Luck of the Draw

Vision/Develop Statement:

Luck of the Draw is a game about surviving in a brand new world in each playthrough. Initially, the player will know nothing about the world that they are dropped into. In order to survive, they must explore the hostile environment around them and see what resources they can find an experiment with. The further the player explores from the center of the world, the more dangerous the enemies become, and the more powerful the upgrades they find. The goal is to put players in a newly discovered world with each playthrough, with new surroundings to explore to prepare for the final fight against the boss at the center of the world.

Design Process Document:

Pitch: *Luck of the Draw* is a game where the player must explore a foreign environment in search of resources to both sustain and upgrade their character in preparation for a final fight.

Overview:

Luck of the Draw is a game of exploration and experimentation. Players are given control of an armed character in a top-down environment. The player must monitor his character's speed, health, and hunger as different items and npcs may affect them differently. Players need to find the right combination of materials and upgrades to fit their play style. When they feel that they are ready, the players can choose to challenge the boss and potentially achieve victory.

Rules:

1. The player controls a character with a 360 degree viewing arc and directional movement.
2. The player is limited in survival by a health and hunger bar, the latter of which depletes over time. When the hunger bar reaches zero the player's health will naturally drain over time and when the player's health bar reaches zero the player will die.
3. The player can avoid death through starvation by the consumption of fruits produced by the world.
4. The player can shoot projectiles out of their gun in the direction that are looking and produces the projectiles at the end of the guns barrel.
5. The player's gun is made of three modular components that can contain different parts acquired by the player through the physical world.
6. Only one part can be placed in each component of the gun and a varying effect will exist on the gun while that part remains in a component.
7. Parts can be traded out and replaced by other parts found within the world.
8. The loss condition of the game is achieved once the player has reached zero health
9. The player loses health by interaction with enemies from running into them and collision to being hit by enemy projectiles
10. The win condition is achieved once the player has defeated the final boss in the inner arena

Process of Designing:

The process of designing can be divided into a series of different sections and steps.

First, we pooled ideas together. We talked about the feeling we want to convey with whatever we make. We talked about game styles and preferences for creation. After roughly a day of this, we settled on the direction of survival/roguelike.

Second, we talked about the visual direction and feel of the game. This is where we had to compromise. Some of us were indifferent, others were adamentally pushing for a certain direction. This lack of solid foundation unfortunately followed us into the prototyping phase.

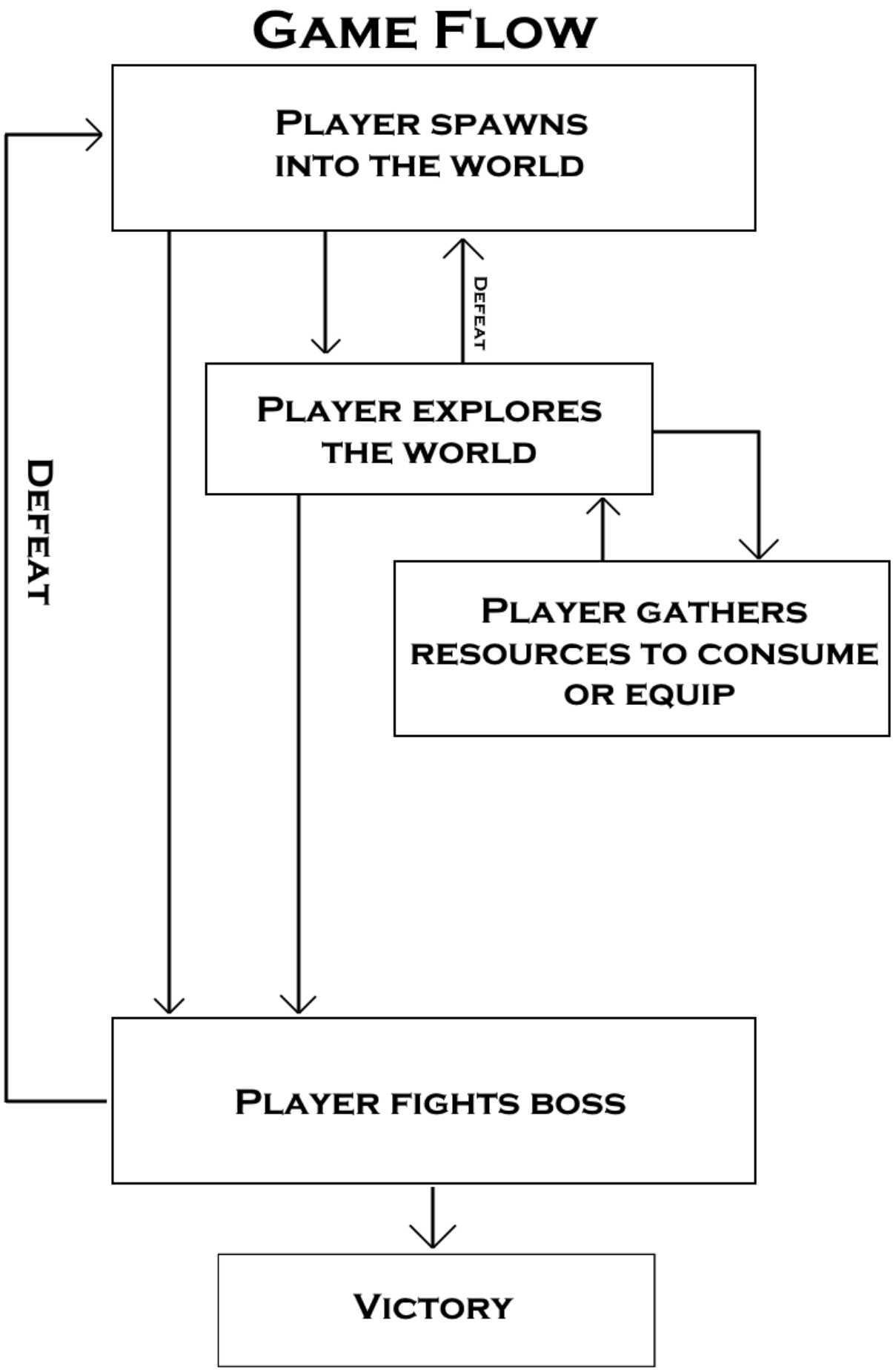
Third - the prototyping phase. This was made up of a series of sporadic design half games. Some of us thought it was more important to convey the feeling of our potential game rather than the mechanic. Others approached this from the opposite direction. Because of this, our prototyping phase took longer than expected. This also created a slight divide between the advocates of roguelike gameplay and the advocates of survival. Eventually we compromised with our proof of concept.

