

PlayerController.cs

Overview

The player's movement controls in the game world. This script is part of the Player prefab object and is designed to be as modular as possible. This script requires the GameManager script to function properly and is simple to setup in the game project (see the Setup section for implementation details). All player-movement code and logic must be placed here to avoid coupling.

Setup

1. In the event that the script is missing from the player object/not using the original prefab object, add the script to the player object.
2. In Ground Detection, find the Ground Check object. Select the transform of the Ground Check object. In the event this object is absent in the player object, this transform is a child object of the player and is positioned at the bottom of the player's model.
3. To enable movement, create an empty game object called: GameManager. Then attach the GameManager script to it and set the Game State to START.
4. To enable jumping, set the Ground Mask to Ground. In the event a Ground layer doesn't exist, feel free to create one. Make sure to set any jumpable terrain and objects to the Ground layer.

Key Variables

Ground Check:

- The transform of the bottom of the player's model.

Ground Mask:

- The layer(s) that allows the player to jump. Multiple layers can be set as jumpable, if necessary.

Methods

-Awake():

- Initializes the player's input and attached CharacterController.

-Update():

- Checks the GameManager's current game state every frame to see if the player is allowed to move.

-HandlePlayerMovement():

- Handles the player's movement in the following order of method calls: CheckPlayerMovementInput(), CheckGround(), MovePlayer(), and PlayerJump().

-CheckPlayerMovementInput():

- Handles the player's initial controller input for movement. The player's current speed is updated here.
- TODO: Handle blend-tree animations in this method.

-CheckGround():

- Check if the player is grounded. The public boolean variable IsOnFlatGround is set here and can be used in other classes if necessary.

-MovePlayer():

- Applies a new 3D movement vector to the player using the attached CharacterController.

-PlayerJump():

- Check if the player has pressed jump and is able to jump.
- Applies a vertical change to the player's y velocity vector using the attached CharacterController.

-PlayerMoveAndRotation():

- Handle the player's movement input relative to the camera's current view position.
- Calculates the player object's rotation.

-OnEnable():

- Allows the player's input to be used.
- Note: Currently unsure as to why this method is necessary as the GameManager's game state should be handling this logic. As of now, it is required to enable the player's input.
- TODO: Move this logic to the Awake() method.

-OnDisable():

- Note: This method will be removed in the future and be handled by the GameManager's logic.