The Great Wave

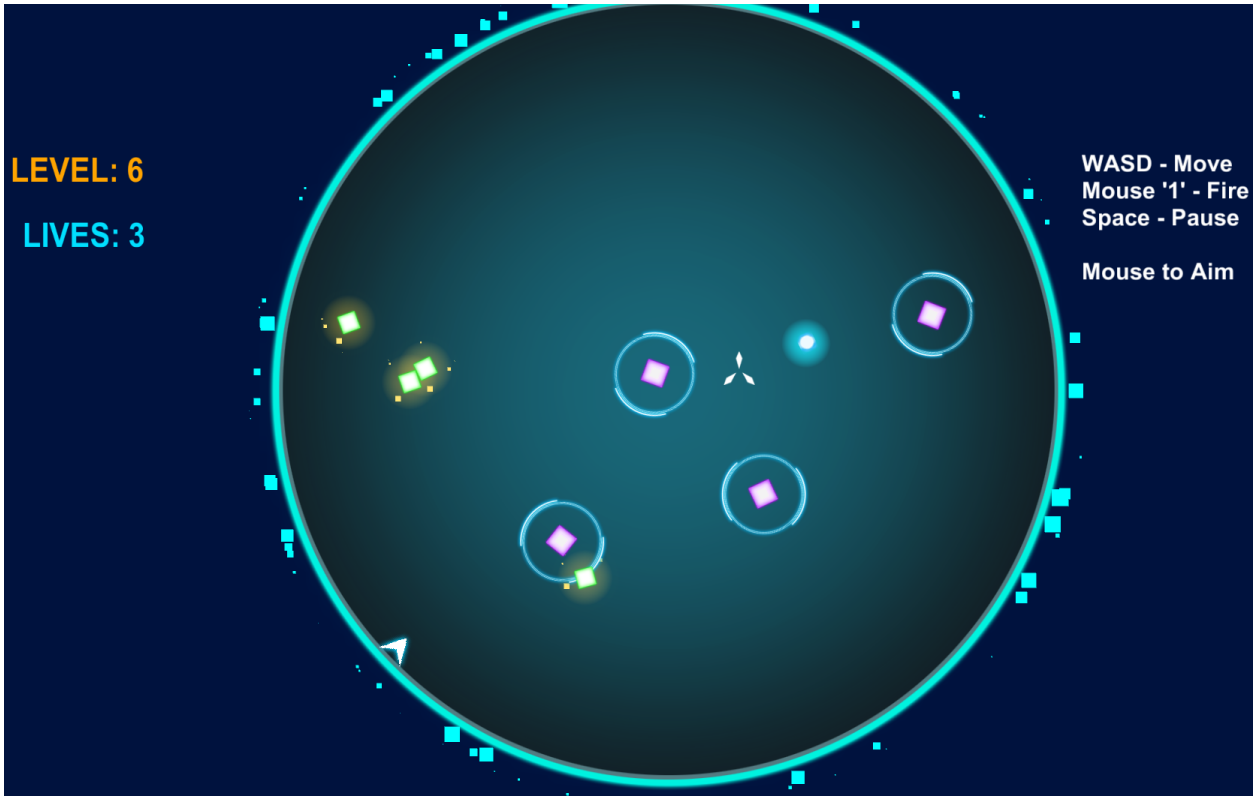
Emily Bao
Michael Leroux
Percy Teng
Kilby Baron

TABLE OF CONTENTS

https://www.dropbox.com/sh/qdte8n47ihpuie9/AADuD9rbmRuM80z60MoqD4Nfa?dl=0
The Game

EXECUTIVE GAME SUMMARY

The Great Wave is a 2D, level-based arcade game where players pass each level by enduring incoming waves of enemy ships. The theme of the game is made to look futuristic with its blend of simple shapes and objects. It's overall feel is intended to be fast paced and similar to an arcade game. One of the most unique aspects of the game are the robust projectiles; player projectiles will continuously bounce off of the game perimeters until they collide with a game object. As a result, the more the player shoots, the more risk he or she can potentially put themselves in. Because of this feature, players must aim their shots instead of wildly firing. This adds a blend of good judgment, tactics, and reflexes to progress through the game's waves of enemies. The goal of the game is for the player to reach the highest level that he or she can. If the player makes contact with their own projectile, an enemy projectile, or collides with an enemy ship, the player will lose one of his or her five lives. Once five lives have been



exhausted, the game will end.