Fourth - The proof of concept. Our was proof of concept began as an amalgamation of different disconnected ideas. We wanted a hunger system, so we got one. We wanted random stat enemies, we added them. We wanted modular weapons, yeah… it's there. This process also took a long time for us because of various disagreements and compromises. But suddenly, things started to look up. With the addition of our artist's sprites and the final unification of our individual scenes, a game started to take form.

Fifth - Where we are now and potentially where we will be - The survival side of our game, unfortunately, is beginning to fade away. We had to settle for something uniform and accomplishable within the deadlines provided. Due to our "grey area" during the beginning of development, we had to make up our minds fast (or let whatever happened happen). Our game is an arcade game. It is bright, warm, and inviting in visual style. It has the

potential to be competitive. In the spectrum of rogue-like, fast paced, rapid decision making, and slow burning, anxious, survival, we are arcade through and through. Now that we've established our overall "feeling" and "semi casual" direction, development and design should flow easy.

Jack Guo, Oscar Ngan, Ian Ouellette, Nick Neissa, Ben Fried



Prioritized Feature List:

Audio- Since we've determined the general gameplay loop for our game, now is the time to start adding audio, both to improve the game's feedback and its general appeal.

Implementing Setpieces- Playtesting throughout our proof of concept phase determined that our world as of now is generally uninteresting, and there's not much reason to explore than to just pick up three upgrades before fighting the boss. To fix this, we're planning on placing structures containing enemies and upgrades throughout the world on generation, which should give players recognizable landmarks and short-term objectives, since the setpieces will be their main source of upgrades.

Expanding the world- For the proof of concept phase we limited ourselves to only three biomes and the boss area. Now that we have world generation working, we can add more biome hexes to the world. Additionally, biomes further from the center will be more difficult, but have better rewards. Our goal is to create two "rings" of biomes hexes surrounding the center tile. These outer biomes will be more difficult terrain to survive in as it will have a higher concentration of enemies as well as stronger ones that are more lethal to the player. However, as stated before, these outer biomes will also provide the player with greater rewards and modifications that they could not acquire in earlier zones.

Wasteland - Little fruit, wide open spaces

Tundra - Little fruit, less open spaces and more difficult terrain elements

Ruin City - Little fruit, minimal open spaces

Overgrown City - Many fruit, minimal open spaces

Swamp - Many fruit, wide open spaces

Lake - Many fruit, impassable open space in the middle of the biome with surrounding resources

Additional Enemy Types- Currently our enemies are rather unthreatening and easy to avoid. To fix this, we're planning on increasing the variety of enemy types and making them more dangerous to the player. While the goal is that the player will actually have to fight to defend themselves, enemies will also be changed to drop consumables so the player has an incentive to actually seek out enemies to fight.

