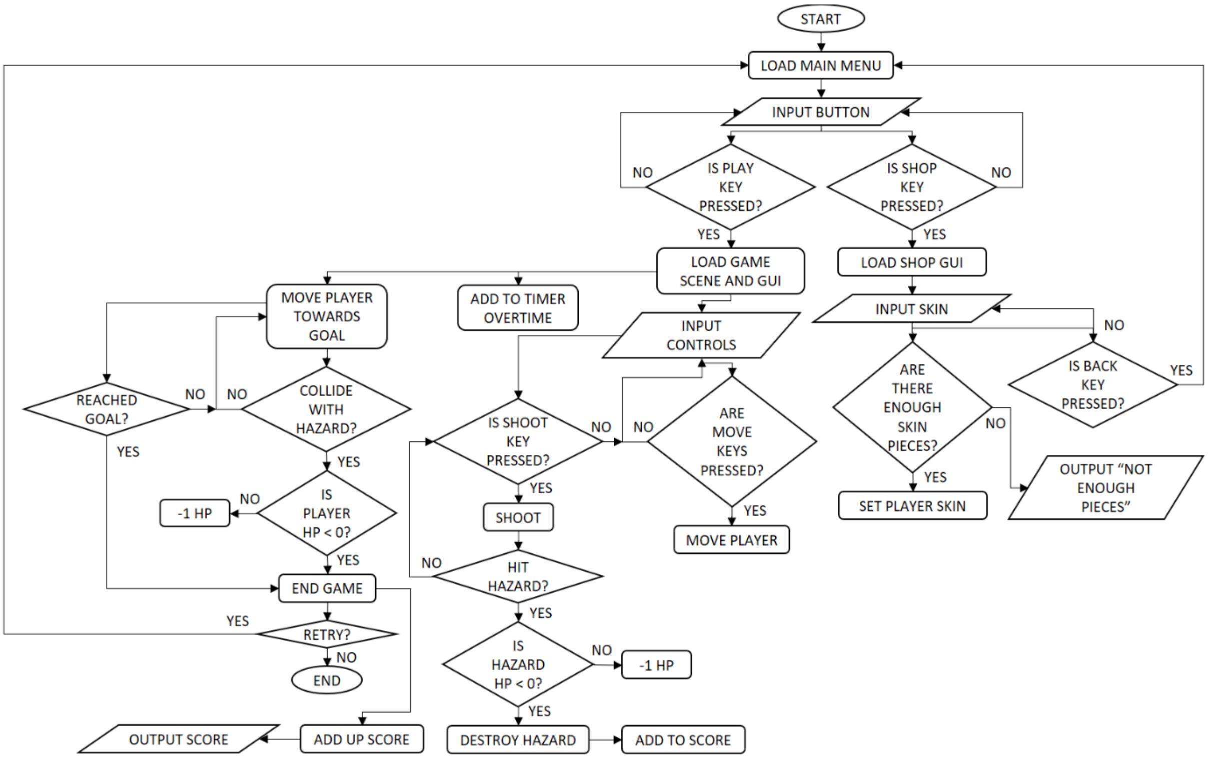


Figure 3.2: Flowchart of game functionality

Chapter 4

Design and Development Decisions

4.1 Introduction

In this chapter we will be looking into the decisions made for the design of the game including feedbacks as well as decisions made for future developing processes including game engines that are most suitable for our game. We will also be looking into the legal and ethical issues that we should be considering before publishing our game.

4.1.1 Design Choices

The designing process of a game can be just as hard as the developing process as we must make sure that our game meets with the user's requirements. An effective way to improve game design is by creating a draft for the game concept, then improving as we create new drafts with feedback from others.

4.1.1.1 First Draft

While I was planning and designing my game, I had to consider the user requirements as the game will specifically be targeting 8-12-year old players. There are many elements in the first draft of the game that I had to change due to the user requirements including the player, controls, and sound effects which are all closely linked.

