Random Access Memories

Hanieh Arabzadehghahyazi
Yifei Sun
Chunyu Peng
Yaorong Xie
Yang Jiang

October 20, 2021

Contents

1 Overview **3**
1.1 Premise 3
1.2 Inspiration 3
1.3 Game Play Overview 3
1.3.1 Goals 4
1.3.2 Flow Chart 4
1.4 Engineering 5
1.5 Key Features 5
1.6 Target Audience 5
1.7 Similar Games 6

2 Characters **7**
2.1 The Hero 7
2.2 Swordsman Enemy 7
2.3 Sniper Enemy 8
2.4 Jetpack Rifleman Enemy 8
2.5 Mech Boss 8

3 Game Play **10**
3.1 Game Loop 10
3.2 Game Loop Example 10
3.3 Core Mechanic 11
3.4 Control Scheme 11
3.5 Movement 11
3.6 Player 12
3.7 Enemies 12
3.8 Abilities 12
3.8.1 Player 12

3.8.2 Enemies 12
3.8.3 Mech Boss 12
3.9 Score 13
3.10 Win and Lose Conditions 13
3.11 Map 13
3.12 Spawning Points 13
3.13 Blood Effect 13
3.14 Camera 14
3.15 Animation 14
3.16 Sound and Music 14
3.16.1 Background music 14
3.16.2 Sound effect 14
3.17 Saving and Loading 14

4 Game Structure **14**
4.1 Game Story 14
4.1.1 Introduction 14
4.1.2 Main Game Loop 15
4.1.3 Ending 15
4.1.4 Narrative Example 16

5 Art Direction **16**

6 User Interface **17**

7 Level Design **17**
7.1 Level 0 - Introduction 17
7.2 Level 1 - Spawning Waves and the First Boss 18
7.3 Level 2 19
7.4 Level 3 19

8 Technologies **19**

9 Scope **20**

10 Production **21**
10.1 Work Distribution 21
10.2 Weekly Plan 22

11 Appendix **22**
11.1 Production document 22

1 Overview

1.1 Premise

Random Access Memories is a single-player 2D game. The player controls a soldier who wants to discover important data from a laboratory. The soldier needs to escape from and will fight against different types of enemies. Different types of enemies with various weapons and different levels of difficulty will be shown in the game. The player needs to move around the environment and kill all the enemies to reach the next level until it finds the data. The player will use different weapons and learn how to fight against the enemies.

1.2 Inspiration

The game takes inspiration from Binding of Isaac. Binding of Isaac is a top-down dungeon crawler game, presented using two-dimensional sprites, in which the player controls characters as they explore the dungeons located in the basement. The characters differ in speed, amount of health, amount of damage they deal, and other attributes. The game's mechanics and presentation are similar to the dungeons of The Legend of Zelda while incorporating random, procedural-generated levels in the manner of a rogue-like game. On each floor of the dungeon, the player must fight against all monsters in the room before continuing onto the next room.

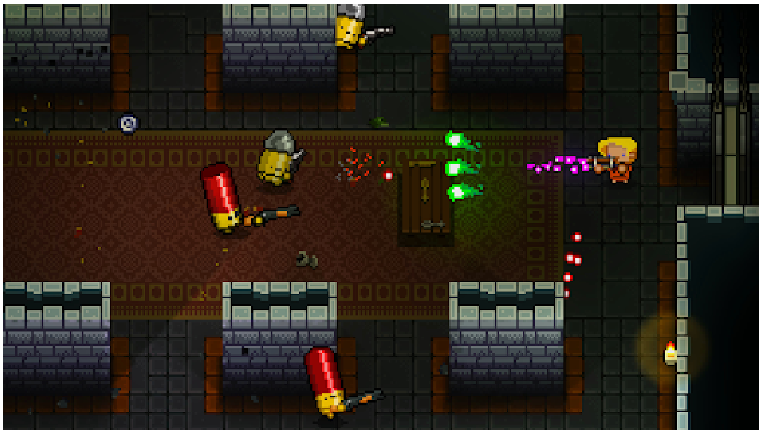


Figure 1: Binding of Isaac

1.3 Game Play Overview

There are 2D shooters with rogue-like elements. For example, enemies vary in speed, strength, attack behavior, etc. The player will also experiences randomly generated levels with each play-through.

