



## Dynamic Parkour System – v1.0

Developer: Èric Canela

### Content

1.	Description .....	2
1.1	Actions.....	2
1.2	Features:.....	2
2.	Setting Up the Project .....	3
3.	Prefabs.....	4
3.1	Player Prefab .....	4
3.2	Setting Climbable Points Prefab (Ledges) .....	7
3.3	Setting Jump Landing Points (Pole).....	9
3.4	Setting Jump Landing Points (Wall).....	9
4.	Contact Me .....	9

## 1. Description:

**Dynamic Parkour System** is a Pack of several already developed systems that allows any character to perform parkour and climbing walls like in Assassin's Creed or Uncharted.

This package is fully customizable with its own Character Controller. Really easy to configure and fast to use. It has the possibility to choose between 8 already build-in Parkour Actions.

### 1.1 Actions

- Vault Fence
- Vault Box
- Vault Down
- Slide
- Climb Obstacles
- Jump to Poles
- Climb Walls
- Climb Ledges
- Jump Ledge to Ledge
- Drop from Ledges
- Braced and Hang Climbing

### 1.2 Features:

- IK System
- Vaulting Obstacles
- Slide Obstacle
- Auto Step
- Climb & Drop Obstacles
- Predicted Jumping
- Climbing System
- Already Build-in Prefabs

**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**

## 2. Setting Up the Project

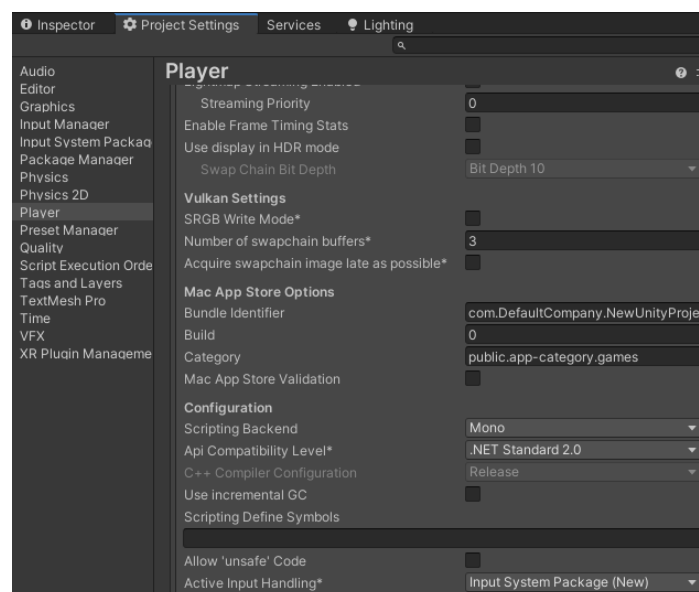
- 1) Import the Dynamic Parkour System package in your Project
- 2) Add the following Tags and Layers in the same order as follows

**Tags:** Pole, Reach, Vault, Slide, Deep Jump

**Layers:** 8: Ledge, 9: Vault, 10: Wall

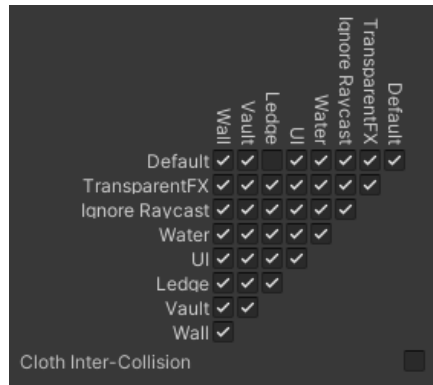


- 3) Import **Cinemachine** and **New Input System** packages from Window -> Packet Manager
- 4) Edit -> Project Settings -> Player -> Other Settings and make sure New Input System is selected



**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**

- 5) Edit -> Project Settings -> Physics and make sure Ledge and Default(Player Layer) are unchecked



