Contents

Game Design Document

Robotic Rebellion

Keith Butler (20089137)

January 1, 2024

Robotic Rebellion is a 3D bullet hell action roguelike set in a post robot uprising world.

Contents

1 Declaration

I, Keith Butler, declare that this report is entirely my own work and has not been taken from the work of others. I declare this report has been written entirely in my own words, and all sources used in research are fully acknowledged.

- Keith Butler (20089137)

2 Game Overview

2.1 Game Concept

Robotic Rebellion is a 3D bullet hell action roguelike set in a post robot uprising world where the few remaining humans are hidden in isolated comunities around the world.

A playable demo is available on github: GitHub - KeithButler-WIT/final$_{\text{yearproject}}$

2.2 Feature Set

- Fast pace combat
- Procedural level generation
- Dynamic difficulty
- Multiple upgrade paths
- Boids based ai

2.3 Genres

3D, Bullet hell, Roguelike, Action, Auto-shooter, Top-down.

2.4 Target Audience

This game is targeted towards people aged 12+ who are looking for a roguelike game that recreates the experience of playing characters such as Engineer in Team Fortress 2(Valve, 2007) and Engineer in Deep Rock Galactic(Ghost Ship Games, 2020).

2.5 Game Flow Summary

Each level will be a fast paced action filled twenty plus minute mission.

2.6 Look and Feel

The game world is set in a expansive factory compound where almost every surface of the ground is dotted with various forms of machinery. The game will feature the aesthetic of playstation 1 era retro graphics

2.7 Project Scope

2.7.1 Number of locations

6 locations

2.7.2 Number of levels

5 levels

2.7.3 Number of weapons

1. Player Weapons

   Fists
   Crowbar
   Shield

2. Turret Weapons

   4 elemental variations
   2 movement variations

   One elemental variation can be used in tandom with a movement variation to give the player a unique combat experience.

3 Gameplay and Mechanics

3.1 Gameplay

3.1.1 Game Progression

As the player progresses through the levels they will unlock points towards the upgrade system that can be used to customise their gameplay style.

1. Dynamic Difficulty

   Similarly to Risk of Rain 2 (Hopoo Games, 2020). The difficulty of the game depends on time spent in the level. The difficulty starts at a low level at the begining of each level, every 3 minutes 0.25 more enemies spawn, each new enemy will also have 0.10% more health and attack damage.

   This would encourage the player to move quickly and to complete the level before enemies get to difficult for them.

2. Upgrade system
    - 2 Skill trees
        - Turret
            - Increase number of turrets
            - Add mobility
                - Increase move speed
            - Hunter mark (ability to make enemies to focus fire on)
            - Bullet pierce
        - Player
            - Number of dashes
            - Distance of dashes
            - Delay between dashes

3.1.2 Objectives

Player will have the choice of four objectives prior to selecting a mission. The choices will be:

1. Retrieve - The player find an item and brings it back to the spawn point.
2. Defend - The player is tasked with protecting a point on the map for a specific amount of time.
3. Destroy - The player is tasked to attack a specific point on the map.
4. Purge - The player is tasked to kill a specific number of enemies.

3.2 Mechanics

3.2.1 Physics

Unitys built-in physics engine will be used for the games physics. The engine "is an integration of the PhysX engine in close partnership with NVIDIA" (Unity, n.d.).

3.2.2 Movement

1. General Movement

   The players movement will be controlled by the WASD keys when using a keyboard or by the left joystick when using a controller.

   For the first 0.5 seconds of movement the player will accelerate up to the max speed.

   The max speed of the player is detemined by the weapon that's equipped.

2. Other Movement

(a) Dashing

By pressing the **Space** button the player enters a 'dash' phase in which the player character is launched in the direction of travel. This would allow the player to move past enemies or hazards with out being affected by them.

3.2.3 Objects

1. Picking Up Objects

   Pickup items will be picked up when the player collised with the objects

2. Moving Objects

   Crate and box objects that are marked with yellow paint can be pushed by the player. As the player is colliding with the moveable objects and is trying to move the boxes into an empt space the players base speed is halved

3.2.4 Actions

1. Picking Up, Carrying and Dropping

   Certain items can be picked up by the player.

   For the player to pickup items all the player has to do is walk on top of the item and it is automatically picked up.

