Project Little Guy

That's it

Aaron Bush - Dariel Ramos - James Zilberman - John Haley - Kevin Insinna

Table of Contents

1. Introduction

This document details all the design decisions made for **Project Little Guy** relating to technical, game, visual, and audio design. As well as some production-related information to keep our team on track.

Please note that this is a **living document**, meaning its contents can change as the game evolves. Regular reviews are recommended to keep the game's development on track with its design objectives.

1.1 Game Summary

Project Little Guy is a puzzle game that is specifically designed for the Tilt Five technology.
We plan to make a Captain Toad-inspired Co-Op puzzle game where two players take control of two characters that are trying to solve a puzzle together. Then they must both use information and tools from each other by working together and learning the area and puzzle to find the solution.

1.2 Game Pillars

- **Teamwork**
- **Spatial Reasoning**
- **Rounded**

1.3 Scope

To keep the project at a reasonable scope we will most likely be only tackling one level at a time. Trying to get the basics completed first on a project that does not even use the Tilt Five technology at all. Then once we have that sorted we can start slowly transitioning the technology into our project and continue expanding with more levels.

1.4 Inspiration

- Captain Toad
- We Were Here
- Degrees_of_Separation
- Fireboy and Watergirl

- Unravel Two

1.5 Project Goal

Our goal for the project is to have a game that showcases the capabilities of the Tilt Five hardware while also showcasing our ability to learn new innovative technology and its application in game design and development.

2. Technical Design

2.1 Project Overview

Overview: Co-op 3D Puzzle Game. Very similar to Captain Toad in terms of mechanics

Prototype: Normal Controller Twin Stick
- Left Stick: Move Little Guy
- Right Stick: Move Cursor/Move Camera
- Some Button: Toggle Cursor/Camera mode

Game Engine: Unreal Engine 5.3.2
Version Control: Perforce

2.2 Engine of Choice

We have decided to use Unreal Engine 5 due to our team wanting to learn more about the engine while having just enough resources to work with it since the Tilt Five technology comes with an Unreal Engine sample project.

2.3 Development Requirements

Aspect	Tool/Software
Engine	Unreal Engine 5.3.2
Documentation	Google Docs
Programming Language	C++
Communication	Discord

Source Control	Perforce
Art	Blender and Substance Painter
Level Design	Paint

2.4 Project Structure

2.4.1 File Naming Rules

File Type	Naming Convention
Static Mesh (Models)	SM_ExampleModel
Skeleton Mesh (Character models)	SKM_ExampleModel
Skeleton (Rig)	SKEL_ExampleModel
Animations	ANIM_ExampleModel_Animation_Variant
Textures	T_ExampleModel
Base Color Textures	T_ExampleModel_BC
Normal Textures	T_ExampleModel_N
ORM Textures	T_ExampleModel_ORM
Materials	M_ExampleMaterial_diff
Material Instances	MI_ExampleMaterial_diff
Blueprints	BP_ExampleBlueprint
Levels	L_ExampleLevel
Widget Blueprints	WB_ExampleWidget

2.5 Feature List

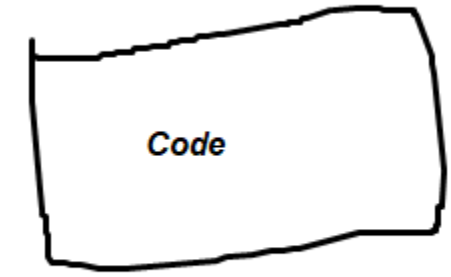
- **Base Features**
    - Player Movement
        - Grabbing Objects
        - Pushing Objects
            - Objects that can only be pushed by both players at once
            - Squish Players if under falling blocks
        - Climbing Ladders
    - 2 Player CoOp
        - Ability to push each other? [TODO: Decide]
    - Interactables
        - Objects that can be activated/deactivated by clicking on them with the cursor
            - Floating Platforms
            - Buttons
            - Ladders
            - Pushable Boxes
    - Ability to Highlight Interactables
        - Makes the objects shine to indicate it
    - Collectibles [Stretch Goal]
        - Placed in more challenging sections of the main level
        - Collectible Tracking in Level Selection
    - Walls becoming transparent when character enters inside an area
        - (only that player is able to see it)
    - Pick up other Player and throw them
    - Player Respawning
        - Player Respawns at the nearest checkpoint
    - Pause
    - Level Selection
- **Non-T5 Version Features**
    - Camera Controls
        - Ability to turn the camera around the main environment 360 degrees
        - Maybe some slight pitch rotational controls so the player can look on top or below a space
        - Zoom
    - Split Screen for the two players
- **Stretch Goals**
    - Characters in separate areas and swap sides mid-level