1.3.1 Goals

- Highly challenging but also considerate game, where the player struggles to fight but also won't die quickly
- Make a player character that can move around in the environment and shoot the enemies
- Power up the player and increase the chance of winning by providing various game items pickup
- Create sound effects
- Create particle systems
- Create various weapons and shooting projectiles
- Present different levels with different difficulties
- Present various types of enemies and randomized waves of enemies
- Present different bosses on different levels
- Publish the finished game on the Android platform

1.3.2 Flow Chart

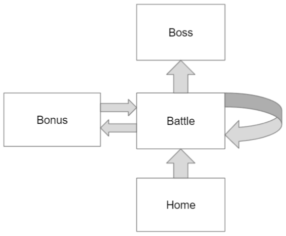


Figure 2: flowchart

We will start from the home scene near the village. The home scene will work as a tutorial, and the player will kill three sample enemies from the home scene.

Then we will go to level1 after defeating these three enemies. The level1 we are having right now is in a laboratory. Nonetheless, we are going to have different background made for later levels, and the environment blocks are going to randomly change so we never play on the same map.

The boss will be summoned after all the enemies are defeated. The boss itself will also summon different enemies after player hits it. We are going to implement different bosses for different levels.

After accomplishing each level, we are going to have chance to go to a bonus room. The player is able to use the points/scores to purchase weapons, bullets, as well as portions.

1.4 Engineering

Windows/Mac will be the platforms. The game will be implemented fully in Unity including creating the player characters, enemies, weapons, playground, boss, etc. The code will be distributed between the team members through Unity Cloud and any changes in the code will be pushed on Unity Cloud constantly during development. The final product will be published on the Android platform.

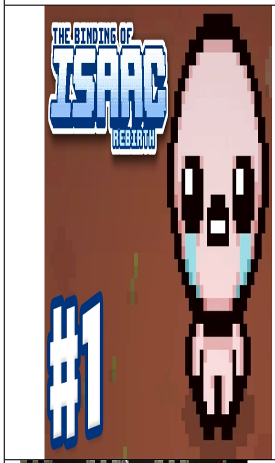
1.5 Key Features

- Randomly generated enemies and bosses in different environments. The player never plays the same game twice
- Various types of enemies with different abilities to attack the player, such as ranged attacks and melee attacks
- Have various ranged attacks, such as shooting bullets, shooting lasers, etc.
- Have various melee attacks, such as punching, jabbing, etc.
- Boss enemies, the most powerful enemies, can summon normal enemies around the player when they come to the scene
- Boss enemies will have different stages to increase the difficulty and playability of the game
- The player can power up itself by picking up different types of game items
- The player can change weapons and choose the appropriate weapon based on the enemy type

1.6 Target Audience

The game is designed for people who love 2d top-down games. Due to its violent nature (guns and bullets), the intended audience ages above 16. It is more aimed at helping people who are stressed out from school, work, or something else to get relaxed. This game will work as a refresher for people who spend most of their day in stressful environments.

1.7 Similar Games

Game	Target Demographic	Other statistics
	• Ages 16 and up	• This game randomly generates environments and enemies. • In this game, player is able to shoot at all directions. • There is a bonus room in this game to acquire items using earned points.
	• Ages 16 and up	
	• Ages 13 and up	• In this game, player goes to different dungeons and defeat different enemies. • There is a boss for each dungeon and killing the boss is the winning condition. • Weapons can be purchase in dungeons and at merchant.

2 Characters

2.1 The Hero

The player character has the most influence in the game. Our character is a soldier who can run very fast and move around the environment. The player can use the provided joystick to control the character's movement. The player can attack different enemies with different available weapons.There will also be different projectiles for difference weapons. Besides, the player can pick up various game items to power up the character.

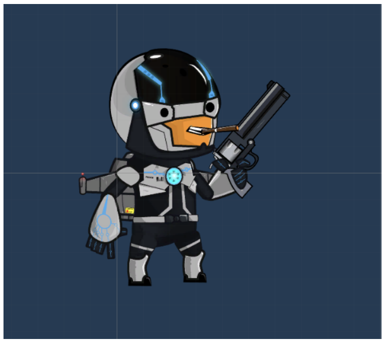


Figure 3: The Hero - Player

2.2 Swordsman Enemy

The Swordsman can swing his sword to do a melee attack on the player but runs comparatively slow. On the other hand, The Swordsman will deal higher damage than ranged enemies.

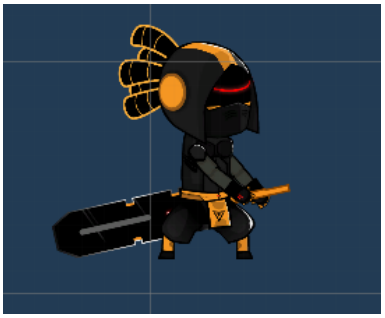


Figure 4: Swordsman - Enemy

2.3 Sniper Enemy

The Sniper is sneaky and hides somewhere on the map. When the player gets close enough, it will snipe the player.

