

One-page(or 2 :) Design Document

Reference GDD HowTo video: <https://www.youtube.com/watch?v=q96lz725glw>

Game Identity / Mantra:

List your single sentence description of the game that you will use to guide design decisions. (Example: *Stylized action platformer about a meatball fighting the dinner table.*)

Mind-twisting puzzle game about helping scientists learn to control a single robot in multiple dimensions, at the same time, in mazes-like levels, getting through a series of obstacles, getting past some enemies and reaching a goal.

Design Pillars:

List up to 3 words/phrases that convey the feeling or emotion you want the player to experience. (Example: *Fast. Action-packed. Mayhem.*)

Exploration - a sense of Adventure -; *thoughtfulness* and *happiness (excitement)* to solve the puzzles.

Genre/Story/Mechanics Summary:

List what the game is from a gameplay and/or story perspective. (Example: *This game uses a unique swinging rope mechanic to tell a story about what it means to be a meatball...*)

- This is a puzzle game, where the player controls one Robot (and its mirror image in another dimensions) at the same time, both robots have their own third person camera behind them.
- Both move forward and backward simultaneously, but left and right are reversed. The robots can also be scaled in size to get past obstacles.
- As there are some enemies too, the Robot (and its mirror image) can also use a “stunning weapon” that creates an “Electric Field”, 2 meters in front of them, disabling any robotic enemies in that area.

Features:

List the cool features or unique elements that you want to include in your game.

- The ability to **scale up or down** immensely.
- The ability to **move two characters** (one being a mirror image of the other) **at the same time**.
- The Robot’s two representations are able to **move boulders** (rocks) and some heavy obstacles by pushing them.
- The ability to **fire a weapon**, creating an “Electric Field” in front of the Robot’s representations, to stun enemies or clear specific obstacles out of the way.

Interface:

List the player input method, the controls, and how the player interacts with your game.

Mouse and keyboard:

- WASD: movement.
- Q: scale down.
- E: scale up.
- Mouse Click: use the stunning weapon.
- The screen is divided by its half, so you can see the two Robots at the same time.

Art Style:

Include references to lots of images and games that have a similar aesthetic to what you're trying to achieve.

- To be decided

Alejandro's contribution: I saw 2 options that I liked:

1. Option # 1: Pseudo realistic (URP + 33% Bloom)-colored:



I would propose doing this by following Lesson 1 of the CodeMonkey Course (TBS: Turn-based Strategy Course, from GameDev.Tv), for **setting up the URP and Post-Processing** for that project.

Notes:

- a. There is a black color around the Maze, to simplify the Level Design and help the Player focus his/her attention on the Map itself.
- b. The appearance of the Maze wall has to be like "a Lab Prototype" for Experiments. This allows us the simplicity and to save some time.

Another example of the interior of the Walls of the Laboratory and Maze (please check the color and patterns in the Laboratory Walls in the background - Note: This Game is: *Half-Life 1*):