- Each player having unique abilities
- Add some flavor interactions in the environment to add more life to the game (example: use wand on a pond to have a fish jump out, etc.)

2.6 Software Architecture

- Player
  - Respawn System
  - State Machine
- Floating Platformers
- Buttons
- Pushable Blocks
- Transparent Walls
- Level End Goal
- Wand Interactables
- Ladder
- Signposts

2.6.1 Class Diagram



2.7 Development Plan

Our main plan is to first develop one contained level that contains all of our mechanics and showcases them well in a contained format. Then work backwards and have more levels that introduce the mechanics slowly so the player understands. Then if we have time we can make more challenging levels.

3. Game Design

3.1 Player Movement

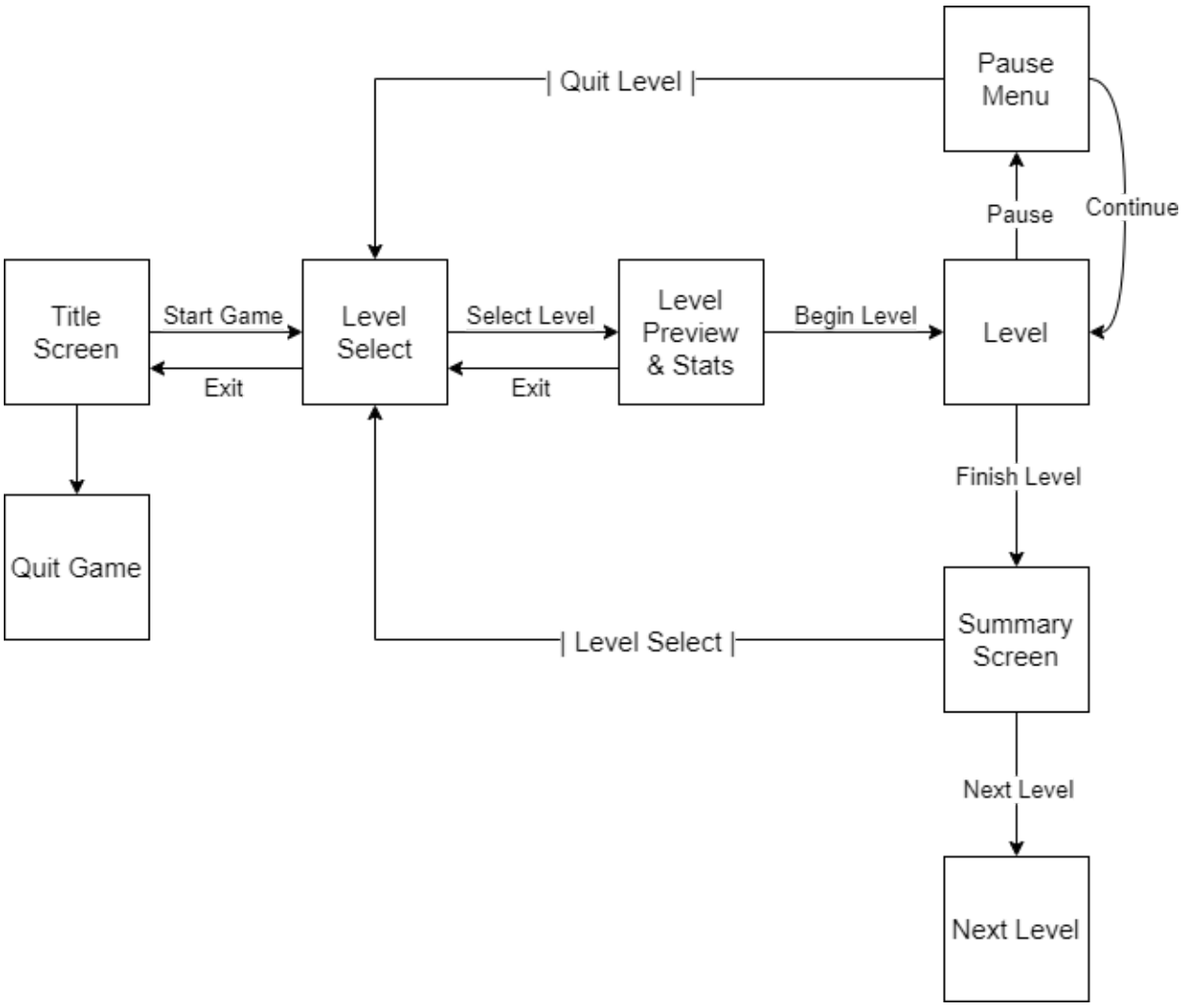
Player movement is relatively simple, keeping it standard omnidirectional movement and making it so the player can not jump but the player can traverse up ramps and platforms and can fall off edges to get to lower elevations.

