Game Design Document - Robolution

1. Vision

- Robolution is a challenging, action oriented 2,5D singleplayer sidescrolling shoot-em-up with a distinct Cyberpunk look and feel and a lot of destruction!

2. Pillars

- Highscore system
- Cyberpunk feeling
- Arcade feeling (for core gamers)
- Satisfying explosions and destruction

3. USP

- Action packed: player will constantly face enemies with **challenging waves and peaks**, always in movement and forced to react
- Challenging gameplay: The player has **a very limited health** and thus has to constantly **move, dodge and shoot**
- **Cyberpunk** setting, which will engulf the player in the world of Robolution
- **Arena Mode:** player will face a weave of enemies

4. Research

a. Art Style

- Ghost in the Shell (Environment)
- Blade Runner

b. Level Design / Gameplay Design

- Metal Slug X
- Mega Man X & Mega Man Classic
- Duke Nukem Manhattan Project

5. Gameplay

a. Space (Environment)

- The game logic is mostly **2D, but 3D assets** are being used. The player is able to navigate on the x & y - axis only, but not able to go in depth.
- “Magic Circle”: The game takes place on the **streets of a cyberpunk-ish city**, with houses in the background.
- The exact art guidelines are to find at the art bible.
- The level path is a **sidewalk** in front of the houses and slightly above the streets, partially taking place on top of their roofs

b. Pacing

- The game is **challenging, with multiple difficulties**
    - Casuals can play through the level on “normal”
    - Experienced players can beat “Robolution” on “hard”
    - “Veteran” is a secret, trial and error difficulty, allowing almost no mistakes
- The speed is rather **fast**, but encounters are still challenging and the player **cannot just “run through”** the level
- The player has a **limited amount of mistakes** he can make and has to watch his steps carefully
- There are **always enemies** shooting at the player, categorized in
    - **Single enemies** (no big threat, player has a feeling of empowerment by successfully dodging and shooting at just this one enemy)
    - **Enemy groups**: **Challenging constellations** (supported by the level structure) the require the players’ skill and concentration
    - **Arena Mode**: an overcome of enemy waves that the player is forced to deal with, locking the camera to the center of the arena

c. Rules & Goals

- The player must reach the **end of the level** and **defeat the boss**
- The player has a set amount of **health** (communicated through UI), and he loses when it falls to 0
- The player has set **movement controls** (see player control), however, the player can customize keybindings for shooting and jumping
- Each enemy type has a defined **moveset** and there are different **enemy types**
- The players weapon has infinite and his upgrades limited **ammunition**

d. Power-Ups

i. Laser

- This power-up will change the player's weapon to shoot a powerful laser ray that **pierces enemies**
- Before shooting, the beam charges up
- The laser **instantly** kills all enemies and **damages** as long as the ray is visible
- It has a limited ammunition of **3** shots

ii. Fire Rate

- Significantly increases the fire rate
- Multiple upgrades don’t increase the fire rate further
- The machine gun doubles the fire rate roughly
- It has **35** ammunition
- If the Player collects another fire rate-pickup, the ammunition will be **reset**

iii. Health

- Instantly refills the players **health by +20** (2 bars).

e. Gameplay-relevant Objects + resultant actions

- **Explosion**(Visual feedback) if enemies are shot
- Enemies **blink** red if they are hit (Visual feedback)
- **Shattering pieces** when enemies explode
- Barricades and boxes as **obstacles** or objects to jump on
- **Powerups** (s.a.)
- **Platforms** that the player can jump on

f. Score

- Score should **reward the player** for a good performance throughout the level
- Score should increase the replayability of the game, providing a well-calculated and **unique resumé** of the level walkthrough
- The exact description of the score system is found in the Score System Document

6. Player

a. Control (operative actions)

