UNITY GRAVITY CONTROLLER.

Rigidbody-based Gravitational Attraction Towards Objects.

USER MANUAL

Version 1.0.0.

KENN JONATHAN PRINS, 2021

Table of Contents.

1. QuickStart.

To test the out the features provided by the system right away:

1. Download and import the UnityGravityController package into a new Unity project.
2. Open one of the demo scenes named as "Sample scene" or "Demo" under Assets/Scenes/SampleScene or Assets/Scenes/Demo.
3. Press play and try out the example character and gravity areas.
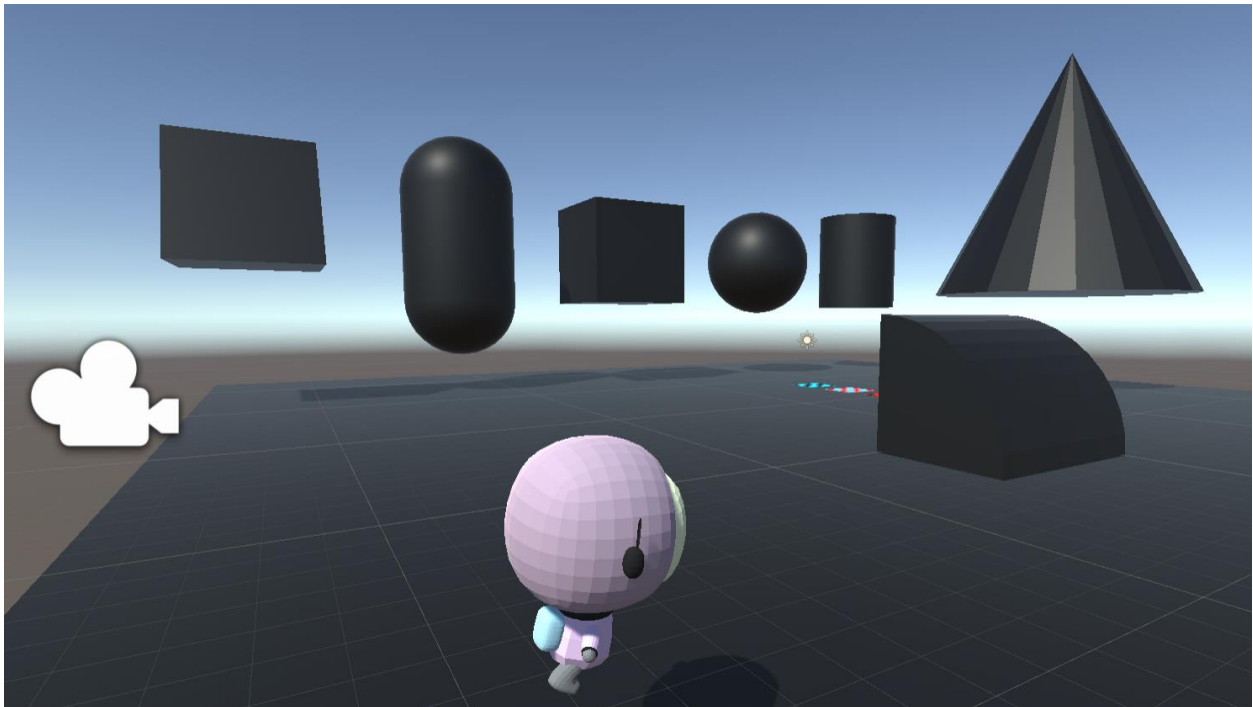