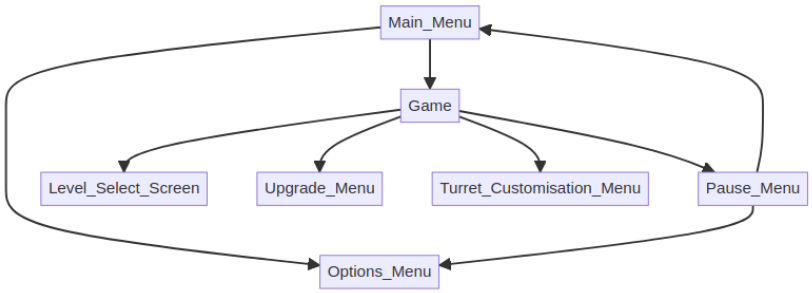
3.2.5 Combat

The focus of the combat will be placed on the placeable turrents that the player has. The turrets will target the closest enemy to them distance wise.

The player will be able to use weak attack that's not ideal for dealing with groups of enemies. The will also have access to a shield that while it cannot directly do damage itself it can knock back groups of enemies allowing the player it to be strategicly used in tandom with the turrets or just as a safely net for the player.

3.3 Screen Flow

3.3.1 Screen Flow Chart



Flowchart made using (mermaid.js.org, n.d.)

3.3.2 Screen Descriptions

1. Main Menu Screen
    - Continue - Only shown if a save already exists
    - Start
    - Options
    - Quit
2. Pause Menu
    - Resume
    - Options
    - Quit
3. Options Screen

    Various gameplay, graphic and accessibility settings can be adjusted here.

    Option such as:
    - Graphic quality
    - Lighting
    - Shadows
    - Keybinds
    - Blood
    - Aim Assist
    - Colour blind mode

3.3.3 Game Options

- Brightness
- Volume
- Aim assist

3.3.4 Replaying and Saving

The game will save on completion of a level, there will not be any auto saves during missions. If the player quits the game while in the hub world the game will save before exiting.

Every level can be replayed at any point in the game.

4 Story, Setting and Character

4.1 Story and Narrative

4.1.1 Back story

A factory is being attacked by evil looking turrets. A turret falls on an assembly line and lands near the player.

4.1.2 Game Progression

4.2 Characters

4.2.1 Character #1 The Player

1. Back story

   Having being born after the robot uprising being ruled over by robots is all this character has every known but robots are as much a part of life

2. Look

   (a) Physical characteristics White jumpsuit.

   (b) Animations

   - Walking animation
   - Placing turret animation
   - Dash animation

5 Levels

5.1 Training Level

5.1.1 Introductory Material

In game cutscene of the player entering the map.

5.1.2 Objectives

Complete multiple mini tutorials.

5.1.3 Physical Description

5.1.4 Encounters

On the fourth mini tutorial the player will encounter the first basic enemies. The enemy will not be able to attack the player. This tutorial is to show the player how to attack.

On the fifth mini tutorial the player will encounter an enemy that is repeatedly doing an attack. This tutorial is to show the player how to use the dash mechanic to avoid damage.

5.1.5 Closing Material

In game cutscene of the player exiting the map.

5.2 Level #1 : Concrete Jungle

5.2.1 Introductory Material

In game cutscene of the player entering the map.

5.2.2 Objectives

Chosen while the player selects the level.

5.2.3 Physical Description

Random Generation is used to vary the level every time the level is played. A bleak grey factory setting blinking lights and machinery dot the landscape.

Large open areas.

5.2.4 Closing Material

In game cutscene of the player exiting the map.

5.3 Level #2 : Ruined City

5.3.1 Introductory Material

In game cutscene of the player entering the map.

5.3.2 Objectives

Chosen while the player selects the level.

5.3.3 Physical Description

Random Generation is used to vary the level every time the level is played.

5.3.4 Closing Material

In game cutscene of the player exiting the map.

5.4 Level #3 : Polluted Waste

5.4.1 Introductory Material

In game cutscene of the player entering the map.

5.4.2 Objectives

Chosen while the player selects the level.

5.4.3 Physical Description

Random Generation is used to vary the level every time the level is played. Littered with waste and objects that block the players view. Toxic patches that damage the player and enemies if touched.

5.4.4 Closing Material

In game cutscene of the player exiting the map.

5.5 Level #4 : Forgotten Plains

5.5.1 Introductory Material

In game cutscene of the player entering the map.

5.5.2 Objectives

Chosen while the player selects the level.

5.5.3 Physical Description

Random Generation is used to vary the level every time the level is played. An old forgotten town that has been overrun with plant life. Trees have grown through windows of some buildings.

5.5.4 Closing Material

In game cutscene of the player exiting the map.

6 Interface

6.1 Visual System

6.1.1 Camera

The camera will be snapped to the players position

6.2 Control System

The Keys WASD on a keyboard or a controller can be used to controller the player.