- The player has a **defined moveset**, such as jumping and crouching which stays the same throughout the level
- The input for the main actions, jumping and shooting, can be adjusted by the player in *main menu -> options ->* ***keybindings***
- Movement should feel **fluid** and at the same time **instant**, so that the player has precise control over his navigation and a clear feedback
- By crouching, the player can dodge bullets
    - Player gets slower if he crouches
- The game offers Xbox / Ps4 **controller support**

b. Variables / Values

- The player will **survive multiple hits**, exact health value dependent on the pacing and the level / enemy design
- For exact variables & values documentation, see Balancing Plan
- Player (& enemy) health and damage are in 10-steps to make later (eventually more precise) damage calculation possible
- The jump mechanic is aimed to provide more **air control to the player**; it does not have to be a realistic jump - therefore, jumping height and the curve in general are slightly exaggerated

c. Mechanics & States

- If the player is hit by a damaging source (enemy melee / bullet), he is **invulnerable** for a short time, and the screen is also blurred for a second.
- If the player is in the ingame-menu, the game is **paused**.
- The **enemies** have an **idle** and an **active** state

---

7. Enemies

a. Destruction

- Shooting at enemies must feel **satisfying**, and as such must be rewarded with epic vfx / sfx and smooth movement and combat control.
- Shattering pieces of destroyed robots and **explosions** will help the player to feel that reward
- Explosions will deal no damage to the player

b. General Behavior / AI

- At first, enemies are **inactive**. When entering their **detection range**, they will be activated after a short **delay (warmup)**.
- Enemies know the players position and will try to **chase** him
- They always try to **keep a set distance** to the player and attack him from there.
- Enemies **charge before shooting**. This warmup time is visually supported by charging particles and a clear signal as to when it will shoot
- Enemies cannot actively navigate on the **y-axis**, except drones

c. Tank

- For variables & values documentation, see Balancing Plan
- Tanks are about **1.25x bigger** than the player and can shoot to the left & right.
- Tanks shoot at a height where the player can **crouch to evade** the bullets, and can still **jump over them** with good timing
- Tanks take **3 normal hits** (30 health)

d. Melee

- For variables & values documentation, see Balancing Plan
- Melees are about **as big as Sentries**(~0.5x the player) and can move with a set speed towards him (cannot jump), causing **damage on collision**.
- When the player is inside their detection area, they **charge** at first and then perform a **dash** in the direction of him
- Melees take **2 normal hits** (20 health)

e. Drone

- For variables & values documentation, see Balancing Plan
- Drones are **half as big** as the player
- Drones are **flying mid-air** and are harder to reach.
- Drones **shoot in a diagonal line**
- Drones chase the player in a limited area in **set time intervals**
- They **reposition** themselves in set intervals in order to give the player an opportunity to shoot at them in a set time span (weak point, no constant movement)
- Drones will take **1 hit** (10 health), because they are hard to hit and are not in the “action line” of the game (the player has to navigate to the right, so shooting drones above is off-the-path)

f. Sentry

- For variables & values documentation, see Balancing Plan
- Sentries are **half as big** as the player and shoot at a line where the player **cannot crouch to evade** the bullets, but **jump over them**
- Sentries will take **2 hits** (20 health)
- Sentries can still be hit without the player having to crouch, but cannot be hit if he is jumping

g. Boss

- For variables & values documentation, see Balancing Plan
- To finish the level, there will be a small boss fight at the end of the street
- Due to missing time, the Boss will be a combination of the **melee model and tank canons**
- The Boss is **twice as big** as the player and **moves slowly towards** him
- The Boss has **4 canons** and a simple **shooting behavior**: He charges one canon after the other and then shoots
- If his health is **below 50%**, he changes his behavior in which he **rapidly shoots multiple bullets for 5 seconds**

---

8. Level Design

a. Boundaries

- The level has a **fixed starting and ending point**, communicated by boundaries and through the camera not scrolling sidewards at the end
- **Modular** Levels : Working with modular assets and small “level pieces” to create fast levels that can be edited easier

b. Arena Mode

- The **camera stops** following the player, listed enemies spawn from predefined areas, if all enemies are destroyed the camera starts following the player again.
- **Laser barriers** form a border from which player and enemies cannot escape. They are active as long as enemies from arena mode are alive

c. Tutorial

- At the beginning the player can mark a **checkbox to play a tutorial**, so he **learns** all the **mechanics** and the enemy’s shooting **patterns**

- In the beginning there are three **stationary enemies** (sentry, tank & drone) to learn how to **dodge bullets**
- A **melee** is placed behind a barricade so the player gets to know his behavior
- At last, a **powerup** is introduced and **two normal enemies** appear, to foreshadow the player a piece of the level
- When the player completes the tutorial, he is teleported into the main level

d. Boss

- If the player encounters the Boss, the **camera** will be locked like in the **Arena Mode**, so the player has a boundary(and low set of movement) and to **see** the boss better.
- 2 **platforms** are place for the players **safety** and to avoid the bullets better