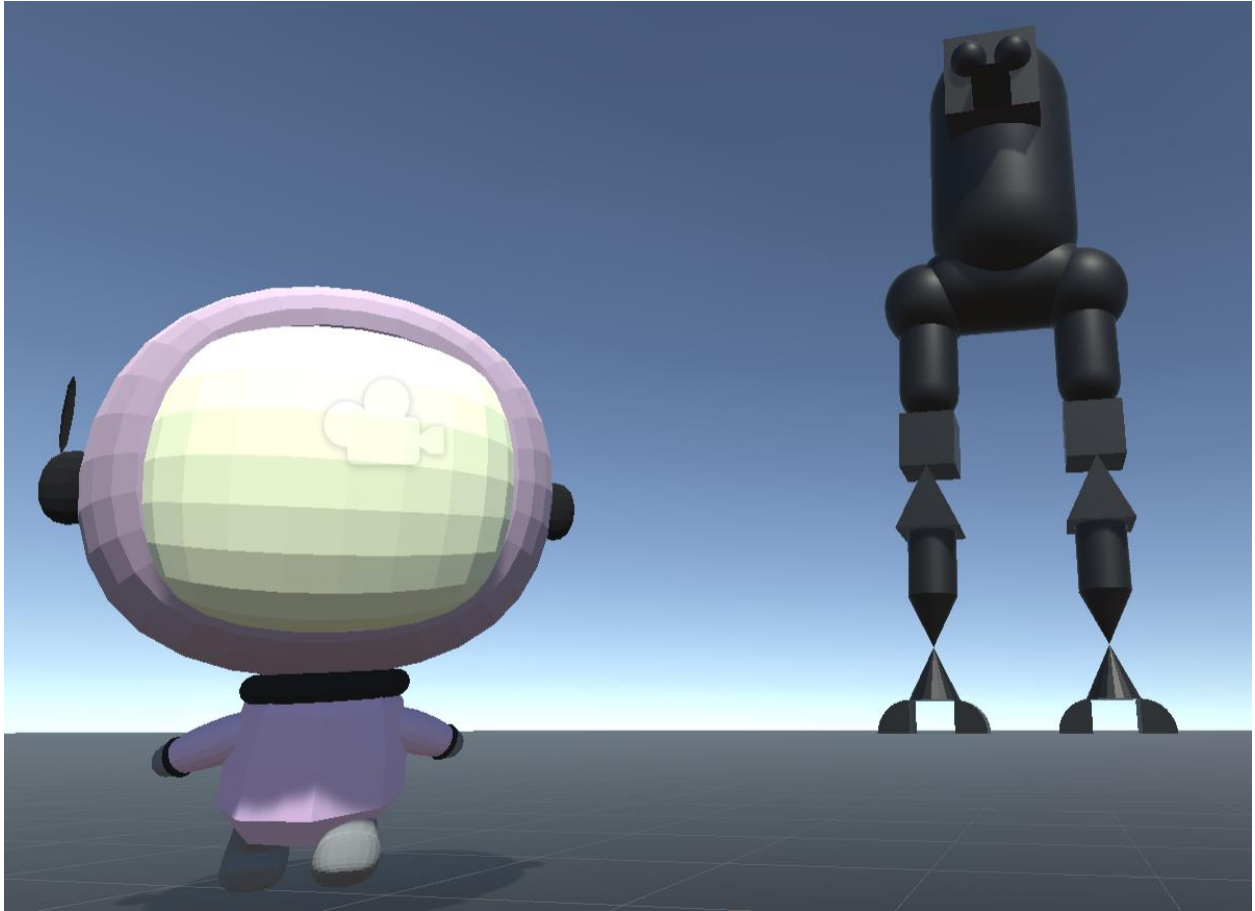
Figure 1: QuickStart - SampleScene

Figure 2: QuickStart - Demo scene

The following inputs are set as key-bindings for the respective player movement actions:

- Move – Keyboard (W, A, S, D, or arrow keys), Gamepads (Left stick)
- Jump – Keyboard (Space), Gamepads (Button South)
- Crouch – Keyboard (E), Gamepads (Button West)

2. Prefabs.

Prefabs for characters, environmental gravity objects and movement multiplier platforms are available under Assets/Prefabs.

2.1.   Character Prefabs.

Two character prefabs are provided under Assets/Prefabs/Characters.

