Overview

This is a third-person puzzle game with a warping mechanic. We want to capture the feel that your “warping” is akin to shooting yourself at enemies and obstacles. Player movement will be fast-paced, player cannot jump, and they will move fast enough such that running is not necessary. We hope for our warp mechanic to cover for any needs to jump/run. This will be explained much more in depth in core game mechanics section right after.

Theme

We want to give the simple feeling of exploration in a lonely world. Since the main character is a robot with no memory, there are feelings of confusion and desperation.

Story

Story will be the travel of a robot with amnesia to find his lost memories. Why is he in such a barren place, and why does he stand out so much?

Target Audience

Players around our age who enjoys puzzle games. This game will definitely progress slower than other platformer or action games, so favors players who are fine with that.

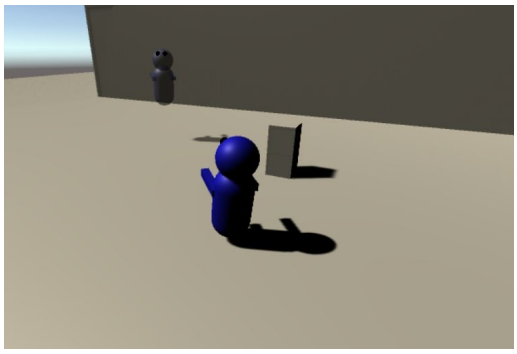
Core Game Mechanic

Player Perspective: The game will be played in third person behind the shoulder perspective where the main character is always placed on the left side of the screen, and the user will generally only see the back of the main character. (E.g. watch dogs or witcher series)

Controls: Since we will be using a xbox controller, here are all the mappings we are planning to use. Core game mechanics are explained along with the controls. The player will not be able to do anything that is not mentioned under the controls mapping.

1. **Left Joystick:** Controls player movement on a 2D plane (left, right, forward, backward), the player will not be able to jump. We are making the movement at a walking pace and the player cannot run to encourage the use of warping.
2. **Right Joystick:** Controls the camera rotation, and when the warp guide is toggled the warp guide will also move with the camera.
3. **Button Y:** This will toggle the warp guide on/off. The warp guide is a translucent version of the player that shows where the player can warp, and it will always be some distance in front of the player. The player can move the guide any direction, however there will be certain restrictions for the guide as it cannot pass through some objects such as the floor of the level.

4. **Left Trigger:** Increases the distances of the warp guide from player, so the warp guide will keep moving forwards away from player.
5. **Right Trigger:** Decreases the distances of the warp guide from player, so the warp guide will keep moving towards the player and it will stop right in front of the player.
6. **Button A:** Changes warp mode to instant warp, the player will instantly appear at the position of the warp guide. If he warps to an enemy or object with this mode he will not affect them as he has zero velocity and appears instantly. In this mode he can warp towards the sky, but will fall directly down towards the ground after warping.
7. **Button B:** Changes warp mode to velocity warp, the player will hurl himself towards the warp guide in a straight line, he will only stop if he hits an object that cannot be moved such as a wall. Anything in his path will be hurled into random directions. He will be travelling very fast and can break glass and move heavy boxes.
8. **Button X:** Warp on top of/towards the warp guide depending on the warp mode.



This is an image of the player, with the translucent warp guide right in front of him. The warp guide can be moved in any direction (other than material restrictions). In this case it is raised a bit higher than player.

Technical Details

So the instant warp will be implemented as a change of player position, where basically their current position is set to the position of the warp. Currently, we haven't implemented how we will prevent the player warping through certain materials. We were thinking that we could add triggers on the warp guide to detect the material it's passing through and if it passes through something like the floor we will indicate to player that it's invalid warp and you cannot warp.

As for the velocity warp, we will make the player move towards the position of the warp guide at a set velocity. Since it's implemented this way, that means the player will not stop until it hits a solid object such as a wall.

Currently the enemy is implemented with a raycast method. It has an arc cone detection range in front of it. So basically in front of the enemy there is a isosceles detection triangle in front of the enemy where the top point starts at player and base away. Since it is using raycast it won't detect players behind objects.

Most of the other parts of the game will be minor implementations, or sfx, or fixes to make the game feel nicer and less rigid.

Level Design

Levels will generally broken down into easy, medium, and hard stages. In terms of number we will probably have 2 or 3 levels for each section of difficulty which will look something like the following. Will go more into depth of a sample level after quick outline.