Since our initial concept proposal, the game itself has undergone significant changes. Our initial approach of the game was intended to be 3D; however, after our prototyping phase we decided to build the game in 2D instead. This change did not affect our idea for the game's concept, and it allowed us to develop a much cleaner design with more extra features.
Now that our team is finally finished development, the final product of our game differs from our initial concept in many ways. Instead of initially orbiting around a sphere, the projectiles now

ricochet off of the wall. Furthermore, some ideas of power ups and game structure have been removed. Yet, the game still consists of having progressive waves of enemies and making the game as an endurance based game. Overall, while the evolution of the game has rapidly changed from our initial idea, the cohesive and aesthetic final product has resulted in an overall better game.

GAME DESIGN

Gameplay Synopsis

The game's theme revolves around a futuristic ship shooter. The game is fast paced and requires the player to have quick judgement, quick reflexes, and sound tactics. The game scales in difficulty and speed as the player progress wave by wave. Different enemy types will spawn with respect to every progressing wave. These enemy types challenge the player in certain ways in which change the tactics of the player. Overall, the player's main objective is to defeat all enemies on screen using projectiles fired from their ship.

Mechanics/Uniqueness

The game introduces a unique blend of fast paced action, while incorporating sound judgement from the player to tactically fire projectiles in certain directions. The main unique mechanic of the game is the progressive bullets. Bullets persist until it has collided with a game object, whether it be a player or enemy type. The game forces the player to use the environment to connect shots in the best way possible. Firing too many shots in a row will fill the world with projectiles and put the player in a lot of risk.

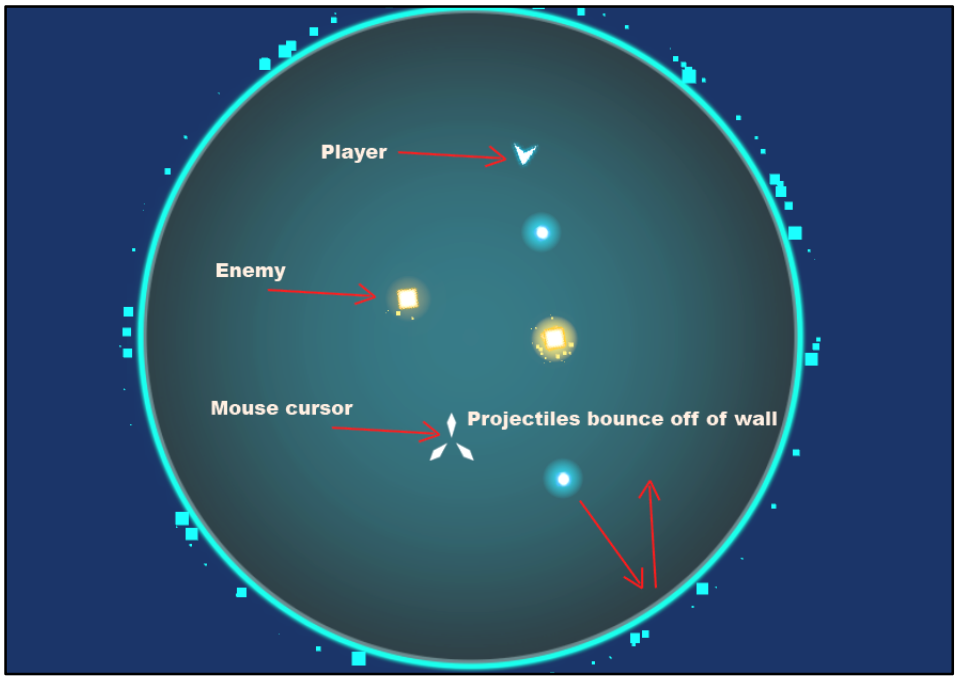
Target Audience

The game does not alienate any age group and is intended for any competitive player who enjoys fast paced action and obtaining high scores. Achiever type players will particularly enjoy this game as the goal is to achieve the highest level possible. There are an unlimited amount of waves which scale in difficulty the higher they get.