Implement a Menu- We have a working build of the game scene itself, but we still need to add scenes for the starting menu and for after the player wins.

Creating a Story- While our game already has some bits of narrative just from the player's goal and the character/enemy design, the game still feels a bit pointless from a story perspective. Having an actual story to the game would make it feel more complete, as well as provide some direction for the player at the beginning.

Tech Document:

World Generation

Focus

The Focus of our world generation was to ensure enough diversity to make the game visually different upon introduction to each new playthrough but still retain aesthetics and features that can be memorable and noticeable to the player in later runs of the game. The introduction of new assets as well as the recycling of old ones for different purposes and effects will create an environment that the player can learn but at the same time be uncertain of when playing in a new world.

Approach

We wanted to take view our world map as a culmination of different biomes that were conjoined to form one massive arena-style world in the form of a hexagon. Each individual biome is a tile in the form of a hexagon that is conjoined together at the very beginning of the scene before the actual generation takes place. Each biome has a number of different assets integrated such as the resources present in the trees, the specific kinds of trees, the impassable terrain objects, as well as landmarks. Each biome contains the initial prefab for each resource and object that will be found within each tile. Once the game has started the biome then runs a series of statements which randomize values which will determine what kind of environment this biome is as well as the specific values for the number and location of all resources and environmental features that will pertain to this

specific biome. The biomes also all contain a generator which handles the production and diversification of all species of edible fruits that the player will encounter. Upon execution of the game, a list of different fruit species is created and once tree prefabs are instantiated from the biome they take from that list to fill their branches will consumables for the player.

The final feature with world generation that we are looking to create is the introduction of a ring system within these biomes that ties to it's difficulty. There will be to layers of tiles with the innermost layer ranging in easier difficulty than the outermost layer which will provide a more challenging yet more rewarding experience for the player.

Enemy AI

Focus

In order to create an interesting experience as the player moves through the world, obstacles need to be placed in their way. While hunger acts as a way to keep them moving, the need to 'survive' on its own is not motivation enough to make a player want to explore. Thus, the need to create interesting enemies—in terms of behavior, at the very least—came up. This does include the boss, of course; how it acts also sets up the 'final challenge' for the player to overcome.

Approach

Each enemy 'archetype' is spawned with a different AI pattern; for example, some are docile, and simply wander about on a timer (picking a random direction and moving thus), while others may charge at the player once they are in range. This change in AI is done by first defining a parameter within the enemy and changing how it moves in turn, which itself is based on a switch statement that takes that parameter as an input.

Enemies that do not move (e.g. shooters or spawners) simply instantiate and object (be it a bullet or another enemy) and send that at the player.

A few enemy archetypes are triggered by being within a certain radius of the player. To do so, each enemy under this category constantly compares the distance between the player's position and its own, and if that distance with less than or equal to a certain range, will act accordingly.

The boss is a unique instance in that it changes phases upon hitting certain HP thresholds. This is done by using a counter to check if a boss has hit any one of those thresholds, then changing phases accordingly. The boss currently cycles between 3 existing enemy AI patterns (spawner, shooter, and rush), but the system is modular enough to allow for more options to be slotted in or out should they prove to be too mundane or excessive for our liking.

Player Controls and Interface

Focus

How the player moves and sees the world is essential to how the game is played; without a simple control scheme and UI, the game's already confusing premise can easily become lost in the chaos. The goal in turn is simple: conveying what the player needs to do and what information they need without overloading them from the very beginning.

Approach

A minimalist approach was adopted in terms of controls; using the WASD keys to move and the mouse to aim and shoot means that the player can quickly intuit how to react, as such a control scheme is common enough to the point where it does not need much explanation. The gun moves around to point at the mouse's current location as well, which allows the player to get a sense for how it acts even before shooting once.

In terms of interface, we established a UI with 4 core aspects: the HP bar, the hunger bar, the speed bar, and the upgrade slots. The first two are self-explanatory; HP increases or decreases when the player is hit and determines if they are dead or not, while the hunger bar determines if they

are in danger of losing health from starvation (and is marked by a drumstick to indicate ‘food). The speed bar is gradually revealed as the player consumes food that increases that value, however, and is located directly under the HP bar for ease of viewing. Most important of all, though, are the upgrade slots; here the player is able to not only see what materials they have placed into their gun in order to change its properties, but also what they do when placed into a certain slot. This will be elaborated upon further in the following section.

