Piia Laine

Testing Document

Unity Game

Contents

[0](#_Toc46917498)

[1. Expectations and Goals 2](#_Toc46917499)

[2. Successes 2](#_Toc46917500)

[3. Problems 2](#_Toc46917501)

[4. Research 3](#_Toc46917502)

[5. Website 3](#_Toc46917503)

[6. Summary 3](#_Toc46917504)

# Player Movement

The player has two options of moving.

The first is normal walking. While the player is walking, I wanted to make it so that the player doesn’t walk faster when walking diagonally. This was done by using “Input.GetAxisRaw”, instead of just “Input.GetAxis”.

The second moving option is basically the same, but faster. When holding down the space button and moving, the player sprints.

I gave the player a little bit of friction with a physics material, so that it looks more natural, when the player walks against the walls.

# Player Attack

The player can attack, when pressing the K button. This initiates an attack animation.

The actual hitbox is not triggered in a C# script, but rather recorded in the animation. Basically, I have the hitboxes turned off by default, but on the first attack animation step, the hitbox gets turned on and then turned off again at the last animation step.

The hitbox is a trigger zone that can activate switches and break pots, if it gets in contact with one of those objects. It also triggers a sound effect, when hitting either a switch or a pot.

# NPC & Signs

The NPCs and signs work the same.

They stand still and have an actual box-collider2D and one extra box-collider2D that works as a trigger zone. When the player enters that trigger zone and presses the M button, they can either speak with the NPC or read the sign.

# Pushing Objects

Pushing objects can be seen similar to NPCs, just more complicated, because it has 4 trigger zones and when you step into one of those trigger zones and hold down M, while walking in the direction, it can glitch into a wall.

To prevent that, I had to set up so that the opposite trigger zones detects a collision and then turns off the canMove variable, which stops the object from moving into that direction.

# Switches

Switches are quite easy, but still require a little bit of thought setting them up.

When the player or an object is standing on the switch, it should stay activated for as long as someone or something is standing on it.

In my case, the switches either make something appear or disappear in the world.

I made a switchActive variable that is true, when someone is standing on it, which then sets the target object to active and makes it appear.

However, for the case of wanting to make something disappear, I made a public bool variable called reversed. If that is set to true, then the target object disappears, instead of appearing.

# Tile Maps

I tried out quite a few different things for the tile maps.

At first, I tried Unity’s tile map system, but then found a guide online for Tiled, which is a simpler option to make tile maps, so I decided to use that.

However, Tiled has an odd tearing problem, so I decided to change back to the Unity map, because it doesn’t have that issue.