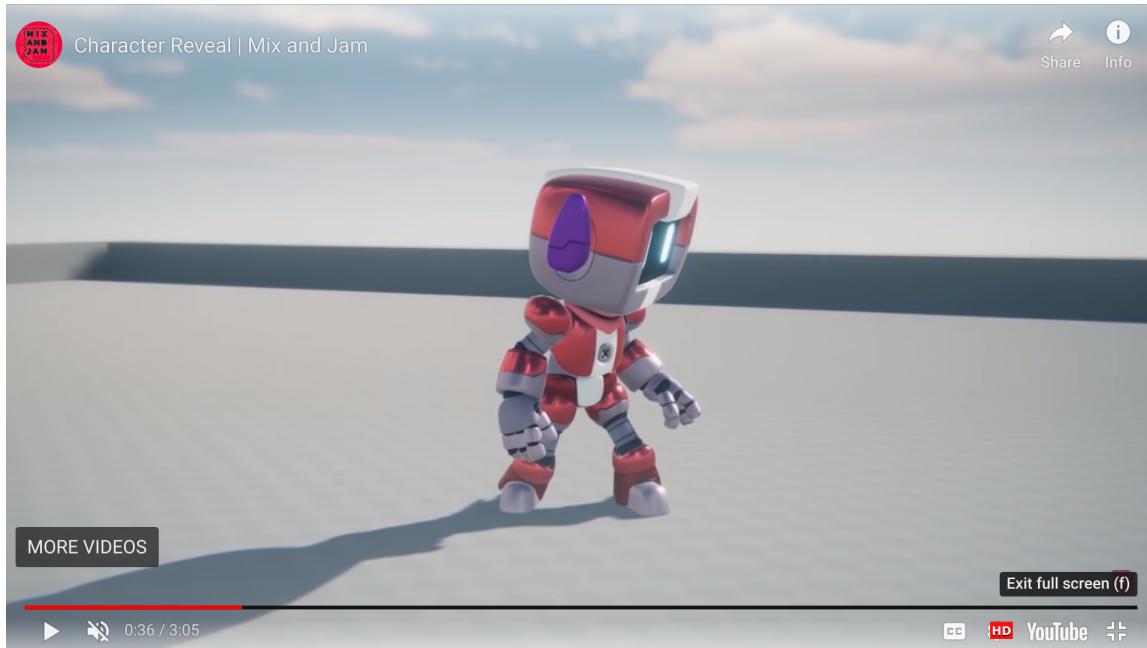


Screen Split example: (please also see the Camera angle from “Player 1”, semi-Top-Down, as an example - Note: This Game is: Super Mario 64 - Local LAN Multiplayer, Split Screen).



Playback speed: 105%
Playback speed: 100%

And our “Hero”, (the Robot) could be the Jammo Character made by **Mix and Jam**, for Game Jams (Note: the Walls could be like this too..., let’s say that some experiments could have a “simulated sky” (by using Holograms, etc), or they’re done outdoors, like that one):



Asset Store download:

<https://assetstore.unity.com/packages/3d/characters/jammo-character-mix-and-jam-158456>

GitHub Repo: <https://github.com/mixandjam/Jammo-Character>

2. **Option # 2: Toon Shaders**, using Unity's Open (source) Project Assets: [Open Project](#)

#1: "Chop Chop": <https://www.youtube.com/watch?v=GGTTHOpUQDE> (GitHub Repo:

https://github.com/UnityTechnologies/open-project-1/tree/devlogs/1-toon-shading?utm_source=YouTube&utm_medium=social&utm_campaign=evangelism_global_generalpromo_2020-11-17_open-projects-devlog-branch).



I think this option is not as good as the previous one, because our Robot is metallic, and the Art Style that best highlights its color-attributes is a *pseudo-realistic*-colored **URP** setting, like in Option 1 (although I am open to listen to opinions and see more options).

Music/Sound:

Include links to music and sound design similar to What you're trying to achieve. You can also list the emotional responses that the sound should invoke in the player.

- To be decided

Alejandro contribution:

- **Music:** I propose a series of tranquil but dynamic + futuristic (because of the Robots..., we need some “sense of movement” too...) soundtrack OST (for example: Like “**Dire Dire Docks**”, in Super Mario 64: <https://www.youtube.com/watch?v=Zqa2mgjbOIM>), for free, with Licence CC0.

CC0 - licensed examples: Some Good Options I found by searching here:

(<https://incompetech.com/music/royalty-free/music.html>)

- **Never Wave:** <https://youtu.be/UrAxswzjwnE>
- **Neon Laser Horizon:** <https://youtu.be/EYcAo2DGoKU>

(Optional Repos: <https://incompetech.com/music/royalty-free/music.html>)

- **Sounds:** Realistic sounds with a funny tone to them, we can use this repository:
<https://freesound.org/browse/>

Development Roadmap / Launch Criteria:

Platform: WebGL **Audience:** any aged boy or girl who likes solving puzzles and the “3D Platformer” games genre.

Milestone 1: Design & work split
- **21/05/23**

Milestone 2: Mechanics *Design* complete
(each member in parallel)
- (start: 21/05/23) — End: **22/05/23**

Milestone 3: Mechanics *Programming*
(w/Programmer Art for prototyping) complete
- (start: 22/05/23) — End: **25/05/23**

Milestone 4: Levels complete - **26/05/23**

Milestone 5: Art Polish (Visuals + Sounds + Music) complete
- (start: 26/05/23) — End: **28/05/23**

Milestone 6: Final Code *Polish* (Bug Fixes + last minute Optimizations, if any at all)
- (start: 28/05/23) — End: **29/05/23**

Milestone 6: Polish complete - **29/05/23**

Launch Day: 29/05/23