Item Modularity and Upgrades

Focus

Simply giving the player a gun to shoot with is relatively straightforward, but the gun on its own does not make for an interesting experience, particularly when the player is not able to perform any fancy actions or movements. This, in turn, called for a new mechanic: the ability to ‘change’ the player’s weapon as they saw fit, provided they had the materials to do so.

Approach

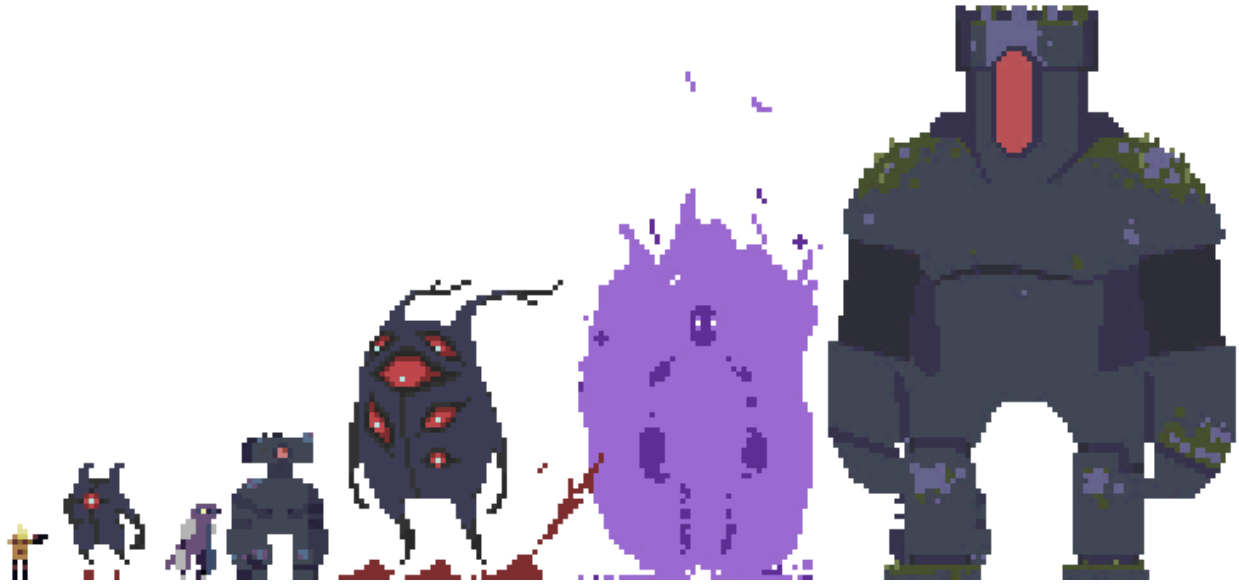
This is done by means of an InventoryManager script, which takes the Resource object being used as a material and ‘hides’ the instance of the object being equipped. From there, we make use of a PlayerShooter script to parse the integer value on the Resource that corresponds to a specific effect and apply that to the gun (or what it shoots) in turn. Each object can affect either the gun (in terms how it shoots), the bullets (and how they act or move), or what the bullets themselves do to what they come in contact with.

Vignette Shader Plugin - This is a unity asset that helps us modify post-processing shader effects. We intend to use this with the UI in order to delineate certain player imminent effects such as hunger.