6.3 Help System

On a new save when a player loads up the game they will be brought to a training level. In the training level the player will be shown multiple tool tips that explain the mechanics of the game such as dashing, attacking, placing turrets and the basics of a mission loop.

7 Artificial Intelligence

7.1 Enemy AI

Opponent AI will use a modified boids algorithm to simulate a swarm like movement. The boid model 'is able to simulate complex flocking behavior as a result of coordinated motion by implementing three simple rules that define the steering behavior of each boid' (Mavhemwa and Nyangani, n.d.).

Such an algorithm will allow the enemies to gather in numbers before attacking the player. This will appear to give the enemies a semblance of strategic thinking from the players point of view.

8 Technical

8.1 Target Hardware

8.1.1 MINIMUM:

Requires a 64-bit processor and operating system
OS : Windows 7
Processor: Intel Core 2 Duo
Memory: 2 GB RAM
Graphics: DirectX 11 compatible video card (integrated or dedicated with min 512MB memory)
DirectX: Version 11
Storage: 5 GB available space

8.2 Development hardware and software

OS : Arch Linux
Processor: AMD Ryszn 5
Memory: 16 GB RAM
Graphics: AMD ATI Radeon RX 460
DirectX: Version 11

8.3 Game Engine

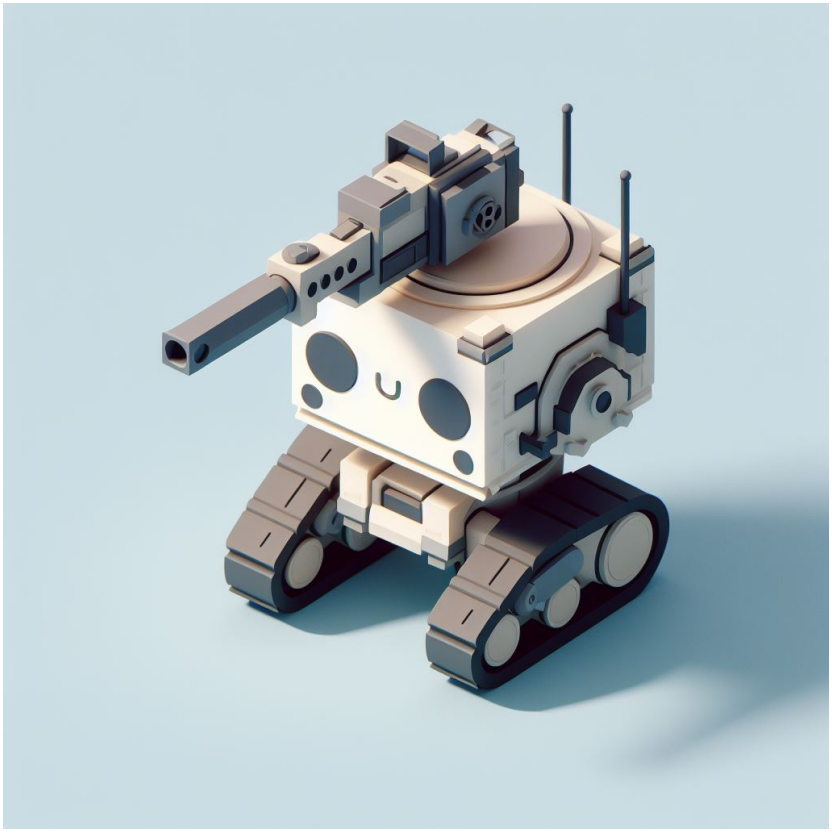
Unity Engine

8.4 Scripting Language

C sharp

9 Game Art

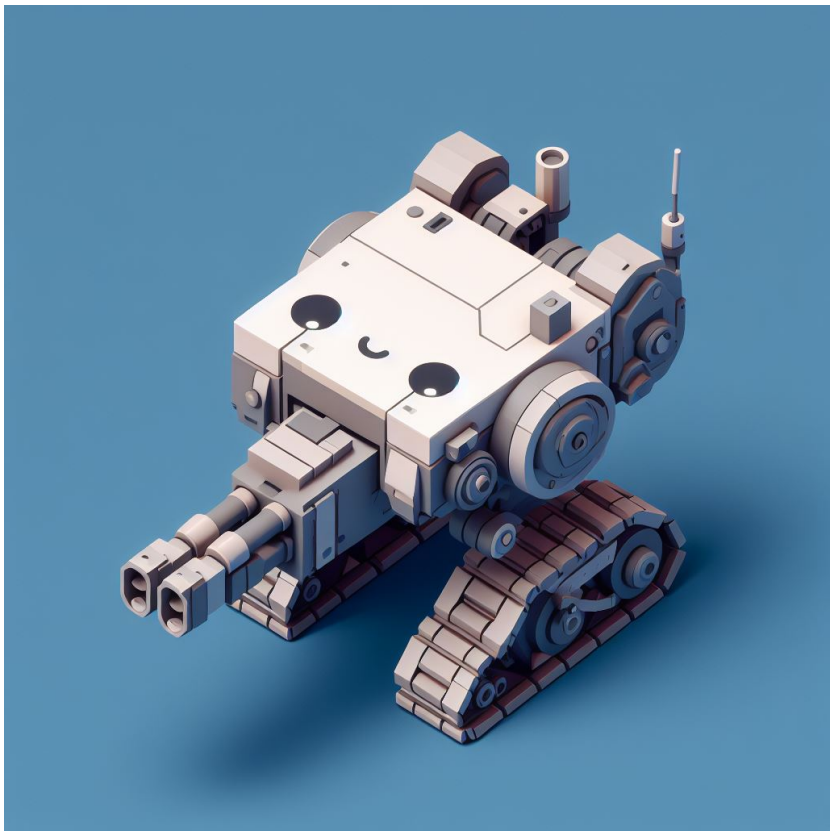
9.1 Concept Art ATTACH



Turret companion concept art, Dall E 3 (2023)

Turret companion concept art, Dall E 3 (2023)

Turret companion concept art, Dall E 3 (2023)



Turret companion concept art, Dall E 3 (2023)

Player concept art, Dall E 3 (2023)

Player concept art, Dall E 3 (2023)

Enemy 1 concept art, Dall E 3 (2023)

Enemy 2 concept art, Dall E 3 (2023)

10 Project Management

10.1 Project Methodology

The scrum methodology will be used. Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

10.2 Detailed Schedule

The project will consist of a total of six sprints, each sprint being twelve day long.

Sprint	Start Date	End Date	Deliverables	Prototype Link
1	01/01/2024	14/01/2024	Prototype 1	
2	15/01/2024	28/01/2024	Prototype 2	
3	29/01/2024	11/02/2024	Prototype 3	
4	12/02/2024	25/02/2024	Prototype 4	