When designing the player, my first idea for the model was a gun, which would complement the player's ability to shoot and reload. This gun model would also have shooting and reloading animations along with sound effects to make the gun more natural and cooler. However, this is an ethical issue that encourages violence (see content 4.1.1.2 for more information on legal and ethical considerations) which is inappropriate for my young audience. The gun player idea was then scrapped, and the model was later simplified to a cube. I have also removed the reload system and sound effect as this could make the gameplay much more complex, which would not meet with the user requirements.

4.1.1.2 Legal and Ethical Considerations

When creating a game, it is important to make sure that it follows the legal and ethical standards otherwise the game would be considered inappropriate or illegal.

Ethical issues in games are problems that puts the player in a situation to choose between alternatives that must be evaluated as right or wrong. In the games industry, there are many ethical issues that influence a wide range of audience such as violence which is inappropriate for young audiences. Ethical issues in video games can also create bad habits for gamers such as addiction, which can make them less engaged with the real world. However, there are also ethical issues that can make games unethical for all audiences like racism.

Legal issues in games are problems that require a court's decision on whether the game is illegal or not. There are many laws that can protect an author's work from other creators such as copyright, trademark and patent. Copyright protects your content by restricting others from using it without your consent. Trademark on the other hand prevents competitors from using similar names to yours in their own products that could cause confusion with your own consumers. And patent protects you during development, preventing others from copying your idea before you publish it.

4.1.1.3 Feedback

Feedbacks are essential for the process of game design as they can help us develop from our old drafts to an improved version that satisfies the user's requirements. Table 4.1 below shows three reviews from my peers which I will be responding to.

Name	**Feedback**
Damian Mason	"I like the challenging aspect of the game with the hazards, but I think that it would be more challenging if some of the hazards are moving, making it harder for you to dodge them. The player should come across more of these moving obstacles in harder levels."
Joseph Ajose	"I like the concept of the game, but I think that you should add more stuff to the GUI other than just 'lives' and 'score'. A 'level of difficulty' text would be cool, as it let would let the player know the current difficulty of the obstacle course."
Yasin Mahamud	"I like this game. It is simple. However, I think that it would get repetitive in the levels if the only hazards are the obstacles. Maybe, introduce new hazards as the player progresses to make the gameplay much more interesting."

Table 4.1: Table of feedbacks

4.1.1.3.1 Response

In response to Damian, I agree that the obstacles need an upgrade, and I do believe that unbreakable obstacles would be challenging and yet fun to have in the game as the player would only have the option to flight when coming across this type of obstacle other than stationary and moving obstacles. This new unbreakable obstacle meets the user requirements as it would make the game more challenging and exciting as the player comes across new types of obstacles in the game. Figure 4.1 below shows the concept of obstacles 2.0, including the unbreakable obstacle.

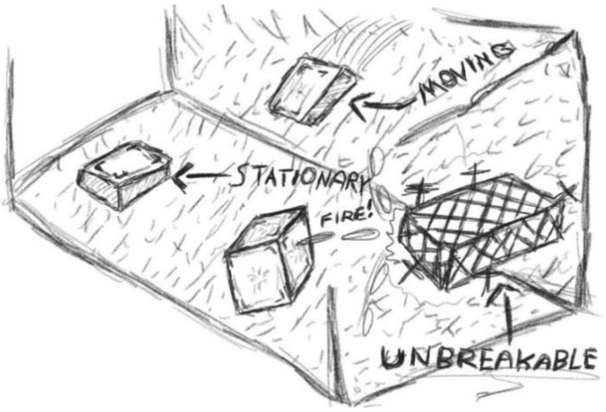


Figure 4.1: Concept of obstacle 2.0

In response to Joseph, I agree that the GUI of the game need an upgrade as well. However, I don't think that it would be necessary to add a 'difficulty of level' indicator as that will be specified when selecting the level. Instead, I added a new counter 'best score' which will display the best score that the user achieved in the level. This will make the game more competitive as players would challenge themselves to beat their high scores. The new GUI meets the user requirements as the amount of information on the screen isn't too overwhelming for the players. I have also changed the layout of the GUI including the "best score", "current score", and "lives" as shown in Figure 4.2 below.