Controls

The basis of the game is simple. Players control their avatar ship using the 'W, A, S, D' keys to accelerate up, down, left, right, respectively. The game has fixed directions. In other words, the player does not move with relative to the mouse position. The player is also able to aim and rotate the ship relative to the position of the mouse cursor. The player is able shoot a projectile from their ship using the 'Mouse 1' key. Players may also pause the game using the 'Space' and quit the game using the 'Escape' key. The mechanics of the controls are simple.

Rules



The rules are simple: survive as long as you can and defeat as many waves as you can. These rules are of course bounded by simple mechanics of the game where players have a limited number of lives and are in conflict with enemy AI. The game field is bounded within the circular game level. Instances where the player is challenged occurs in different dimensions. Firstly, projectiles will persist until they have collided with another game object, whether it be the player or an enemy entity. The default projectile fired from the player will bounce off of the outer ring of the game map. The projectile will continue to bounce once it has made a collision with a game object. Another aspect that challenges the player are the enemies themselves. Firstly, if the player makes contact with an enemy ship, the player loses a life. If the player makes contact with an enemy projectile, the player also loses a life.

As players progress through the game, the game will become more difficult. Levels are separated by waves. In each wave, different enemy types will spawn adding various challenges. One main concept of the game is based on an endurance objective, whereby player strive to last as long as possible. Since having to add potentially infinite amount of waves, we intend to structure game progression in a manner that adds uniqueness to each wave, while avoiding to manually build a tremendous amount of waves. The player will progress through a specific amount of waves where each wave is directed with specific number of enemies and different enemy types. Once the player passes the last wave (at this point the player will have encountered all enemy types), waves will be procedurally generated based on the level given. In procedural generation, the enemy composition will be randomly generated, along with the number of enemies spawned. As the player progresses, however, the algorithm in which generate the enemy waves will be influenced the by the current level.

Game World

One of the most dramatic changes to our planned game us the game world itself. Through working on nuances of the game, we decided to change our game world from 3-dimensional to a 2-dimensional background. As a whole, a 2-dimensional game was more feasible in the given time frame. We were surprised to find that the game play not only looked, but also felt a lot better to play. This is likely due to the fact that the simplicity of the two dimensional environment matched our game's chic and simple art style. Overall, the basic design of the game world is complete, as it is a crisp circle that cannot be crossed by any game objects. The original concept in 3D was for objects to orbit the world infinitely until colliding with another object. To carry on with this concept in 2D, objects simply reflect back into the arena when they strike the edge of the circle.

Core Design

The core design of our game is comprised of the elements that must be completed in order for the game to be playable and enjoyable. The core game satisfies all eight of the "Formal Elements of Games" at the most fundamental level. From the time that we started our game to our current state of development, these core elements have evolved based on what we learned and what we plan to do.

The following elements of player, objective, procedure, resources, and conflict have remained constant since the beginning of development. The game follows the interaction pattern of a single player versus the game. The player controls the main character, a space ship, which can fly around in the game world. There are three core procedures for the main character, it can propel itself around the screen using the direction keys, spin using the mouse, and fire a bullet using the left mouse button. Using these procedures, the core objective of the game is for the player to destroy waves of invading enemy characters. Enemies will spawn in waves of increasing difficulty; if the enemy hits the main character with it weapon, the player will lose health point resources and eventually be destroyed. This is the core conflict of the game, while the player tries to destroy all of the enemies, enemies will be actively working to destroy the player's character before he/she can achieve his/her goal.