3.2 Puzzle Design

For our game we want the puzzles to showcase the power of the Tilt Five technology by having most of our puzzles based on spatial and perspective challenges. Both players will need to work together in order to solve environmental puzzles to progress through the level. The characters will be controlled in a shared space but some puzzles might require each player to split up for a solution.
https://www.getpostcurious.com/post/puzzle-testing-process

3.3 UI/UX

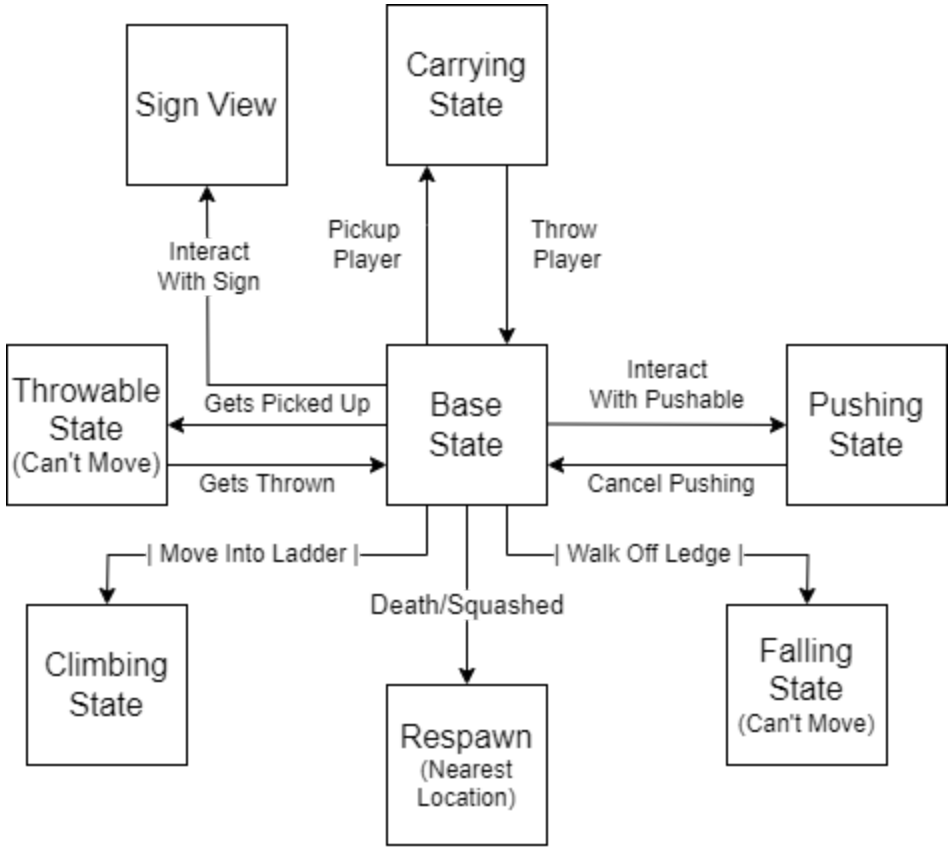
3.3.1 FlowCharts / State Diagram



Menu Flowchart

The diagram above outlines the general flowchart for navigating the menus of the game. The game will launch to the title screen and the player can continue into a level select. From the level select they can choose one of various levels or backout to the title screen. Selecting a level will show a preview of the level to the players and allow them to see any stats such as collectibles they have obtained within the level.

