

# Documentation

---

## Controls:

WASD to move, Mouse to turn and look around

T to transition camera from first and third person

P to load the Level 1 mesh object into the scene and move the player into the position

## To know:

- the enemies will follow you and once they get to a close enough distance they will shout at you and get big then start attacking
- the tetrahedrons can be picked up, as of now, they are transformed to another random position.

## Intro Screen

I implemented an intro screen by creating image textures in photoshop for all the views in the menu (main, help). I then created a thin cuboid and placed it into the scene and fixed the camera to display only the menu. The menu is interactable using key presses. So if the player **presses 1** it will start the game, if they **press 2** it will change the menu object's texture to the help image, and they can press **escape** to exit the game. I then set the camera behind the player and call the player's `updateCamera()` method.

In the help menu it displays the keyboard/mouse controls.

I would like to add particle effects to spice up the main menu screen; I would like to create fire ember particle effects.

## Primitive Object

I used the tetrahedron object created in the tutorial and added an initialize method to initialize multiple of them in the scene. These will be displayed with different textures, totaling 3 different textures. The `initializePrimitives()` method populates a list of primitives, and it takes in scale and amount as parameters making it quite a flexible method.

I have added a rotation/scale transformation to it so that it looks like an item you can pickup (the current placeholder textures are not good as items). I will change the textures to portray weapons/healing items/power ups/etc..

I have made it so the the primitive object is transformed to a different position when the player gets in contact with it. The collision hit box is changed while the power up gets bigger and smaller, making it easier for the player to pick it up when big.

## Skybox and Terrain Textures

I have changed both skybox and terrain textures. I am trying to solve the problem of the texture being stretched over a big shape making it look bad, I would like it too tile so that when applying a seamless/repeating texture it looks high-definition.

## Camera Motion Technique

I have implemented a 3rd person and 1st person view. I have changed the **3rd person** template code by mapping the X mouse movement to turning and the Y mouse movement to tilt the camera behind the player. I still need to make it so that the camera movement is limited and the camera does not get to a crazy high position. I implemented the **1st person** camera by fixing the camera in front of the player object's head. It turns with the player and you can also look up and down.

I also transform the camera to display the menu screen at the beginning of the game.

Known issues:

- the camera has no limit when moving up and down, so moving the mouse vertically can change this
- this also leads to a weird camera position when transitioning between first and third person.
- The camera can clip through objects/ground/level walls.

## Mesh based Object

I have implemented 2 mesh based objects. The first is the minotaur enemy. This is initialised in the main\_layer's constructor with the `initialiseEnemies()` method and is populated into a list `mEnemies`. I then can use `updateEnemies()` method to update all of them at the same time.

The bigger the minotaur, the slower it moves. When it gets close enough to the player, it shouts, increasing its size, and then starts attacking. I have kept the movement speed of the enemies slow at the moment.

When the enemies are not moving and not aggroed, they can be in an idle state. In this submission there isn't a moment where they are not chasing the player; however in the final submission, I would like there to be multiple enemy types that act differently. One can be a shouter that can increase its size, another can be a tracker that chases you down faster than any other type and can call the other enemies over and so on.

There are multiple animations that these mesh based object and multiple levels of transformation (scale and position towards the player). I implemented an `animationHandler()` in Enemy.cpp to change the animation based on what the enemy is doing, this works seamlessly. The enemies also textured.

The second object is the level\_1.dae that I created in blender. This can be loaded in by pressing **P**.

## Goals

I would like this game to be in the same spirit of a Dark Souls game (or its spiritual predecessor Kings Field). This will entail:

- the player having 2 attacks (heavy and light), a stamina amount which is depleted on attacks;
- item pickups. Health, Keys, Power ups; secret walls/passages : that can only be accessed by doing certain actions;
- different enemy types;
- a final boss, etc.

I would like the killed enemies to have a ragdoll effect so the player can interact with them after they have killed them. The player's object will change into a different one than it is now with different animations.