Once the game began development, some of the core elements had to be revisited, namely the boundaries. For our initial game concept, the game is set on a three dimensional sphere, with the boundary being restricted to the sphere's surface. The initial rule was that objects orbiting the sphere will last forever until striking another object. In practice, this setting turned out to be both overly time consuming, and less polished than a two dimensional alternative. Our new setting is inside of a circular arena with a brightly coloured edge. The boundary of the game is the inside of this edge, no game object can pass it. The rules of the game are simple: if an object strikes the edge it is reflected back into the arena, if the player's projectile strikes an enemy the enemy disappears, and if a projectile strikes the player he/she loses health points. The outcome of the game is inevitably that the player is overcome with enemies and loses all of his/her health points, ending the game.

Character Design

In terms of character design and types, the game play evolved significantly since the start of the game making process. Our team made a firm decision to change the character type from a 3-dimensional shape to a 2-dimensional type; this was done in order to make gameplay more easily adaptable to mobile devices and corresponded well with the given time frame. The main character is designed as a minimalistic white arrowhead, with a subtle blue glow emerging from its edges.

The enemies changed from large battle ships to brightly glowing squares that mutate their edges slightly while in orbit; as a note of detail the enemies appear to sparkle and leave a trail of luminescence. Currently our team has designed 4 unique type of enemies. The first of which glows yellow in colour, and slowly stalks the players as he/she fly's around the planet. This enemy deals damage by physically coming in contact with the player's character, and keeps the player moving at all times. This character is easy to deal with at first, but as more similar enemies are added they become much more challenging for the player to kill and avoid.

The second type of enemy poses more of a threat. This enemy glows red in colour, and also slowly follows the player. However, instead of crowding the player completely, this enemy will not come within a certain radius of the player's ship. This is because instead of dealing damage through collision, this second type of enemy has bullets of its own. This enemy will follow the player until it is within firing range, and will then fire small red bullets at the player, which will bounce around the arena until that enemy is destroyed.

Another enemy type is the purple enemy. These enemies will strafe around the playing field. On a given interval, the enemy will send out an expanding "freeze ring". If the player comes in contact with the freeze ring, the ship's thrusters will be frozen for several seconds. This will prevent the player from controlling the ship, but it may continue to drift if it was previously in motion.

The last enemy type is the green enemy. These enemies are smaller in size, and they move much faster than all the enemy types. The goal of these enemy is to rush you. These enemy will follow you as quickly as possible. This forces the player to move around rather than being stationary.

Game Characters

The characters in our game were never meant to have distinct personalities and backstories. Our goal was to create a game that plays similar to an arcade game, engaging the player by being fast paced and challenging, rather than trying to form an emotional connection. However, as the game came together, we learned that the characters seem to take on certain personalities despite their lack of backstory. We learned that through their look and movement, certain enemies seem to come off as angry, while others seem smart and cunning.

Media List

Sprites:

We currently have one sprite to animate the player. Since the game is a two dimensional aerial view of the arena, one sprite can capture all angles of the main character. The player can also fire bullets, which are small blue sprites that can bounce around the screen.
Like the player, the enemies of the game can also be captured from all angles with one sprite. We currently have 2 different enemy sprites, one for each type of enemy. A small red dot sprite is used as one of the enemy's bullets, which again can bounce around the screen.

Animations:

Since the game takes on an aerial perspective, there are no animations needed for player or enemy movement. However, our team implemented an aesthetic animation of flowing coloured squares that emanate from objects in the game. This animation is a subtle detail that greatly enhances the visual appeal of the game as a whole.

Sounds:

The game currently has an electronic theme song that plays during the game. In addition, we have implemented sounds effects for when the player fires a bullet, an enemy is destroyed, and for a new level. These sound bytes were each carefully selected from free online resources.

Story

A linear story progression is not a focal point in our game design. Instead, we chose to try to capture the frantic intensity of an arcade style game. We believe that the appeal of the game lies in the adrenaline of the ten second intervals, rather than the emotional connection with the player.