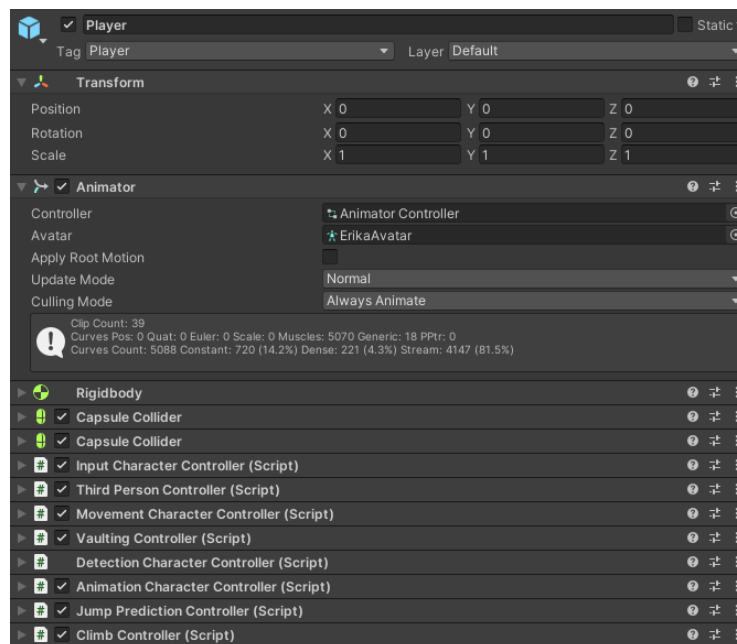
### 3. Prefabs

In Prefabs folder there are already prepared some prefabs of different objects to drag and drop on the Scene and work instantly. Also, the player prefab is already configured to work once placed in the scene.

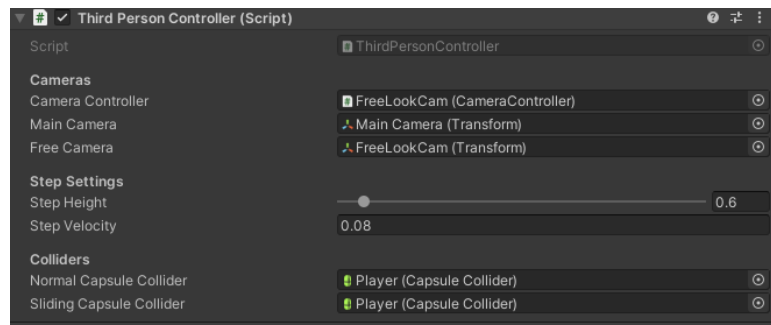
#### 3.1 Player Prefab

**IMPORTANT: All the values in the inspector have been tested and assigned depending on the Erika Model. Feel free to change them if you need to adjust them for your model.**

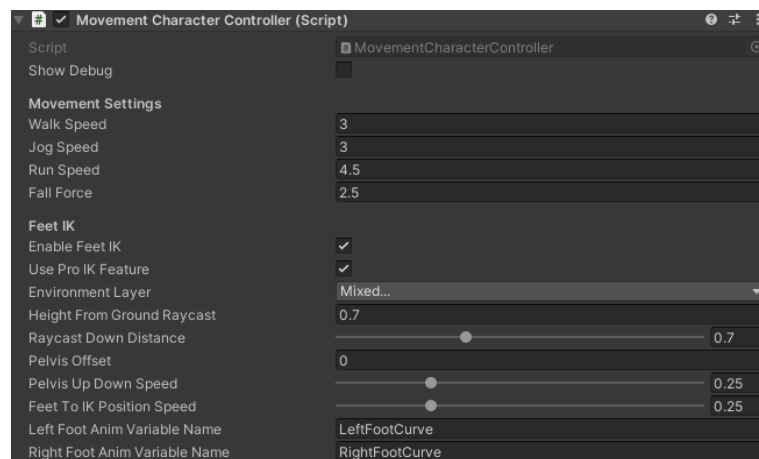
The Player needs to have included the following Scripts and Components. Recommended to use the Prefab Player as a Base and change only the mesh and Animator Controller.



**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**

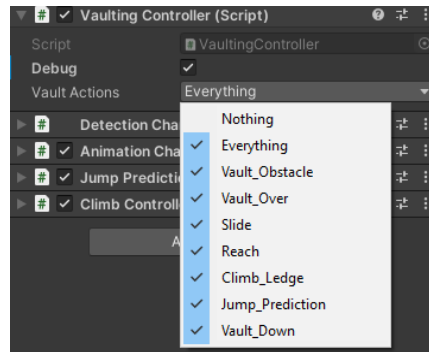


- **Camera Controller:** The Main Cinemachine Camera, in this case a FreeLook Camera, needs to have a Camera Controller.
- **Main Camera:** Main Camera on Scene
- **Free Camera:** The Main Cinemachine Camera, in this case a FreeLook Camera already included in Player Prefab
- **Step Height:** Max Height the player will automatically step up any obstacle
- **Step Velocity:** Velocity the player Will step up.
- **Colliders:** The player has two colliders to interact with physics, the normal is the taller one and the sliding is to collide with walls and slide down.