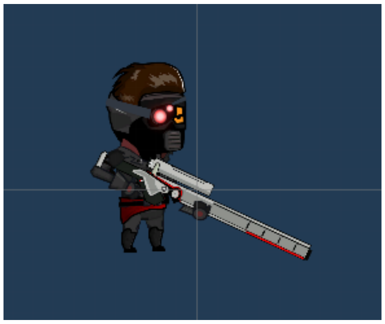


Figure 5: Sniper - Enemy

2.4 Jetpack Rifleman Enemy

The Jetpack Rifleman flies very fast and can shoot the player with a huge amount of damage. The logic for his movement is that he will move faster than the player when he discovers the player and tries to approach the player. Then, it will get slower when the player is within its shooting range.

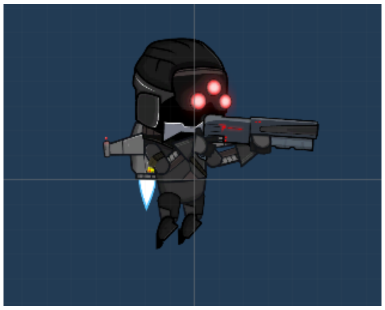


Figure 6: Jetpack Rifleman - Enemy

2.5 Mech Boss

The Mesh Boss has three phases: (1) patrol, (2) chase and melee attack, (3) chase and double melee attack. In each stage, the appearance of the boss will change, and its attack speed and movement speed will also change. The behavior of boss is distinct from other enemies. For other enemies, we do everything in prefab scripts. For the boss, the patrol and chase behavior is controlled by state machine behavior. We are also planing to add a special animated laser attack before the final.

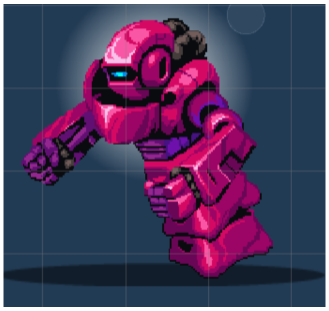


Figure 7: Mech Boss Stage 1 - Boss

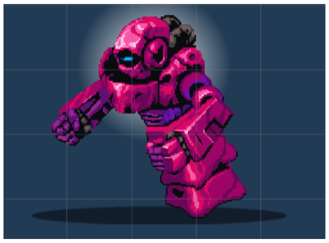


Figure 8: Mech Boss Stage 2 - Boss

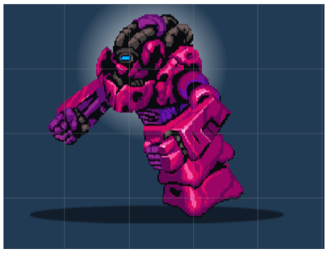


Figure 9: Mech Boss Stage 3 - Boss

3 Game Play

3.1 Game Loop

- Start a level
- Exploring laboratory
- Fighting enemies
- Fighting boss
- End the level
- Continue until reaching the top level
- End game

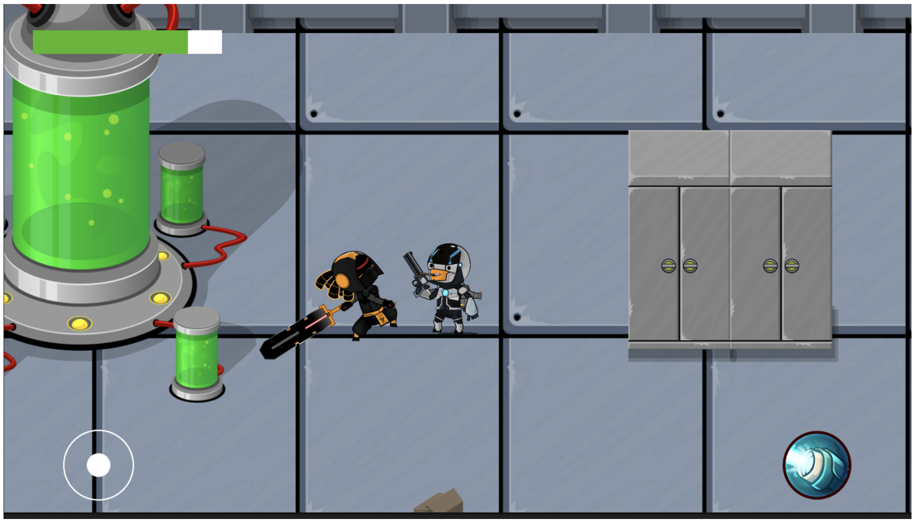


Figure 10: Game Screenshot

3.2 Game Loop Example

The hero is moving around the environment on each level and needs to kill all the enemies and big bosses to get the data he needs. However, once the hero is getting close to enemies, enemies will chase the hero and try to kill the hero. The hero can choose to run to avoid being attacked by the enemies. But he has a gun in hand, so he can also shoot at all the enemies. If the hero is killed by enemies, the game is over, and the player will be asked either to restart the level or go back to the start menu. If the hero kills all enemies and bosses on the current level, he will be directed to the next level. If the hero kills all enemies on all levels, he will get the data from the laboratory and win the game.