DEVELOPMENT

Development Issues and Steps

In the early development stage of our game, we planned to have a power-up system which works to benefit the player with extra abilities such as speed, bullets, fired power and endurance. However, through design and play test, we found that the power-ups were not functioning as we would like them to be. We eventually decided to forfeit the power-up system and instead focus more on the enemy types to keep the game simple.

Another issue for us is when the player makes contact with their own projectiles, the game deducts 2 lives instead of 1 life. We were unable to discover why this is the case without having the rewrite entire sections of code or to change some of the dynamics of the game.

One barrier we encountered while programming the game include calling functions from other scripts in C#. We came across this problem while trying to implement our running hit-points user interface. This was a problem for us because we were not able to call updatePoints from the changePoints script. In the end, we removed the score-based system and implemented a level based game instead, it is a better reflection of player progression for our game.
Another barrier we had was to prevent the ship from moving outside our spherical world. We tried to implement it using the circle collider in Unity. This turned out to be a problem because the circle collider component interferes with the bullet bounce off movement. When the circle collider is added to the sphere, bullets stick to the walls of the sphere instead of appearing at the other end of it. However, we were able to fix this bug. One of the reasons for this was that our character controller was using transforms instead of forces. When the entity moves with a translate, the physics is calculated after the move. Thus, when the player was translating out of circle, the colliders were not calculated. The fix of the projectile were also changed by changing some dynamics of the physics material of the projectiles.

Our initial approach for the theme of the game is to incorporate a mix of both 2D and 3D; whereby, the sphere or game field will be rendered as a 3D object, and the player and enemy sprites will be rendered in 2D. We changed our approach during early development of writing the game. Instead of incorporating a 3D object with 2D sprites, we switched the game into a full 2D environment. The switch was made because we realized a full 2D environment would be more ideal for a simple single player game. The 2D development environment is also a more realistic choice for our given time frame to develop something that is interesting and user-friendly.

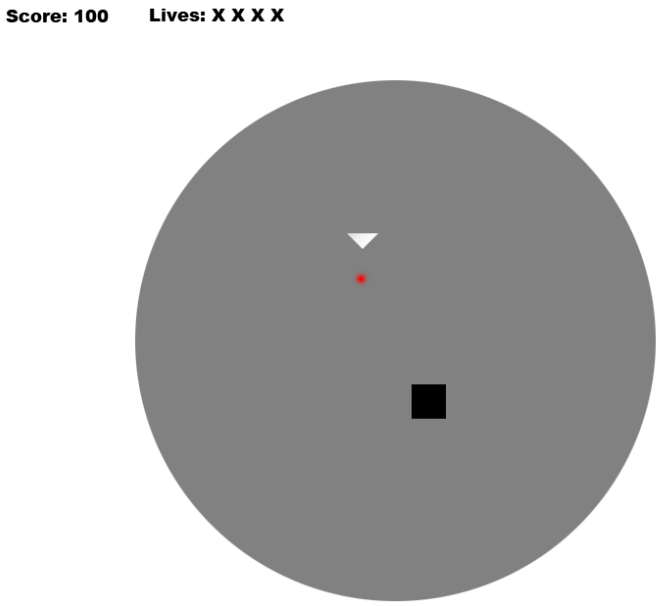
In the game proposal we mentioned that our approach for making 2D sprites would be to grab simple assets from the internet and use Photoshop, during development process we decided not to use asset shop sprites but instead incorporate our own sprites using Photoshop to give the game a more novel feel.

As game development progressed, problems in design revealed by testing came to surface. One of the problems our design had was that with the 8-bit game design concept, it is hard to incorporate the art of our power-up items and power-down items. For example, power up items such as bullet endurance, fire range was harder to display with our art style. Power down items that causes unpredictable weather conditions (eg. tornados, strong wind, overcast sky) were also hard to draw and implement.

Our approach to designing the projectile movement when it hits the boundaries of the sphere is to record the position it hit, find the position 180 degrees across on the other side of the sphere, and calculate the diameter of the sphere divided by the speed the projectile is traveling at to make it appear on the other side at a reasonable timing.

