*Stijn Verstraete*

*Platform Development 3*

*2018-2019*

*2DAE11 N-IGP*

Design Document:

Platform Development 3 End Assignment

Theme:

A third-person shooter where the player has to climb a building and go through a parkour, before taking out enemies in a larger area.

Table of Contents

Core Mechanics 3

Player Mechanics 3

Movement/Camera 3

Action 1 – Ledge Jump 4

Action 2 – Shooting 5

Environment Interactions 6

Pushing Block 6

Jumppad 7

Weapon Pickup 8

Navigation & AI 9

Idle/Roam 9

Search Cover 10

Shoot 10

Extra 10

Flowboard 11

Level Blockout 13

Core Mechanics:

**Player Mechanics:**

**Movement/Camera**:

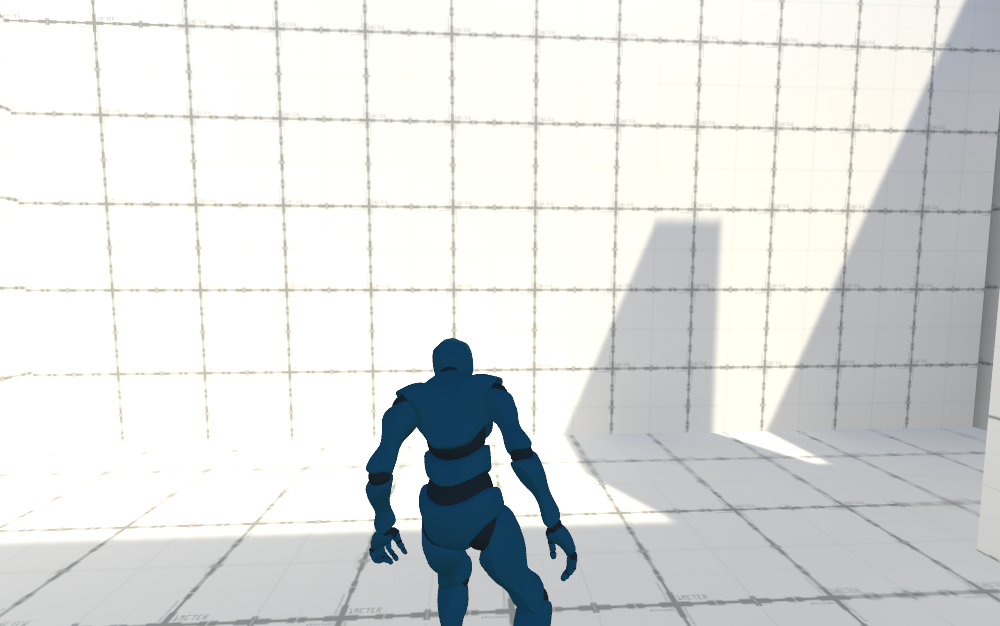
The player walks around in a classic manner: he can run at a normal pace and sprint. Walking backwards and sideward is possible. Movement is relative to the camera: if the character runs forward, he turns in the direction the camera is facing. The player can be moved using the left joystick.

The camera itself is in a third person perspective. It can be moved using the right joystick, rotating around the player.

**Implementation Details:**

A character controller is used. The character uses a run speed based on an acceleration of 1.5m/s² and a sprint speed is based on a 3m/s² acceleration. The character does not use drag physics and regular acceleration physics, but instead has a speed that is calculated based on realistic acceleration.

The player turning in the direction the camera is facing only happens when there is forward movement.



**Action 1 – Ledge Jump**

The player can jump to a ledge above him and pull himself up. This happens by pressing the “A” button while standing below the ledge he wants to jump to. The player then enters the hanging state and is able to pull himself up simply by pressing the left joystick forward.