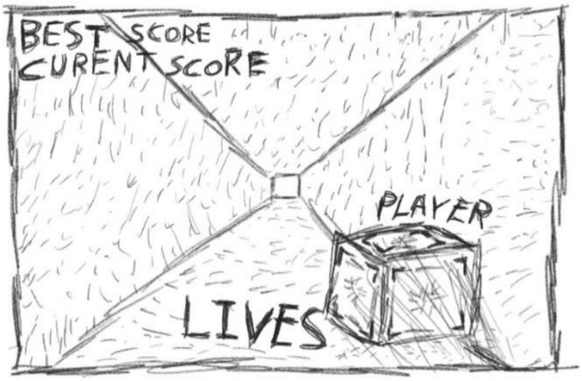


Figure 4.2: Concept of game GUI 2.0

In response to Yasin, I agree that the game needs a new element of hazard to make the game more challenging. I have also been thinking about this while designing the first draft of the game, and I think an enemy would be cool to add to the game as it would be the player's biggest threat in the obstacle course. The enemy would be like a clone of the player: it can move around and dodge bullets, and it can also fire at the player as well. This new enemy hazard meets the user requirements as it is more interesting and unique than other hazards, awarding the player with bonus score when destroying this hazard. Figure 4.3 below shows the concept of enemy.

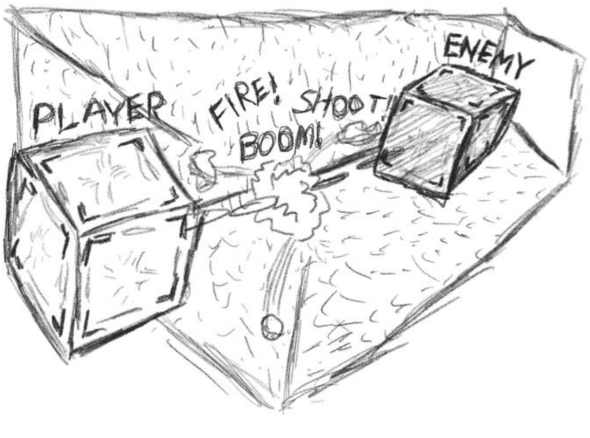


Figure 4.3: Concept of enemy

Unit 8: Computer Games Development

Name: Ariya Bayat
Coursework: Computer Gaming Technologies

4.1.2 Development Choices

When the designing process of the game is finished, it would then be time to start developing. The developing process have two main requirements: a game making platform where you can start creating the game and a programming language that you would be using to make your game function.

4.1.2.1 Game Engine

It is important to choose your game engine carefully as you must make sure that it meets the requirements for making the game. We will be using two examples of game engines to pick which one would be more suitable for the development of the game that we will be creating: Unreal Engine and Unity Engine.

4.1.2.1.1 Unreal VS Unity

Unreal Engine was developed by the American video game company Epic Games and released in 1998. Its initial purpose was for the development of first-person shooter games; However, it has also successfully been used for the other game genres like RPGs, platformers, fighting games, etc.

On the other hand, Unity was developed by video game software development company and released in 2005. Its initial purpose was to be a Mac OS X-exclusive game engine; However, it has been extended to support more than 25 different platforms.

The first most noticeable difference between these two platforms is the comparison of graphical quality, which is where Unreal Engine has the high ground. Unreal is packed with tools and objects like volumetric lights, post processing, and lens flares that you can simply drop into the scene to make the game look stunning. The material editor in this engine also features a node graph where you easily tweak your materials. Also, lighting in Unreal looks a lot more accurate and smoother compared to Unity. We can see these amazing graphics in action with games like Mortal Kombat X.