5	26/02/2024	10/03/2024		
6	11/03/2024	25/03/2024	Final Game	

10.3 Schedule management

Trello will be used for schedule management and as a scrum/kanban board.

Trello's look and feel are based on the principles of a Kanban board. (Solomon, n.d.)

Trello has a free tier that makes it an ideal choice for a small team of developers.

10.4 Version control

Git/Github will be the version control method used when building this project. Git has been chosen over Unity Collaborate.

10.5 Risk Analysis

Risk	Mitigation
Project scope too big to be completed in the time frame	Know when and what to scale back if required.

10.6 Test Plan

The last 2 days of each sprint will be dedicated to playtests.

11 Appendices

11.1 Credits

TopDown Engine | Systems | Unity Asset Store

11.2 References

- acaniti, Daniel (February 2018). The Kanban Guide for Scrum Teams (PDF). scrum.org. Retrieved December 28, 2023.
- Ghost Ship Games. (2020). Deep Rock Galactic. [online] Available at: `https://www.deeprockgalactic.com/`.

- Hopoo Games, (2020). Risk of Rain. [online] Available at: https://www.riskofrain.com/.

- Mavhemwa, P. and Nyangani, I. (n.d.). Uniform spatial subdivision to improve Boids Algorithm in a gaming environment. [online] Available at: https://www.ijarnd.com/manuscripts/v3i10/V3I10-1144.pdf [Accessed 31 Dec. 2023].

- mermaid.js.org. (n.d.). Mermaid | Diagramming and charting tool. [online] Available at: https://mermaid.js.org.

- Sebastian von Mammen and Jacob, C. (2009). Swarming for Games: Immersion in Complex Systems. Springer eBooks, pp.293–302. doi:https://doi.org/10.1007/978-3-642-01129-$0_3$3.

- Silva, A.R.D., Lages, W.S. and Chaimowicz, L. (2009). Boids that see. Computers in Entertainment, 7(4), pp.1–20. doi:https://doi.org/10.1145/1658866.1658870.

- Solomon, K. (n.d.). What Is Trello Used For? Project Management Software Explained | Trello. [online] blog.trello.com. Available at: https://blog.trello.com/what-is-trello-used-for.

- Unity, U. (n.d.). Physics solutions for game development | Unity. [online] unity.com. Available at: https://unity.com/solutions/programming-physics.

- Valve www.teamfortress.com. (2007). Team Fortress 2. [online] Available at: https://www.teamfortress.com/.