- **Animated character prefab** :
  - This character has the following components attached :
    - Transform.
    - Rigidbody.
    - Capsule collider
    - Animator.
    - Player Gravity (Script)
    - Player Animations (Script)
    - Movement (Script)
  - The animator component and the Player Animations script component is responsible for handling the animated characters animations based on user inputs.
  - The Player Gravity string component is responsible for adding force to the animated players rigidbody towards direction of gravity.
  - The Movement script handles all movements of the player according to the inputs from the users.
- **Empty character prefab** :
  - This character has the following components attached :
    - Transform.
    - Rigidbody.
    - Capsule collider.
    - Mesh Rendered.
    - Capsule (Mesh Filter)
    - Player Gravity (Script)
    - Movement (Script)
  - The empty character prefabs behave in the same manner as the animated character.

o   The difference between the two objects is that no custom model or animations are set to the empty character prefab.

o   The empty character prefab is provided for ease of use for the developers purchasing the asset to adjust the character as they want to.

2.2.   Environment / Gravity Area Prefabs.

Eight environment / gravity area prefabs are provided under Assets/Prefabs/Environment. Each of these prefabs consists of two game objects. The parent object defines the bounds of the gravity area and has a type of collider component and a script which is used to identify the direction of gravity. A child object is attached to each prefabs gravity area game object which is set at the ground object onto which the player is to be attracted. The child object has a mesh filter, a mesh renderer, a type of collider, and a material.

The environmental / gravity area prefabs available on the asset are as follows:

- Sphere gravity.
- Cube gravity.
- Capsule gravity.
- Cylinder gravity.
- Cone gravity.
- Prism gravity.
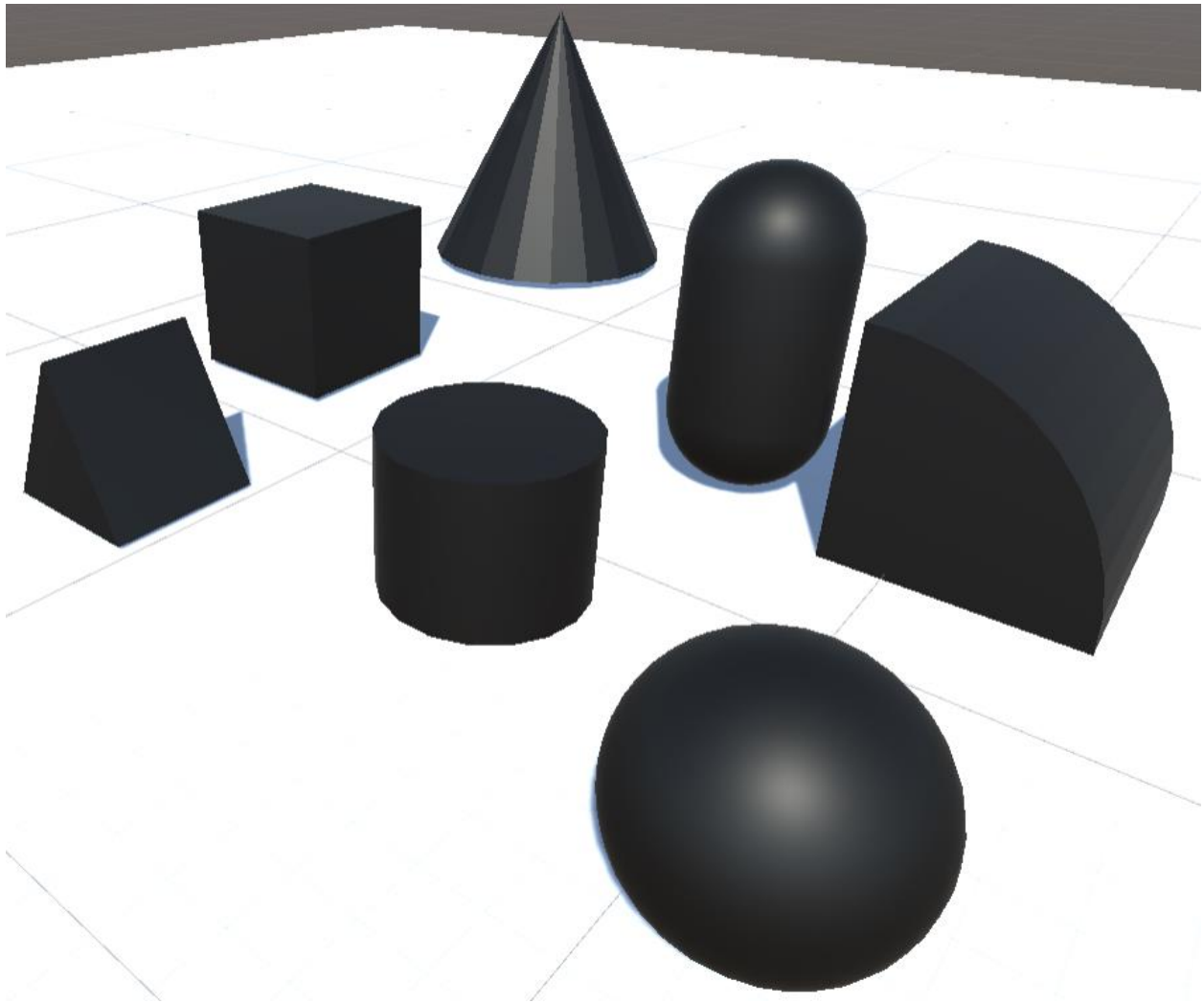- Plane gravity.
- Arch gravity.

