Third Person Controller

Rishav, Omang

February 11, 2025

Contents

---

1. Introduction

Welcome to the third studio. The studio will be quite different from the previous ones. There will be no instructions or tutorials; instead, you will rely on the knowledge and skills you've gained from the first two studios to guide you. The goal for this studio is to make a basic third person character controller (see Demo Video here).

A third-person controller lets players control a character from a behind-the-back or surrounding perspective, offering a clear view of movement and the environment. Common in action and RPG games, it includes movement, camera control, and interactions.

2. Features to Implement

For your assignment, you will be implementing a simple platformer game with the following features:

- A player that can
  - Look around using a third person free-look camera **(0.5 marks)**
  - Move around in an environment where the forward direction is determined by the camera rotation **(1 mark)**
  - Jump to elevated platforms **(1 marks)**
  - Tune the movement and jump parameters to make a well controlled platformer **(0.5 marks)**
- An environment with a large flat plane that contains
  - boxes to jump onto and invisible walls to prevent falling off **(1 mark)**
  - coins that rotate on top of boxes to collect **(1 mark)**
  - upon collection of the coins, they disappear, and update a score value in the UI **(1 mark)**

You will additionally be required to

- Maintain a proper git history, with frequent commits at various implementation steps with relevant git commit messages **(1 mark)**
- Have a readme with the video, similar to the first two studio submissions **(1 mark)**

There will be two bonus marks awarded for the following

- Implement Double-Jumps and a Dash feature
- Maintain Single Responsibility Principle and a consistent coding style where appropriate logic is grouped together under properly named scripts.

3. Tips and Hints

Everything that you've been taught in the lectures, and Studio 1 and 2 will be required for you to implement this assignment.

- The cinemachine free-look camera should give a good starting point for the third person camera.
- A combination of the instructions for handling movement can be used from both studio 1 (for rolling the ball) and studio 2 (for moving the player)

- To control the "forward" direction of the player, the aiming controls from the bowling game can be of assistance
- To control the friction on the ground and slow the player down, physics materials can come handy.
- Jumps can be enabled by checking if the player is making contact with the ground before adding an upward force. This can be done via OnCollisionEnter and a CompareTag to see if the player is touching the ground, and then OnCollisionExit can track when the player has left contact with the ground.
- By exposing the movement and jump forces in the inspector, and changing the friction forces, the overall feel of the character can be tuned to feel well in control rather than floaty.
- Designing the environment should be easy at this point with the lectures and the first two studios. It just needs a few boxes scattered around which the player can jump onto. Beyond that is up to your creativity.
- Importing a coin asset will be similar to how other assets were imported for the second studio and during the lecture. Setting up a coin to rotate by itself will also function similarly to the earth model rotations from the lectures, and collecting it will utilize OnTriggerEnter collisions, at which point they will be destroyed and increment a score value.
- The score system can utilize a simplistic UI similar to the second Studio assignment. It just needs to reflect the total number of coins collected in the game at any given point in time. Perhaps a Game Manager can come in handy.
- Finally, as and when you make these features, be sure to commit them with the relevant git commit message. A good git history will allow you to easily "reload to a previous save point" if your current project gets completely destroyed. See the previous two studio guide hints on good times to make a commit. You can never really make enough commits, so feel free to commit buggy features as well which you indicate are "incomplete" or "not working" and then address it later on ("X works now").

Here is a demo video which showcases all the relevant functionality for reference. The first 21 seconds showcase the main functionality, and the rest of the video showcase the optional bonus functionality. Note your environment doesn't need to be the same, but should have a layout that allows the player to move around and jump on stuff. You can use more models and assets for the environment, the player, etc, but the coin object is the only thing that needs a custom asset.
Good luck with your development journey, and don't hesitate to reach out if you have any issues or troubles!