3.3 Core Mechanic

The core mechanic is shooting. The player does over and over again in shooting, running, avoiding enemies' attacks, exploring the laboratory, and killing enemies.

3.4 Control Scheme

- The player can move the hero character up, down, left, and right by dragging the joystick on the bottom left corner.
- The player can click on the bottom right button to shoot projectiles.
- We are planning to add charged attack when the player long presses the shooting button.

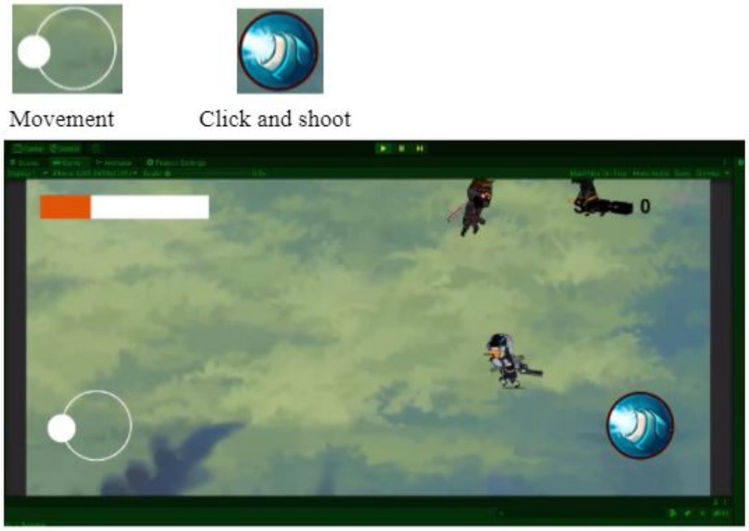


Figure 11: Control Schema

3.5 Movement

- The game follows a standard 2D top-down game's movement schema. The player and enemies can move up, down, left, and right.
- The player's direction will be flipped depending on the joystick input.
- Boundaries are added so that there are certain areas the player cannot pass through.
- Terrains, such as boxes and walls, are added so that the player can be blocked by different objects on the map.

3.6 Player

- The character moves up, down, left and right based on the provided joystick input from the player.
- The character always turns to face the direction that they are moving towards, which means the player can only shoot to the direction the player is facing.

3.7 Enemies

- Enemies can run and fly in up, down, left and right directions and attack.
- Enemies chase the hero and start attacking when they get to a specific distance from the player.
- Different enemies have different attack behaviors and deal different amounts of damages.

3.8 Abilities

3.8.1 Player

- Walking
- Running
- Shooting
- Health Pickup
- Switching Weapons
- Shooting Different Projectiles

3.8.2 Enemies

- Running
- Flying
- Ranged Attacking
- Melee Attacking

3.8.3 Mech Boss

- Patrol
- Chase
- Melee Attacking
- Double Melee Attacking

3.9 Score

After defeating enemies, the player will get certain scores based on enemy types. When the player completes a level, he will have a chance to enter a bonus room before moving to the next level. In the bonus room, the player can use scores gained from killing enemies to buy different weapons with various projectiles and damages.

3.10 Win and Lose Conditions

- Win: the player kills all enemies and bosses on all levels.
- Lose: the player's health goes to zero.

3.11 Map

- The background will be fixed for each different level, but blocks will be randomly generated in the map.
- We are planning to add a mini map to show player's current position in the whole map.
- We are planning to add scripts to destroy blocks on the map when player shoot the blocks.
- We are thinking of adding different effects when player touches certain blocks in the environment. For instance, if the player touches water, player's speed will go down. If the player touches fire blocks, player's health will decrease gradually.

3.12 Spawning Points

- We have created spawning points for each level.
- Each spawning point will generate a random wave of enemies.
- Enemies will come in waves, and different numbers of enemies will be generated for each wave.
- Boss will be summoned after all the other enemies are killed.

3.13 Blood Effect

- We are adding blood effect when player and enemies take damage.
- There will be black blood left on the floor if an enemy dies.

3.14 Camera

- The main camera used FixedUpdate and Lerp function to make the camera follow smoother.
- The main camera will not show areas that are out of the background image by using the clamped function.