2.3.   Platform Prefabs.

Platform prefabs are game objects in which when the player is colliding / standing on top of the object, the players movement and jump speeds are modified at run time. The platform prefabs available on the asset are as follows:

- Jump Pad : Increased the players jump height.
- Bouncy : The player bounces continuously while on top of the platform.
- Mud : The players movement speeds are reduced.
- Speed boost : The players movement speeds are increased.
- Windy : The player slides off the platform if idle.

3. Scripts.

All scripts are available under Assets/Scripts. The scripts are used to control the behaviors of the game objects in a scene. All scripts are well commented are all functions are explained thoroughly. The scripts which are available in the asset are as follows:

- **Movement :** This script is responsible for handling the players movements based on inputs.
- **Player Animations :** This script is responsible for communicating with the animator component in order to set the correct animations to the animated player game object based on the input actions.
- **Player Gravity :** This script is responsible for identifying the gravity areas that the player is colliding with and identifying the gravity area with the highest priority. Then the script rotates the player towards the direction of gravity and adds the force to the player in the direction of gravity.
- **Gravity Controller :** This script is responsible for identifying the direction of gravity for a given gravity area. This abstract class is extended by the following scripts which returns the direction of gravity of a given gravity area:
  - **Gravity :** This script can be used to apply gravity towards almost any game object.
  - **Inverse Gravity :** Returns the inverse direction of gravity of almost any game object. Can be used when an area needs to repel incoming objects instead of attracting them.
  - **Downwards gravity :** A more efficient alternative for the **Gravity** script which can be used for gravity areas which needs to attract incoming objects towards a specific direction (Upwards, downwards, sideways, etc.)
  - **Spherical gravity:** A more efficient alternative for the **Gravity** script, which can be used for gravity towards spherical objects.
  - **Sphere inverse gravity :** A more efficient alternative for the **Inverse Gravity** script, which can be used for repelling objects away from spherical objects.

4. Character Model and Animations.

The assets provide a custom character model with animations which can be used for player movements. The character model comes with the following components:

- An armature.
- Mesh of the character model.
- Material for handling colors of the model.
- Animations.
    - o T Post animation.
    - o Idle animation.
    - o Walk animation.
    - o Fall animation.
    - o Jump animation.
    - o Crouching animation.
    - o On crouch animation.
    - o Crouch walking animation.