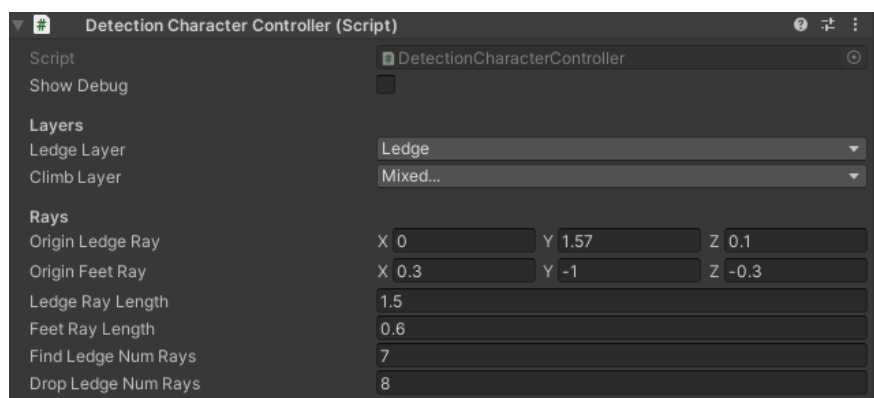


- **Movement Settings:** Movility settings like speeds and forces affecting the player
- **FeetIK:** Use IK on Feet
- **Environment Layer:** Layers the feet will try to be positioned
- **Height from Ground Raycast:** Raycast height to ground
- **Pelvis Offset:** Adds this value to the pelvis bone in case of needing to adjust to a model.
- **IK Speed:** Speed in which the limb will move to the IK Position.
- **Anim Variable Name:** Each Walk/Run animation has two curves created with those names to know when the foot in the animation is on the ground.

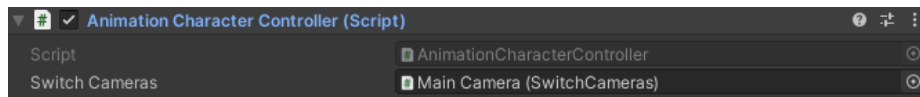
**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**



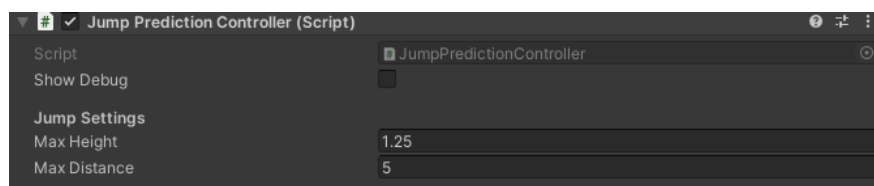
- You can Activate each Action the player can perform by checking/unchecking the action in the list



- **Layers:** Defined layers for the different collision test required for the Parkour System.
- **Rays:** Value settings of raycasts to detect different surfaces.

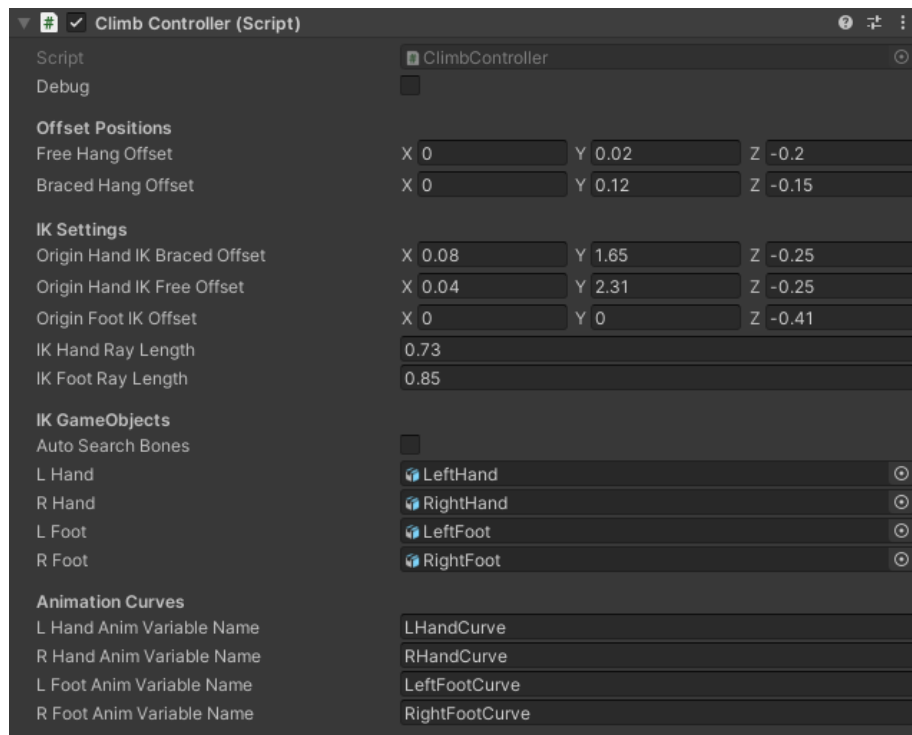


- **Switch Cameras:** Is used to blend between the FreeLook Camera and the SlideCam, that follows the player, when the player is sliding.