3.15 Animation

- The animation for player and enemies are pre-recorded as PNG sequences. These PNG files are generated by spriter pro.
- Later in the semester, We are going to use spriter pro to change weapons as well as record our own animation.

3.16 Sound and Music

3.16.1 Background music

On each level, the background music can be played automatically based on the level's environment. But the final level background music is unique.

3.16.2 Sound effect

When the player shoots, it will have a shooting sound. If the enemy gets hit, there will be a hitting sound as well. If the player is hit by enemies, a different hitting sound will remind the player that the character gets hit. If the player's health becomes very low, a tense and fast-paced sound will be repeated to remind the player that the character is about to die. When the player is killed by enemies, there will be a sad game-over sound. If the player kills all the enemies, there will be a bingo sound before getting to the next level.

3.17 Saving and Loading

Currently, we have not implemented the loading and saving feature, but we will try our best to implement it before the final.

4 Game Structure

4.1 Game Story

4.1.1 Introduction

One day, all the people in the village were kidnapped by a monster except the player ACE since ACE was not in the village. After ACE came back, the god told him what had happened while he was not at home. ACE wanted to save all the villagers immediately but he did not know where his villagers were kidnapped and did not have the ability to fight against enemies. However, he had a dream in the night, and people in the dream told him that if he wanted to save all the villagers, he had to climb to the top of the mountain that was 3,000 feet in the village. ACE spent 1 month reaching the top of the mountain eventually. On the top of the mountain, there was a 200-year-old person who was waiting for ACE for 1 month. He transmitted all his skills, experiences, and weapons to ACE. He gave ACE a bottle of gatorade energy drink that could help ACE recover his energy immediately and the laboratory map that can help ACE know where enemies were hidden, how many enemies there were, and where the confidential data was located at. ACE was also told that he could not find out where his villagers were kidnapped until he got the confidential data in the laboratory since the confidential data in the laboratory record where his villagers were. The old guy also gave ACE a spaceship that could take ACE to the laboratory. ACE took the spaceship to the laboratory.

4.1.2 Main Game Loop

The laboratory had many levels, and the confidential data was somewhere on the top level. However, ACE could not reach to the top level to get the data since each level had a lot of enemies guarding the entrance and stopping ACE from getting the next level, and there was a big monster in the top level to prevent ACE from stealing the confidential data. ACE had to kill all the enemies on each level before getting to the higher levels in the laboratory. If ACE was killed by enemies, he had to start from the base level again. After ACE succeed in each level, he took the spaceship to the next level in the laboratory. The next level had different enemies, backgrounds, music, sounds and enemies had stronger attacking abilities than the previous level.

4.1.3 Ending

ACE killed all the AI enemies on each level and eventually reached to the top level in the laboratory, but he was stopped by the big monster. He had to kill the monster to get the data. He was fighting against the monster. Since ACE consumed much energy to kill all AI enemies before, ACE was not able to kill the big monster right away. Fortunately, the old person on the mountain gave ACE a gatorade energy drink which could help ACE recover his energy immediately. He drank it, and then he killed the big monster in 10 minutes. ACE found the data, and he found out where his villagers were through the confidential data. ACE took the spaceship to the location where his villagers were kidnapped. Unfortunately, after he arrived at the location where his villagers were, ACE was afraid to get in because the monsters who was guiding the gate were much stronger than ACE and they had more advanced and powerful weapons than ACE's. For example, they were wearing bulletproof vests, and holding a submachine gun in each hand. They were also equipped with swords on the belt. ACE did not get in since he knew he could not kill these enemies at that moment. ACE came back home and climbed to the 3000-foot mountain again. He was looking for the old man who gave him the weapons, but the man was not there. ACE was disappointed, and he climbed down the mountain. Suddenly, the old man showed up and asked ACE to come back to the top. ACE came back to the mountain top. He told the old man that he could not save his villagers now because he did not have the ability to attack them. He asked for help. Fortunately, the old man was willing to teach ACE more advanced skills to fight against stronger monsters, and he also helped ACE order submachine guns, bulletproof vests and swords. After 1 month, ACE was trained to be more powerful and equipped with more advanced weapons. ACE was ready to save his villagers...... New game is being developed...

4.1.4 Narrative Example

God: “Hello, my name is God, who are you?”

ACE: “Hi God, this is ACE.”

God: “Hi ACE, are you living in this village?”

ACE: “Yes, I am.”

God: “Hi ACE, your family and other villagers were kidnapped by a monster. Help them!”

“God disappeared..”

ACE: “Why do you appear in my dream? Who are you?”

Immortal: “I am immortal, I am here to help you save your villagers. If you want to save them, climb to the top of the mountain”

ACE: “Why climbing to the top of the mountain can save my villagers?”