**Implementation Details:**

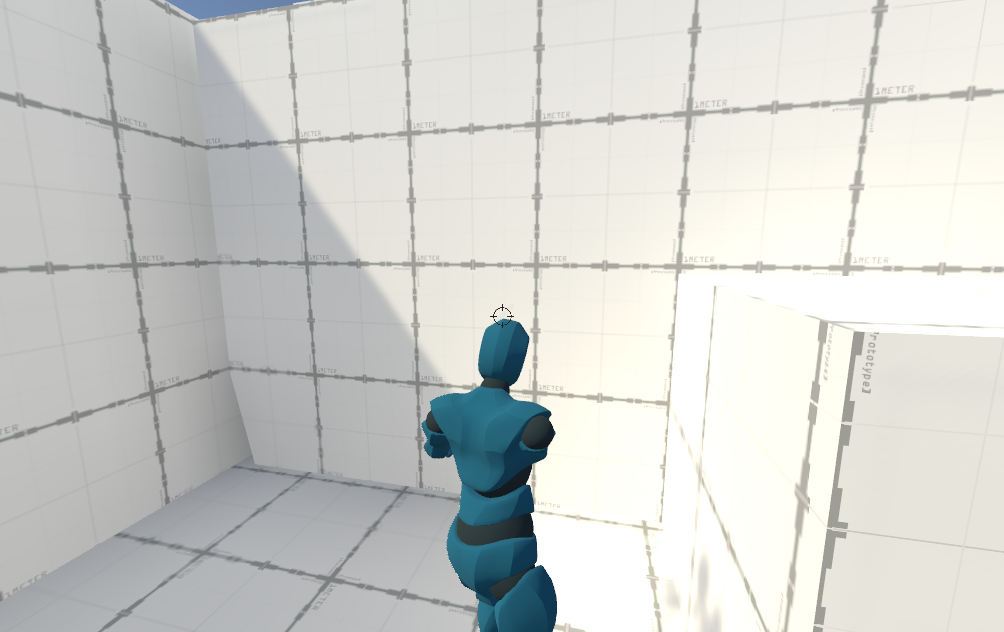
Ledges that can be jumped to are always two meters high. Each ledge has a trigger box underneath it, to check if the player is able to reach it. If it can be reached, the player is snapped to the correct position. When he pulls himself up, an animation is played with an animation event being used to put him in the correct position once the animation has finished playing.



**Action 2 – Shooting**

The player can find a weapon to shoot enemies. Shooting happens when the player is aiming (by pressing down the right joystick) and presses the right trigger. When the player is aiming, a crosshair becomes visible. The trigger has to be tapped to fire a shot and can’t be held down. There is no limit in the amount of bullets. While aiming, the player always looks in the direction the camera looks, regardless whether he’s moving forward or not. A small muzzleflash is displayed when the gun is fired (using a simple sprite).

**Implementation Details:**

IK is used to make the player aim up and down. Player rotation is fixed to camera rotation while the player is aiming. Player moves slower than usual when aiming, things like sprinting are disabled.

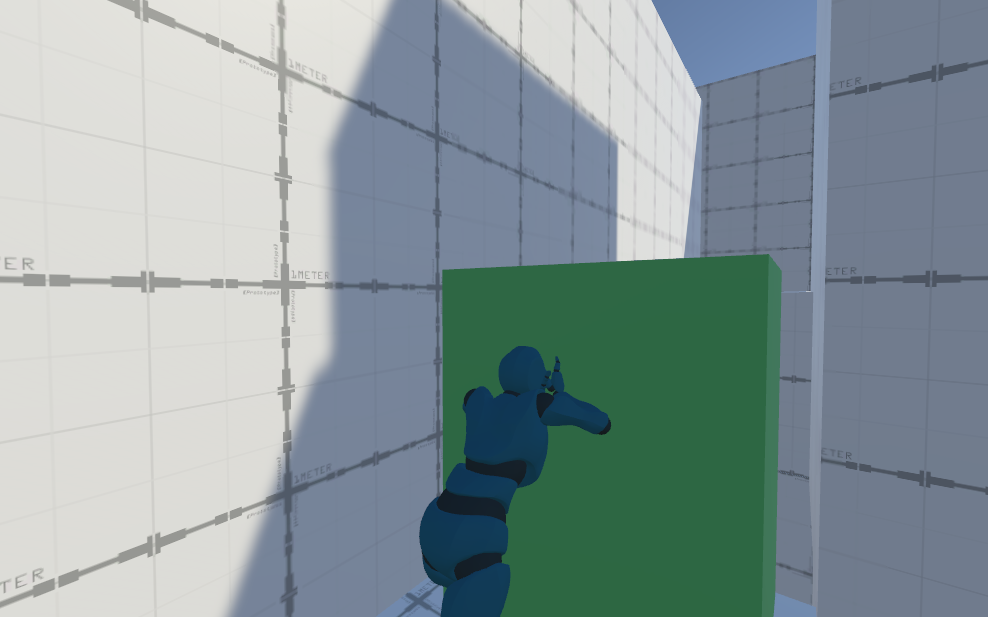
* + **Extra**: The player dies when he gets shot by the AI 5 times or more. Every time the player gets hit, a little animation plays.