Tutorial: Learn to walk and warp
Easy Levels

1. Introduction to interactable objects such as boxes, windows, and how to break or move them
2. Introduction to enemies, how they behave and how to defeat them

Medium Levels:

1. A level that requires careful planning and timed execution
2. Level that eases the player simpler level
3. Many enemies, requires rapid reaction (harder)
4. Generic medium level

Hard levels:

1. Boss that will end the story and combine all the knowledge player previously learned

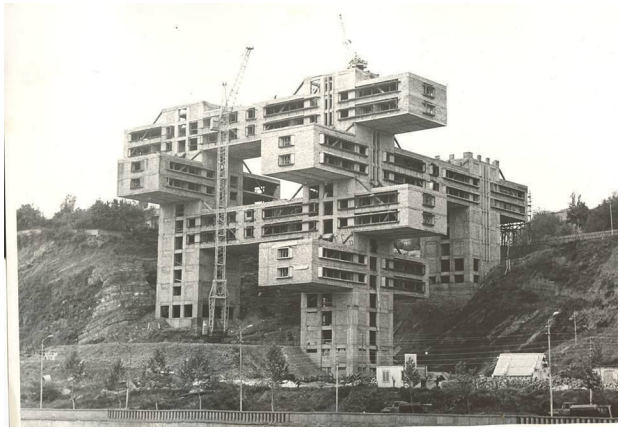
A sample level will be something similar to this. The player appears in a predetermined location, then the player will know the goal since it will be shown as a beam of light into the sky, maybe some sort of broken/malfunctioning warp machine, and the player has to reach it to progress to the next stage. The player always be surrounded by the broken buildings discussed in the art section. However, there will always be some sort of path laid out for the player so he knows the general direction to take.

A puzzle that takes advantage of the warp mechanic will be similar to this. There is a window high up and the only way to see through it is to warp up into the sky then look through the window and then warp to intended location. Another one is a wall that you can see but not reach so you can warp towards it and hit it and fall down towards maybe a moving platform or safe spot.

Another puzzle will be something similar to this. The main route is blocked by a building that has fallen over, so the player has to find another route. In order to find another route, the player may need to break the window of a nearby building and navigate through the building to come out on other side. Or the player will need to maybe find a false wall that’s hiding a secret passage.

Visuals - Art Style

Environmental Design

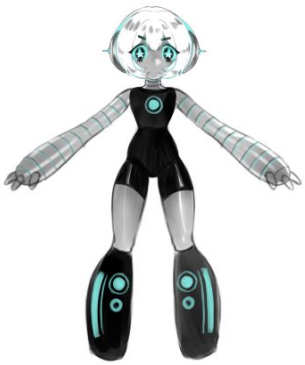


[1] Brutalist Architecture
The image is a good example of brutalist architecture, where the exterior of the building looks blocky. Since our game will be exterior, we will have blocky buildings that will be modular to show off this style of architecture.

[2] Nier Automata
We want our game environment to have a similar feel to the buildings in this picture from nier. The player will be in a sandy/dusty environment surrounded by broken down buildings that show off the brutalist architecture style.

Overall, the color scheme that we want to use for the environment is a darker tone to represent the dead buildings and atmosphere in the game. As for the buildings we were thinking of a modular approach, where we create a few blocks that can be a part of a building, then rearrange them in various ways to make it seems like they are different buildings.

Player Design



This is concept art for our main character. The character is created as a contrast to the game environment. The origin and reasoning of why the character is like this ties to the story.

Enemies Design

This is concept art for our enemies. The enemy design tie in with our environment design, where they have a darker tone and feel.

Visual Effects

Effects that will be added

1. Instant warp effect that signifies the player has moved to the position of the warp guide
2. Velocity warp effect that shows player is actually flying through the air, maybe a flailing arms effect
3. Impact effects where if the player crashes into a wall it will show cracks as an aftereffect

Timeline

Our timeline will be divided into milestone fragments we wish to achieve prior to each of the following stages. Each stage is meant to build up from one another, therefore it is implied in each successive stage that we will try to improve everything listed before it in addition to working on the new items.

By Alpha
Technical: Improved warping mechanic, player movement, camera work, a playable level with a puzzle utilizing our core mechanic as best as possible
Visual: Main character model, environmental objects like ground/wall textures
Audio: Main BGM, warping/movement/idle sound effects

By Mini-Demo 3
Technical: Player-enemy interaction, progression/reward system, tutorial level
Visual: Enemy character models, Brutalist-inspired buildings/ruins mockup
Audio: Enemies/collision/death/damage sound effects

By Beta
Technical: Menus, GUIs, additional levels
Visual: Special effects, animations, Brutalist-inspired buildings/ruins model

Audio: Ambiance sounds, menu sound effects

By Game Release
Technical: Linking together levels, reorganizing levels
Visual: Polishing scene as a whole
Audio: Adjustments to fit into the final scenes

Project team

David (Programmer): character implementation, and level implementation
Felipe (Programmer): level design and level implementation
Jason (Programmer): some physics, and enemies, and level implementation
Tina (Artist): character design, promotional art, 2D art aspects
Jonathan (Artist): audio, textures and terrains
Richard (Artist): 3D modelling

Citations

[1]http://georgiatoday.ge/news/4009/Tbilisi-Soviet-era-Building-Listed-Among-the-Best-Brutalist-Architecture-Examples
[2]https://www.oneangrygamer.net/2017/02/nier-automata-endings-explained/24861/