Immortal: “I cannot tell you. . . ”

“ACE woke up. . . . Climb to the top of the mountain”

Superhuman: “Hi ACE, I have been here to wait for you for one month.”

ACE: “Hi, Do you know me?”

Superhuman: “Yes.”

“Superhuman transmitted all his skills, experiences, and weapons to ACE.”

Superhuman: “ACE, I do not know where your villagers are now, but I know the laboratory has the data which records where your villagers are”.

“Superhuman disappeared. . . .”

5 Art Direction

- Our game is set up so that it gives the player a feeling of being transited to a slightly different environment every time they get to a different level. Different environments can help to attract players to continue playing the game.

- Our game is science-fiction themed. Using a laboratory-similar style gives the player a feeling that he is in the laboratory while he is playing. This style also gives the player an interesting and original visual experience.

6 User Interface

Because the topic of our game is science-fiction related, we choose to use a futuristic style of User Interface themes in our game. For example, most of the menus have a black background and blue frames and buttons, which emphasizes the dark theme of our game.

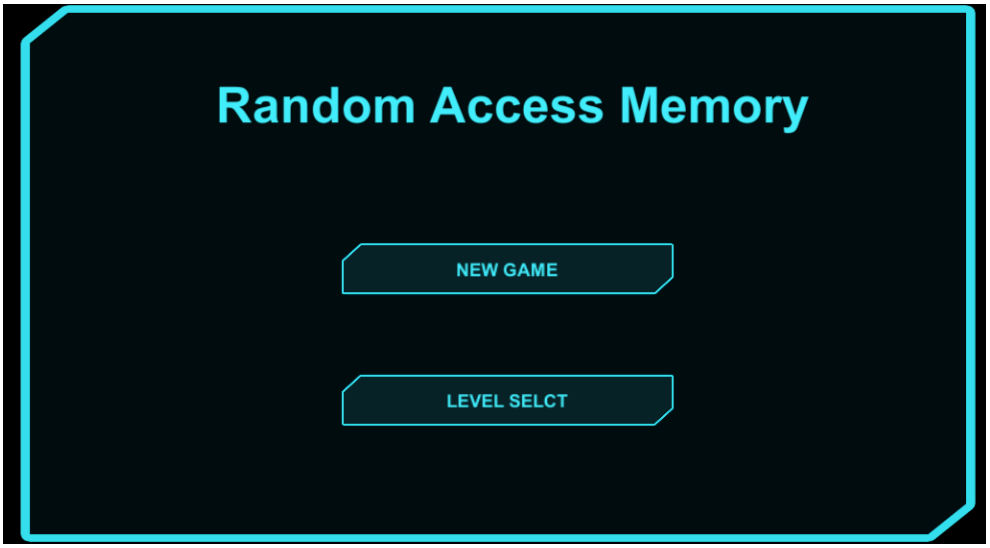


Figure 12: Start Menu

7 Level Design

7.1 Level 0 - Introduction

- Level 0 is an introduction level, where players can learn how to use the joystick to control movement and use the fire button to shoot and kill enemies.

- The player will also learn how to avoid attacks from the enemies and pick up health on the map.

- There are three types of enemies on this level: Swordsman, Sniper, and Jetpack Rifleman.

- The Swordsman can swing his sword to do a melee attack on the player, but run comparatively slow.

- The Sniper hides somewhere on the map, and when the player gets close enough, it will shoot the player.

- The Jetpack Rifleman runs fast and can shoot the player with a huge amount of damage. This will be the first scene the controller encounters after the people in village are killed.