While in a level, the player can pause the game at any time and either continue to unpause or quit to return to the level select screen. When finishing a level, a summary screen will appear letting the player know they finished the level and which collectibles they obtained (Similar to the level preview and stats breakdown). From the summary screen they can either continue to the next level or quit to the level select screen.



Player State Diagram

The above diagram highlights the various states the player can enter within the game.

Base State: The Base State is the general state every player starts in and allows them to move and interact with various game objects. If the player falls into a death zone or is squashed by a falling block, they will respawn at the nearest location to where they died.

Pushing State: Players can push various blocks around the level, so interacting with one of these blocks will have them enter a pushing state. In this state, the player can only move in the axis they are pushing the block from.

Carrying/Throwable State: Players can also pick up and throw each other. When picking up another player, the player being picked up will enter the throwable state where they cannot move until they are thrown, the other player will be in the carrying state and can throw the player in the throwable state a certain distance.

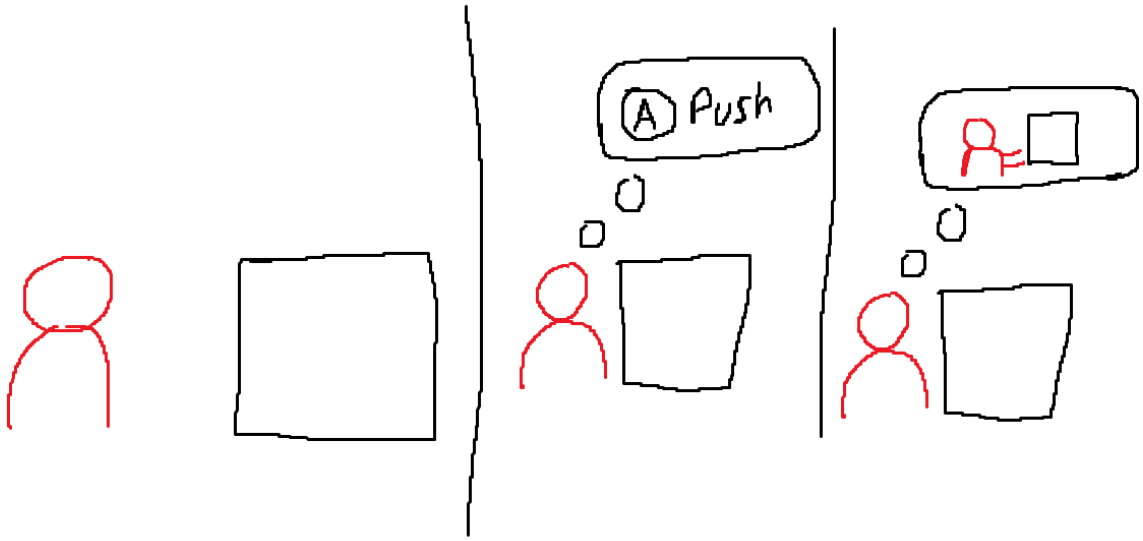
Falling State: If a player walks off a ledge to a lower part of the map, they will be in a falling state where they cannot move until they hit the ground via gravity. This prevents

players from being able to make maneuvers in the air that can skip/break parts of the level.

Climbing State: Some levels will have ladders that can be climbed to reach higher elevations. When a player walks into a ladder, they will enter the climbing state and can only move up and down until they get off of the ladder.

Sign View: Finally, signs will be placed in levels to introduce and explain some mechanics, such as pushing and throwing. When a player interacts with a sign, they will be unable to move and be shown a diagram of what the sign shows on their screen (Player headset specific for Tilt5).