Prototyping

A crucial step in game development is implementing effective prototypes to flesh out game design decisions, pacing, controls, and several other critical components to a game. As much as we hoped to have extensive game prototyping, more could have been done. Overall, however, the end approach in effectively discovering game design issues was settled through some play testing.



Drawing it Out

Designing the game level and game world was designed by drawing the level on the computer using photoshop. Simple shapes were drawn out to represent the game world. Getting a feel of how the game would look was accomplished by polishing some the shapes and features of the

assets. The idea was to keep the game as basic as possible, while having some charm to it. As such, the game objects were rendered as triangles and squares. This allowed the game to have a arcade like feel to the game.

First Approach

Since the basis of the game and was relatively fast paced, using paper and dice, or other methods were not necessary. A base game had to be fleshed out in order to get the feel for the game. As such, a base game of a ship flying around a circle was designed. Our first approach was to display the early game state to other individuals to gain the a general perspective on the feel of the game. At this point, the game was in very early stages whereby many of the features were not implemented. For example, the scoring system had not been completely implemented, the wave system was still under development, and there an abundance of features left out in this initial stage. First impressions from other individual were relatively positive and had appreciated the fast paced nature of the game's mechanic.

Play Testing

The game was tested by several users to get a feel for how the game worked. Something that was notable from all the users was the difficulty of the game. Users who were not accustomed to the game type had a difficult time initially. One of the main goals of the game was to ensure the game was balanced. At first, the game sent out walls of enemies. This made the game easy in some sense since players were always connecting shots on enemies. However, as the player player progressed after the 8th level, the difficulty ramped up too much. The red enemies' projectiles were too fast and too reoccurring. Moreover, the red enemy projectiles would ricochet off of everything increasing in velocity. This made it extremely hectic and chaotic if there were too many on screen. As a result, we reduced the amount on screen and saved them for later levels. Another comment made on the game while testing was the control scheme. Despite the controls being straightforward, some had some time getting used to. For example, in our game, the player does not move with respect to the mouse position. The W, A, S, and D keys move the player up, down, left, right respectively. Half of the users did not mind this control decision while others were used to moving forward and back relative to the mouse position. We felt that this was not a major issue and we wanted to stick with the original control scheme in having a fixed direction. Also while play testing, we discovered some glitches in the game. Sometimes if the player hugged the wall too much, they would clip outside of the world. This, however happened very rarely. Finally, another glitch was that when players would die and have no more lives, their player avatar would respawn anyways. Overall, the initial impression of the game was very positive and many enjoyed the novel idea of progressive projectiles. Moreover, many people appreciated the simplicity of the game. Because of this, we strived to make the game as simple and fun as possible.

Lessons Learned

The process in developing a videogame is demanding and lengthy. Throughout the development of the game, there were several stages involved in attempting to produce a functional and novel game. One of the biggest lessons we feel we learned is the appreciation of how much work is needed to have a fun game. There is an enormous amount of things that need to be considered and extensive play testing and prototyping is required to make the game playable. Continuing with this thought, it is critical that the development of the game remains consistent throughout the course and that amends are constantly being made to improve the game. There were instances where some aspects of the game were put off and left more last minute. Because of this, the game was not completely polished to perhaps its maximum potential. Moreover, it is important to acknowledge that it is easy to spend an extensive amount of time to add onto the game. The best approach we feel when developing this project is to have something simple and not to add too much to the game. The time spent attempting to add to the game could have been used to add polish to the game and making it run smoother. A balance is needed to be successful when developing the game. Overall, the biggest lesson we learned is the ability to work together as a team and attempting to organize the work. This lesson in teamwork among one of the most fundamental lessons that we can obtain. Especially in the field of computer science, there are numerous instances where there are multiple people working on the same project. It is crucial that the work is distributed properly in order to have the game be developed in a consist manner. To conclude, the project was a terrific experience. After having developed the game, we feel that our skills using Unity has vastly improved. Moreover, some other things that we learned after this project is the ability to code, debug, and the ability to discover creative ways to code and problem solve aspects of a novel video game.