T1 New Mechanic Hints

Customizing Roll-A-Ball

The Tutorial 1 assignment requires you to implement several new features to the completed Roll-A-Ball tutorial. The following hints should expedite your efforts.

Jump Mechanic Hints

Add a "jump" mechanic to the game, where the ball will hop when the player presses the space bar. The ball should only be able to jump when it is on the ground (touching a wall is also ok).

- Remember how to detect a button press from the player?
- If you select your player gameobject, you should see several components in the inspector window. Which one is most relevant to physics, and thus relevant to velocity? Do you remember how to get a component so you can mess with / change its values?
    - Exploring the functions and properties of a component is much easier if your code completion / intellisense is working in your IDE. **It's worth your effort to get this working.**
- How do we know if the player is "on the ground" and thus able to jump?
    - Could we listen for / detect collisions with the ground and adjust a "is_on_ground" bool?
    - (advanced) Alternatively, could we look beneath us to check if some ground is there?

Fall-Reset Mechanic Hints

Add a falling-reset mechanic, where the game restarts if the ball falls too far underneath the level (items get reset, the player gets reset-- all of it resets).

- When the player gets too low in the world (they're falling), we want to reset the scene. How do you know what the player gameobject's current height is? If you look in the inspector, which component has information relating to height or Y-position?
- In order to perform a scene-switch in Unity, we can use the SceneManager.LoadScene() function. This is typically used for "going to a new level". What happens if you attempt to load the scene you're already in?

Custom World / Layout Hints

Add additional floors, walls, collectables, and move everything around to create your own unique "Roll-a-Ball" level (please only submit this level). It doesn't need to be a large level, just different than what the tutorial leaves you with.

- If you add additional collectible blocks, don't forget to change the required count for victory in PlayerController.cs
    - Ask Yourself : Is it a sign of bad code if a designer must remember to update a script every time the number of collectibles changes? Can you think of a way this could be automated?
- Control+D / Command+D will duplicate the currently selected gameobject, like so.
- If you hold the control key while moving a gameobject around, it will "snap" onto a grid, allowing for easier alignment, like so.

- Using your left hand, the “w”, “e”, and “r” keys at your fingertips will switch between translation, rotation, and scaling modes quickly-- expediting your level-design workflow like so.