3.3.2 UI Sketches



Contextual Interaction UI

3.4 Player Object Pushing

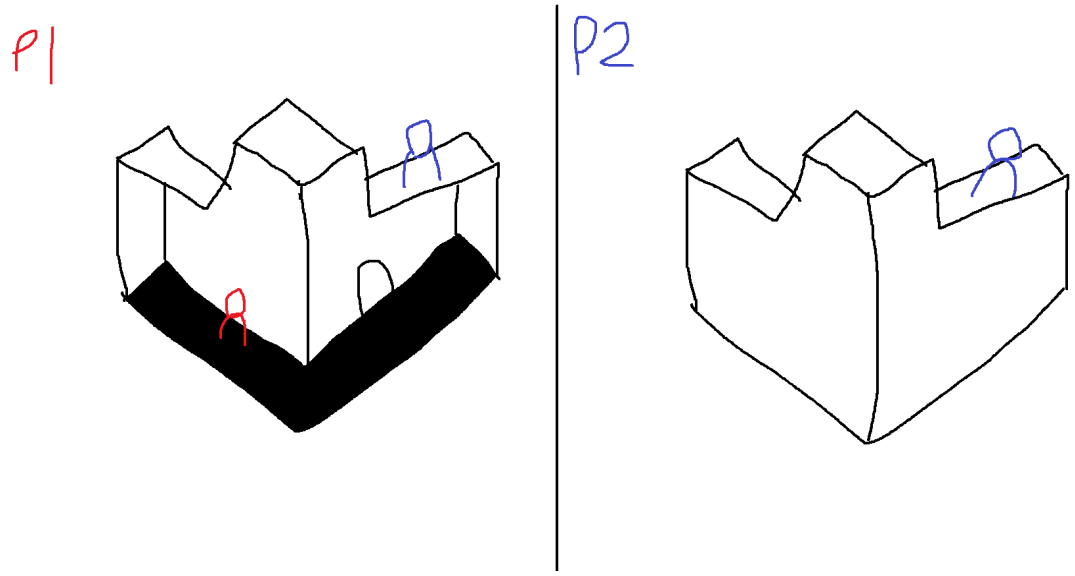
If we want the player to have the ability to push an object like a box around in a level we will need to do it in a certain way for our controller prototype to keep inside the limitations of the T5. Due to the limited controls the best way to handle pushable objects would be to have the player enter a pushing stance upon interacting with the object. Any movement done in this stance will move the object as well and when the object is interacted with again they stop pushing the object.

3.5 Player Throwing

Players should be able to pick up the other player and throw them in an arc. This will allow for unique puzzles where players need to get to a higher elevation and allows us to have some level of verticality in our environments without the need for a jump interaction. Also throwing people is fun :)

3.6 Transparent Walls for Inside Areas

When a player enters a section that is supposed to represent the inside of an area the player camera that is actually inside the section will make the exterior walls transparent to showcase the insides while leaving the other player's walls the same. This is so each player has a unique perspective and adds to the challenge of players having to communicate specific aspects of the room.



3.7 Block Mechanics

The game will feature blocks that can be pushed by the players in order to solve puzzles and gain access to new areas. Some blocks can be pushed by one player while some blocks will require both players in order to move.

Blocks can be pushed off of ledges in order to build new paths across gaps and cliffs or to stack blocks on top of each other. Stacked blocks can be ridden by one player and pushed by the other to gain access to new heights.