**Environment Interactions**

**Pushing Block**

The player can push blocks around to add a little puzzle element to the game. Blocks can be pushed into holes in the ground to fill them up and allow the player to traverse them. A block can only be pushed from one side. The player can start pushing a block when he presses “A” next to it. Walking away from the block is required to stop pushing it. A block can only be pushed forwards.

**Implementation Details:**

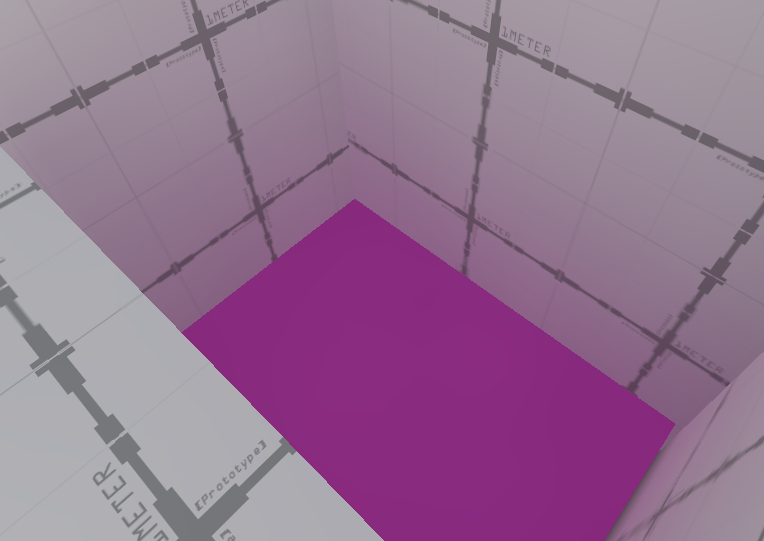
Blocks use rigidbodies. Forces are applied when the player walks forward. While pushing a block, only forwards and backwards movement is possible, with moving backwards being used to stop pushing the block. IK is used to place the hands on the surface of the block.

**Jumppad**

By falling on a jumppad, the player gets pushed two meters into the air. This helps the player with reaching higher ledges, by timing the pressing of the “A” button right they can grab onto it. While the player is in the air, he has no control.

**Implementation Details:**

The following formula is added to the y velocity of the player:

Mathf.Sqrt(2 \* Physics.gravity.magnitude \* \_jumpHeight)

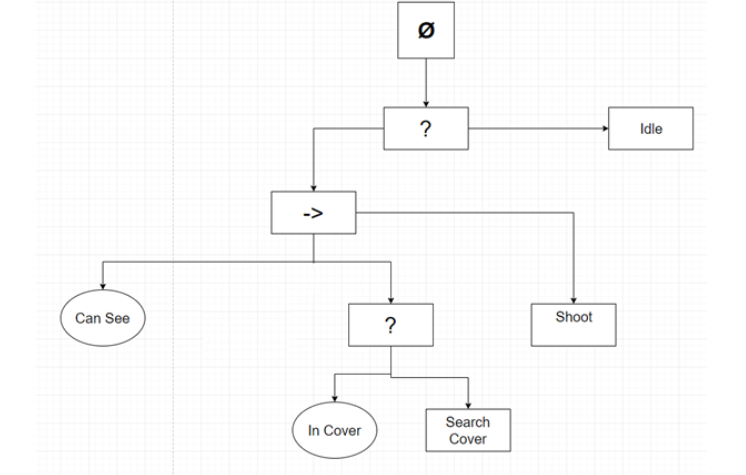
**Weapon Pickup**

A pistol can be found in the level. It can be picked up by standing over it and pressing the “A” button. IK is used here to make the pickup seem smooth. Once the player has the pistol, he can aim and shoot (as described earlier in this document).