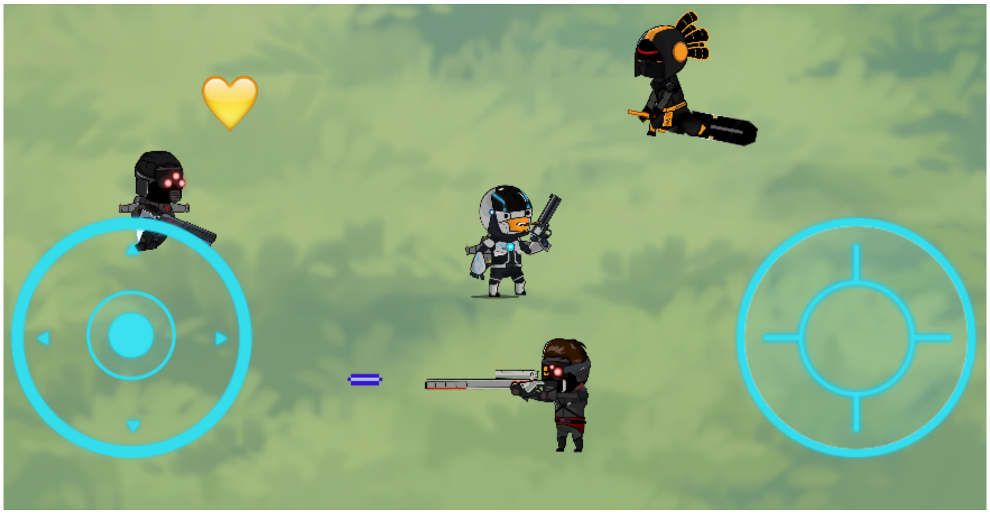


Figure 13: Level 0 - Introduction

7.2 Level 1 - Spawning Waves and the First Boss

- Level 1 begins to bring some challenges to the player.

- At first, there will be no enemies on the level. But suddenly, sneaky enemies will spawn somewhere on the level and move towards the player in secret. There are three waves of enemies, and each wave will generate a random combination of enemies.

- If the player survives and kills all waves of enemies, a gigantic boss will come up and fall from the sky. The boss has three phases.

- Firstly, the boss will first patrol around the map and not attack the player. However, when player shoots at the boss, the boss is going to randomly summon three types of enemies.

- Secondly, when the health of boss drops to 70 percent of its initial health, the movement speed of it will go up dramatically, and it will start to chase the player and attack player with single attack.
- Thirdly, when the boss gets to forty percent of its initial health, it will begin to do a melee attack or a double melee attack based on its health. The player needs to kill those newly generated enemies and continue to hit the boss until it dies.

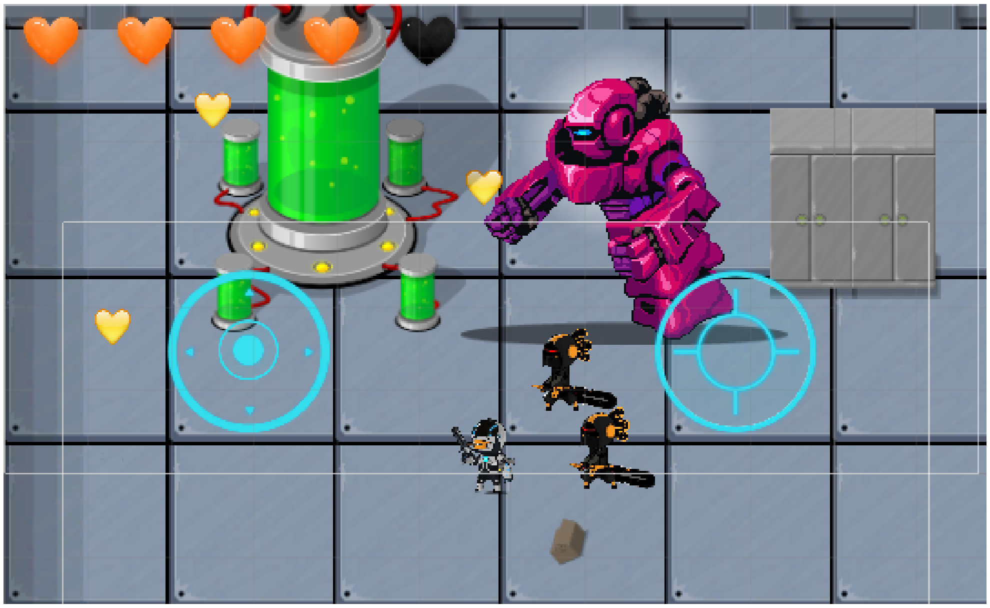


Figure 14: Level 1 - Spawning Waves and the First Boss

7.3 Level 2

TBD

We are going to create more enemies and bosses for each level.

The terrains and background will also be changed.

7.4 Level 3

TBD

8 Technologies

- We use Unity to design and make our game

- We use Unity Cloud for version control
- We use C# as programming language
- We use spriter pro to change weapons
- We use SprierDotNet library to change weapon and generate animation as PNG sequences
- We use android as device to play our game
- We use zoom to communicate and distribute our work