It is also important to consider if your PC can withstand the powerful graphics of Unreal compared to Unity. Since, Unreal is much more demanding, this can make the game development process a bit slower compared to Unity, making it more it suitable for an average PC.

In terms of programming, Unreal supports C++ while Unity supports C#. In terms of performance, C++ has the edge here. This is the case because C++ can manage memory allocation/deallocation on any objects created which can increase the performance of your game. In terms of usability however, C# is the better choice since it is a higher programming language than C++, making the development speed faster.

In conclusion, I decided to use Unity Engine for developing my game since I have 3 years of experience with this platform as well as the programming language C#. Although I want my game to be visually appealing, I am not aiming for extreme realism since this is a kid's game targeting 8-12-year olds. Also, Unreal Engine may not be suitable for my PC due to its lack of RAM, which would result in performance issues during development.

Unit 8: Computer Games Development

Name: Ariya Bayat
Coursework: Computer Gaming Technologies

Chapter 5

D2 | Evaluation and Conclusion

5.1 Evaluation

Game designing can be just as hard as its development, especially if there are lots of requirements that you need to consider when planning out your game. Throughout the journey of designing my arcade game “Shoot Fever”, there were a lot of changes made to produce a final draft that would satisfy my 8-12-year-old players.

The final draft of the game features an obstacle course where the player can dodge and shoot hazards as they collect coins, power-ups, skin pieces and reach the goal to complete the level. This simplistic gameplay meets the user requirements as it is challenging but fun, introducing new types of obstacles from static to moving to unbreakable and then the ultimate hazard, the enemy. The gameplay meets the requirements since it is also thrilling as the level is riddled with rewards such as coins which can add up to your final score to beat best scores, power-up orbs which can significantly increase the player’s shooting speed and skin pieces which can later unlock new skins for customising the avatar.

The two skills that helped me in my game design process to visualise and plan out the internal processes that would occur during gameplay was the use of pseudocodes and flowcharts. I created a pseudocode of the technical processes and calculations that would happen in the background of the game like the lives and score system, controls, etc. I also displayed these processes visually on a flowchart for a more structured and ordered perspective. This flowchart also included the shop system where the player can change the skin of their avatar.

The legal and ethical issues had a lot for me to consider about when designing my game to ensure its legality and ethicality. There are many ethical issues that encourage gamers that I don’t want to feature in my game like violence which would make the game rated PEGI 16-18, which is above my targeted rating PEGI 8-12. Considering legal issues, I must make sure that I don’t violate protection laws like copyright, trademark and patent as it would result in the removal of my game. I also decided to copyright my game once it is published to prevent others from stealing my content without my permission in my games development career.

To create the final draft for my game, I got feedback from peers to hear their thoughts of the game. They commented on the emptiness of the GUI, a need for an unbreakable obstacle and a new type hazard to prevent repetitive gameplay. In response, I added a new ‘best score’ counter and changed the overall layout of the GUI, I added the new unbreakable obstacle which would be seen in harder levels, and I also added a new AI enemy which mimics the player’s dodging and shooting abilities to make the hazards more interesting.

After finally completing the final draft of the game, I made some decisions on the development process of the game including the game engine and programming language I will be using in the future, considering a suitable development platform and tool that would suit my needs. The game engine that I chose to use over Unreal Engine is Unity Engine. This is the case since my PC can handle Unity better than Unreal and I am not planning to achieve extreme realism as it is not necessary for younger audiences. And for my programming language I choose to use C# since Unity is only limited to supporting C#. Above all, I have 3 years of experience of Unity and C# which would help me adapt to the environment easier than I would with Unreal and C++.

5.2 Conclusion

I believe that the biggest factor for the positive changes made to the first draft of the 'Shoot Fever' game is the feedback you receive from users. The more feedback the design of your game has, the better picture you have on what the audience wants. For instance, we see reviews with ratings of android games listed in the Google Play Store. This does not only help the developers to improve their game, but it also helps gamers decide whether they want to play the game or not based on the ratings and reviews.