**Implementation Details:**

 Animation events are used alongside the IK to snap the pistol to the players hand once the animation is at it’s deepest point. The pistol becomes a child of the righthand once picked up.

Navigation and AI



The AI Walks around in an area. The player has to take as many of them out as possible. The AI can do the following things:



**Idle/Roam**

Roam around when idle. Occasionally he walks to a new position near him. If the player is spotted, all AIs are alerted and they go into combat mode. This means that they will search cover/shoot at the player.

 **Search Cover**

Search cover behind obstacles based on where the player is in relation to him. The AI relocates if there is no cover between him and the player. When searching for cover, the AI animation becomes more low profile, to make him harder to hit.

**Implementation Details:**

A linecast from the player to the AI is used. The information from this linecast is then used to calculate the new position the AI has to get to to be in cover.

**Shoot**

Shoot at the player. If the AI is in cover, he will shoot at the player. If the player gets hit 5 times he dies.

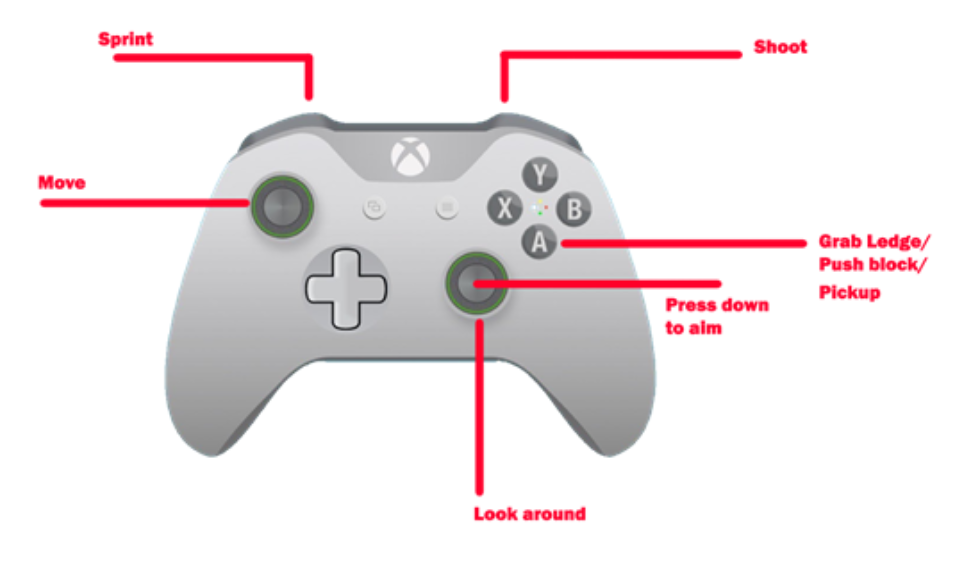
**Implementation Details:**

If the AI hits or not is based on an accuracy percentage combined with a li necast to see if the player is visible.