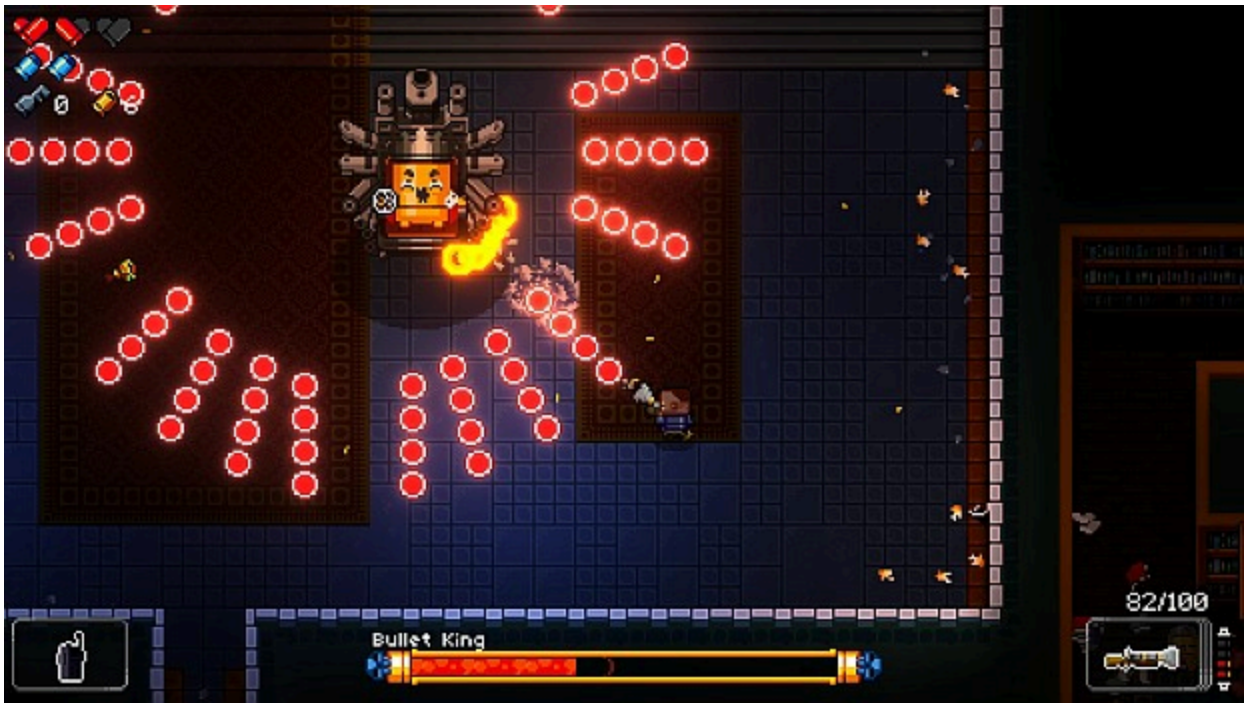
Visual Style Pillars:

Our initial pillars for the visual style are *Don't Starve* and *Binding of Isaac*, as we were inspired by their world-building and atmospheric elements. Although these the former of the two games has a rather dark and scratchy style, we went with a slightly lighter tone as we added more action elements.

As we ventured a bit farther away from horror and survival, we looked at other shooter/roguelike games such as *Nuclear Throne*, *Risk of Rain* and *Enter the Gungeon* for reference. Although the visual style for these games are all made of pixel-graphics, we took inspiration from the unique and interesting designs of the enemies, guns, and characters.



The final visual style has been hard to pin down, as the game has been struggling between the identity of being a survival game and being a roguelike shooter. However, we decided that we can embrace both genres, as *Don't Starve*'s various DLCs have given us great reference for creating and populating different biomes and environments, while bosses, enemies and gun effects can be taken from games like *Nuclear Throne* and *Gungeon*.


Bullet King
82/100

Audio Style Pillars:

Ambience:
The background of *Luck of the Draw* is meant to give off a sense of a foreign world. We plan to use slow to moderate rhythms and melodies to let the player feel in foreign world and convey a sense of unknown.

SFX:
The sound effects for *Luck of the Draw* will be in contrast to background music deliberately to give the player feedback with each action and each object they encounter. Since the game is about surviving in a foreign environment with each playthrough the noise of how each object affects the player internally as well as externally will be noticeable and recognizable.

Jack Guo, Oscar Ngan, Ian Ouellette, Nick Neissa, Ben Fried

Asset Scope Plan:

The assets we want to have finished by the end of this course includes all sprites and textures for the basic game, as well as animations for the player. Moving from the proof of concept, we decided to rework some of the story and thematic elements, and thus a basic rework of many of the sprites (such as biome textures, trees, etc.) is needed.

Sprites	Rework Needed (?)	
Player		x1
Gun		x1
Enemies	Yes	x6
Boss	Yes	x1
Trees	Yes	x3
Fruit	Yes	x3
Minerals	Yes	x5
Bullets		x2
Biome Setpieces		xTBD

Biome Textures	Yes	x6
Gauntlet Texture		x1
Gauntlet Setpieces		xTBD
Animation		
Player Movement		x1
Player Firing Weapon		x1
Player Taking Damage		x1
UI		
Modularity UI		x15
Title Screen		x1
Lose Screen		x1
Victory Screen		x1
Opening Cutscene		x1
Overscope Assets		
Enemy Movement		x6
Boss Movement		x1
Animated Setpieces		xTBD

Production Schedule

Jack Guo, Oscar Ngan, Ian Ouellette, Nick Neissa, Ben Fried

Production Worklog