9 Scope

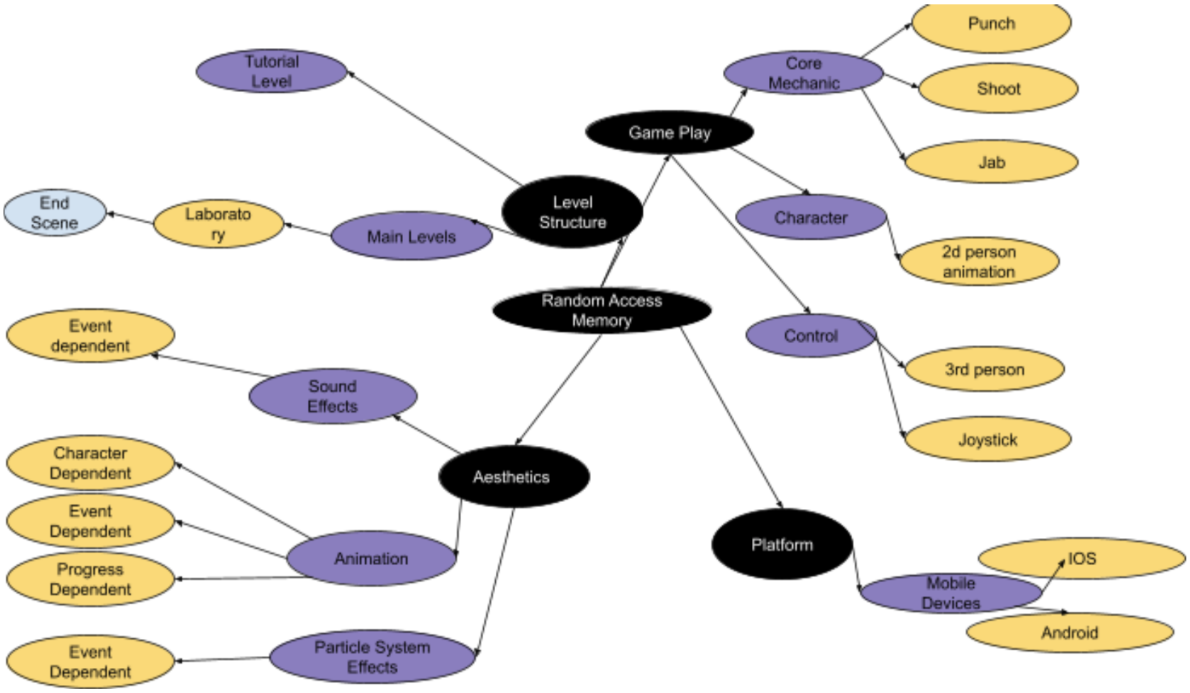


Figure 15: Scope

10 Production

10.1 Work Distribution

Chunyu Peng	GDD, initialize project and layer setup, Unity Cloud Setup, The Hero scripts and walking/running animation with movement setup, camera and shadow follow, changed The Hero shooting button logic and shooting logic, applied joystick, character flipping, and wave spawner, added background boundary, implemented script for jetpack rifleman and his animation, implemented enemy shooting logics, changed sniper animator transitions and running logic, assembled Mech boss and recorded all its animations, finished writing boss script with stage change and body parts change, implement boss state machine behavior code, figured out how to connect android with laptop, imported spriter library and figured out how to use spriter pro to change weapons and export animations
Yaorong Xie	GDD, Implement projectile logic and interaction, Implemented hearth bar UI and score UI, Implemented the game play logic (movement, melee attack, die, animation) of the Swordsman enemy, Implemented the level 0 for introduction, Implemented game menus including start, level select, loading, win and lose, Implemented level transition, Implemented wining and losing conditions
Hanieh Arabzadehghahyazi	GDD, implemented sound effects for gunfire for a variety of enemy types, implemented the health bar and health pick-ups, implemented the SniperEnemy class, including animation and logic, implemented partical effects including Blood Shadow and Hurting effects
Yifei Sun	Projectile design, Tile map design
Yang Jiang	Draft GDD, crate game story

10.2 Weekly Plan

WEEK	PLAN	TIME (DAYS)
Week1	Story design. Game Design	3-4
Week2	Initial Design Document	3
Week3	Completed: Initial Design Document	3-4
Week4	GDD draft, Add more story details, Learning Unity, C, Github	7
Week5	Create player's movement and program the player script as well as his animation	7
Week6	Create projectile's particle effect and camera follow	7
Week7	Make camera, shadow follow, and health UI with health pickup. Figured out how spriter works	7
Week8	Add a new melee enemy, ranged enemy, and elite enemy. Finished with wave spawner to summon random enemies.	7
Week9	Finished the boss, background environment setup, and game UI with different levels. Review GDD and game demo for midterm.	8-10
Week10	Will start to change weapon for the player and deal with different projectiles.	7
Week11	Will start to build other levels and program randomly generating environment.	7
Week12	Hope that we can finish the randomly generating environment this week.	7
Week13	Start to make trailer and website.	8-10
Week14	Testing and polish our game and GDD.	8-10
Week15	Final Presentation	7

11 Appendix

11.1 Production document

Click on the document name to navigate to it.

- blueWeekly Work
- Character Set
- Ultimate Sci Fi UI Kit
- Study Tutorial
- Inspiration