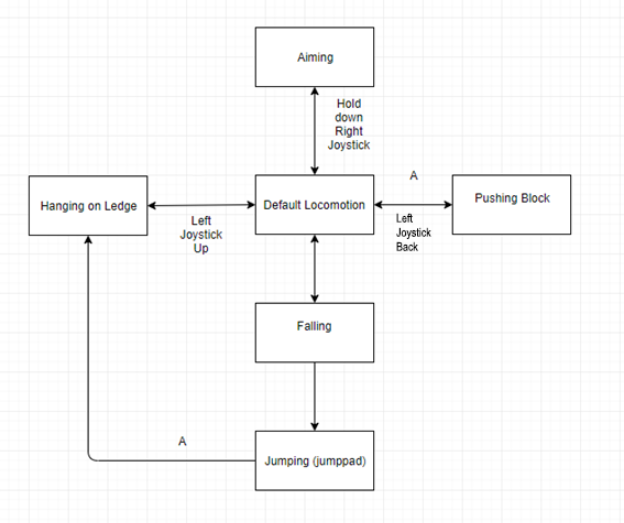
**Extra**

The AI dies when he gets his 3 times. An offmeshlink is present in the navmesh, in the form of a little ditch the AI can jump over. Once one AI gets alerted, all AIs get alerted.

Flowboard

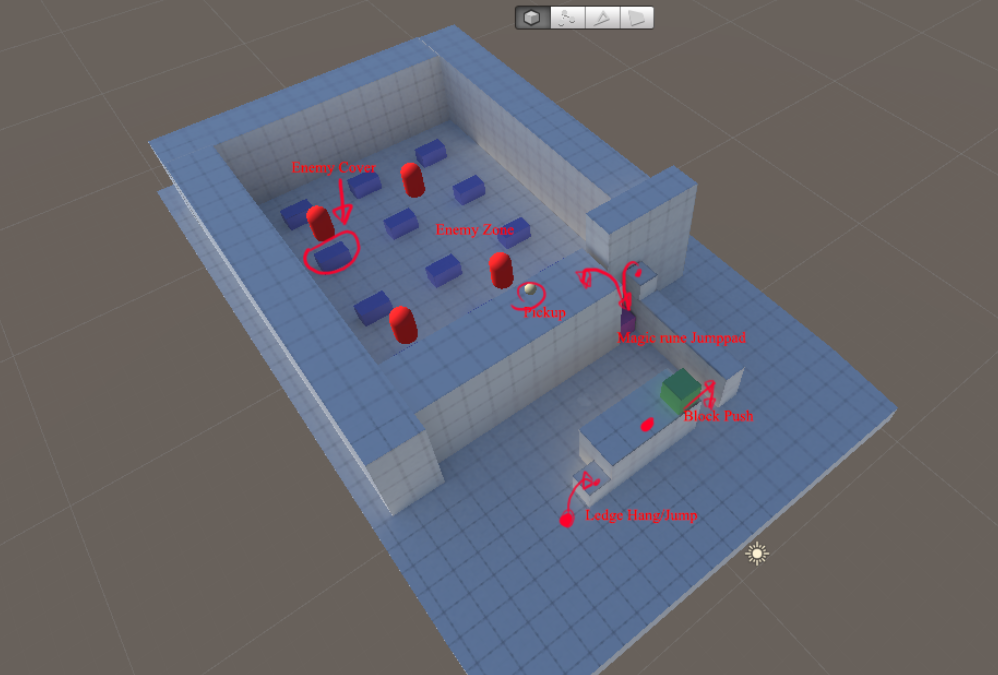


* **Default Locomotion**: the default gameplay mode, the player can walk around. By pressing down the left trigger sprinting is possible. Walking happens with the left joystick, while the right joystick is used to look around. By pressing down the “A” button interacting with the environment is possible.
* **Hanging on Ledge**: While in this mode, the player can push the left joystick forward to pull himself up.
* **Pushing block**: While in this mode the player moves slower. By moving forward the block gets pushed, by moving backwards the block gets released.
* **Falling:** While in this mode the player has no control. To exit to mode he has to land or grab a ledge by pressing the “A” button.
* **Jumping (Jumppad)**: When the player lands on a jumppad, he keeps bouncing until he manages to grab a ledge by pressing the “A” button.
* **Aiming:** By pressing down the right joystick the player enters aim modus, during which he can shoot by pressing the left trigger.



The arrows show the transitions. Further details about what button has to be pressed are next to the arrows.

Level Blockout



Enemy Cover: blue

Enemies: rood

Jumppad: purple

Pushable Block: green

Extended version