3.8 Player Respawn

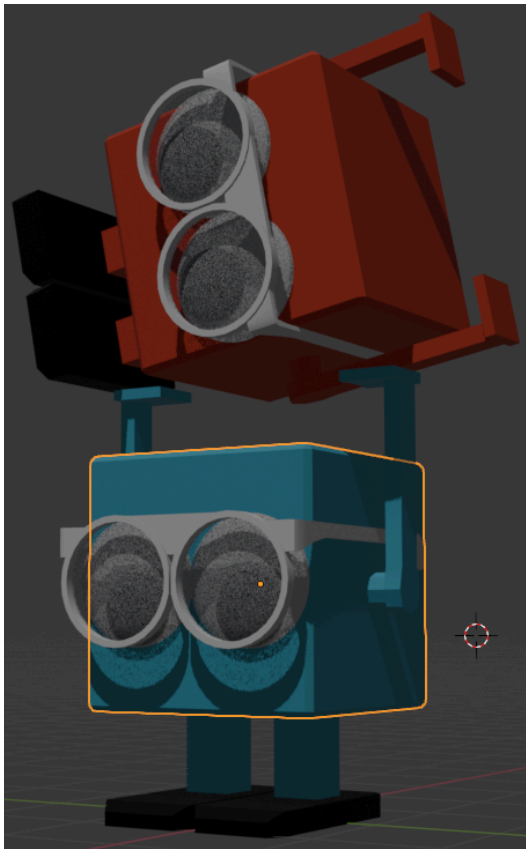
When players traverse a level they will hit invisible checkpoints. If the player dies they respawn at the previously hit checkpoint. If there is none then they will just spawn back at their starting location.

4. Visual Design

The style is a very simplistic blocky and low poly visual style with solid colors in order to maximize visual fidelity. Colors for the characters and important puzzle assets are vibrant where the rest of the environment is more dull in order to increase the visibility of the main puzzle and player.

4.1 Character Design

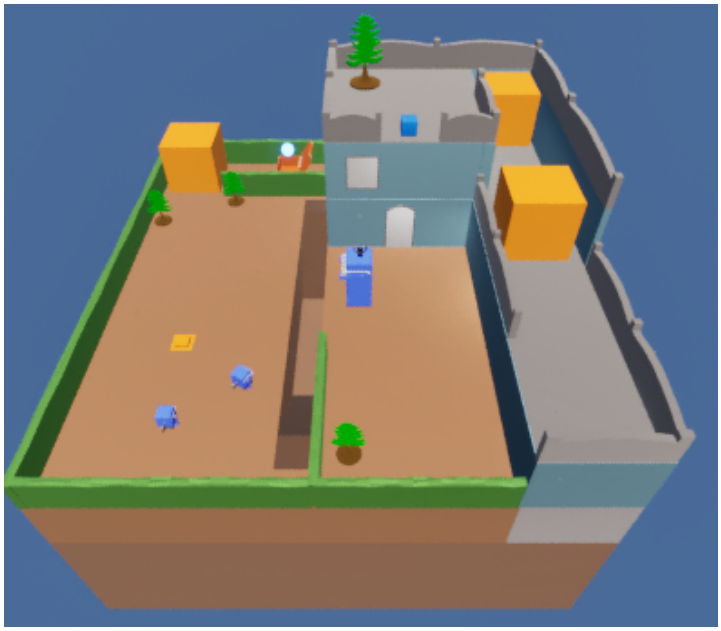
Characters are designed to be small silly wind up toys. They are whimsical little guys. They are blocky in order to simplify their silhouette and to give a cute vibe to them. They are either blue or red (depending on if its player 1 or 2) with black shoes and large white glasses which are colors that aren't as prevalent in the environment. Below is the hexcode for the player blue and red.



365DEB	E53124
RISD Blue	Chili red

4.2 Environment Design

Environments are clean and assets are designed with the puzzle and maintaining clarity the whole way through. Models have a simplistic toy look similar to the character models . Textures are mostly solid colors with very slight variations or markings when needed (like the grain in planks). Hallways and tunnels are at least a head taller than the character model and just as wide to ensure that players can see the player character at all times and move their player without any visual obstructions unless otherwise needed by the level's puzzle. Levels are created in a rectangular prism shape containing everything needed for the level with some small asset just outside of the prism to break up the harsh shape a bit.



4.3 Art Asset List

Character based assets:
-player 1 character
-player 2 character

Mechanic based assets:
-Ladder
-Button
-moveable block
-openable door

Environmental assets
-wall variations (with and without windows)
-doorway variations (with and without windows)
-ceiling variations (with and without windows)
-bush variations (various sizes)
-tree variations (different species)
-garden wall variations (various sizes)
-toy table
-toy chair
-toy bed
-toy chess
-race track ramps
-toy volcano

5. Audio Design

5.1 SFX Design

5.2 Music Design

5.3 Audio Asset List