- To see the Player even in front of building walls, we placed **boxes** to compensate the perspective distortion.
- For more exploration, some **pickups** are **hidden**
- For more variation in the level itself, **platforms** are placed **vertically** as a **background asset**
- There is only **one long level**, like it is the case in Arcade games
  - But there will be a **tutorial**, to teach the player the mechanics
- At the **start**, there are **less enemies** placed to get a good start and to get to know the enemy behavior; After reaching the first arena, the level itself **gets harder**, with more enemies(like enemy groups) placed.

---

9. User Interface (Interactive - Buttons etc.)

a. Detailed information are to find in the UI Document.

b. In-Game

- In game, the player can press escape to pause the game and enter the **pause menu** screen. From there, he can **resume**, access some **parts of the options** in a new options menu, go back to the **main menu** or **quit** the game
- Options menu in-game is restricted so that the player only has access to the most needed settings and not everything else he could set up in detail. The player can only access the **volume** settings.

c. Main Menu (Pre-Gameplay)

i. Play
- Loads a predefined scene (Level01)

ii. Quit
- Exits the application

iii. Options
- Allows the player to set **music volume, brightness, sfx volume, keybindings & difficulty**

iv. Difficulty
- There are 3 difficulties, **normal, hard & veteran**. For more information, see difficulty information in pacing
- The game is balanced on normal
- Veteran difficulty is only unlocked after the player has completed the level once, to prevent frustration

v. Credits
- Displays the names of Git Gud entertainment, either in an image or in a short animation

vi. Score Menu
- The score menu stores the highscore of the top 5 playthroughs
- Information about highscores and a check whether the player has completed the level once will be **stored externally**

---

10. User Interface (Presentative - Screens etc.)

a. Intro

- At the start of the **application**, there is a short **intro animation**
- After actually starting the game, a short film is played, displaying the **stories framework** and explaining (by Voice Over) the setting

b. Ending Screen

- The game ends after the player has reached the **end point** of the level
- His last encounter is a Boss Fight
- If the player has 0 health, the Game Over Screen pops up
- The screen provides the options to replay the level, go to the main menu or quit the application
- The **final score** (calculated with final time multiplier) will be displayed there

- Could have: The player can enter his name and save the score to a high score list

---

11. Player Learning

a. Controls

- The **inputs** of robolution are **simple and straight**. The player only has to learn how to move, crouch, shoot and jump
- **Feedback to input** is instant and as such, the controls and their effects are easily understandable
- Controls are oriented on **conventions**, so that the player instinctively uses them correctly
- Player learns **new weapon types** throughout the level, and this is **communicated** through UI, shape of weapon and the color of the bullets

b. Enemy behavior

- Enemies have a **similar base movement pattern** and, at this point, are solely different in small tuning variables (Sentries, Drones and Tanks all try to keep a set distance to the player and attack from there on, Melees always chase the player).
- Each enemy has an aspect that makes him unique, such as the drones repositioning themselves in intervals to give the player a chance of hitting them, or the melee charging up for a dash
- **Straight** forward and linear movement for better prediction
- They have clearly **visible signs** of what they are about to do, such as charging particles when they are going to shoot

c. Visual language

- The **background** is dark and vague, not drawing too much attention to it. Thus, the player automatically focuses on the gameplay-relevant objects
- **Enemy shots** are red and have a relatively big sprite, so not only does the player know in advance when multiple enemies shoot; he always has an overview of where the shots are
- **Player shots** are yellow and smaller(for the default weapon), so he know in which direction he is shooting (and consequently what he is facing) without getting confused by enemy shots

- Player learns new weapon types throughout the level, but is communicated through UI, shape of weapon and the color of the bullets
- The players **texture** colors are in contrast to the background and are remarkable through higher saturation
- **Enemies** have a diffuse map and a **glow**. Plus, they **blink** in red when they are hit
- **Power Ups** are in the color of the player, or green for a health powerup, so that he can already assume the meaning of the powerups by their look
- **-> in general, gameplay-relevant information are clearly structured, simplified and visually highlighted to help player orientation and information processing**

Attachments

Balancing Plan
Score System Document
UI Document