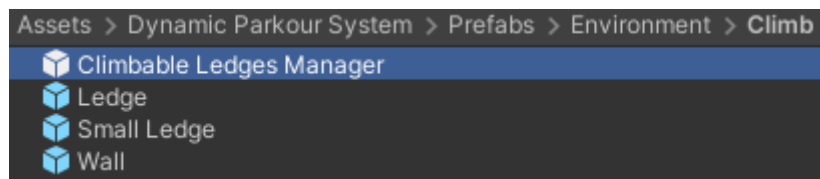
- **Max Height:** Max height the player can jump to a landing target
- **Max Distance:** Max distance the player can jump to a landing target.

**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**



- **Offset Position:** Offset positions to reposition the player while on Ledge.
- **IK Settings:** Raycast Settings to detect surfaces to place the hands and feet.
- **IK GameObjects:** GameObjects of the Player Skeleton Limbs, Auto Search will assign automatically the IK bones of the Humanoid Avatar.
- **Animation Curves:** Some animations have custom curve lines to know when an IK needs to be positioned

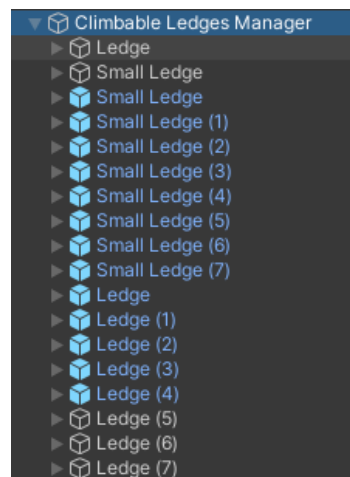
## 3.2 Setting Climbable Points Prefab (Ledges)



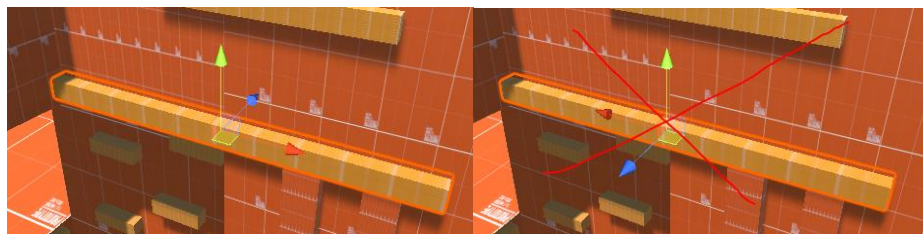
- 1) Drop a Climbable Ledges Manager in the Scene

**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**

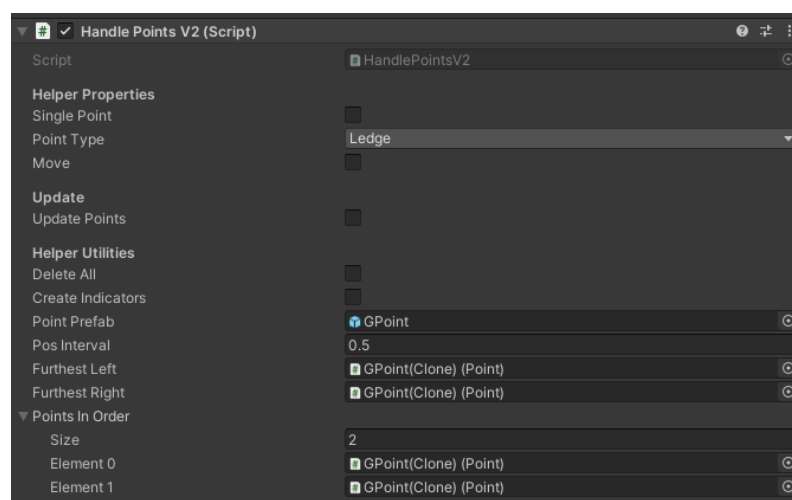
- 2) Add Small Ledges, and Ledges inside the Manager in the Scene



- 3) Make sure the direction of the Ledges is pointing towards the direction the player will grab.



- 4) If you need to Scale Up the Ledge horizontally, you'll need to Update the Climbable Points. Click to the Points child. And you will see the following Script in the Inspector

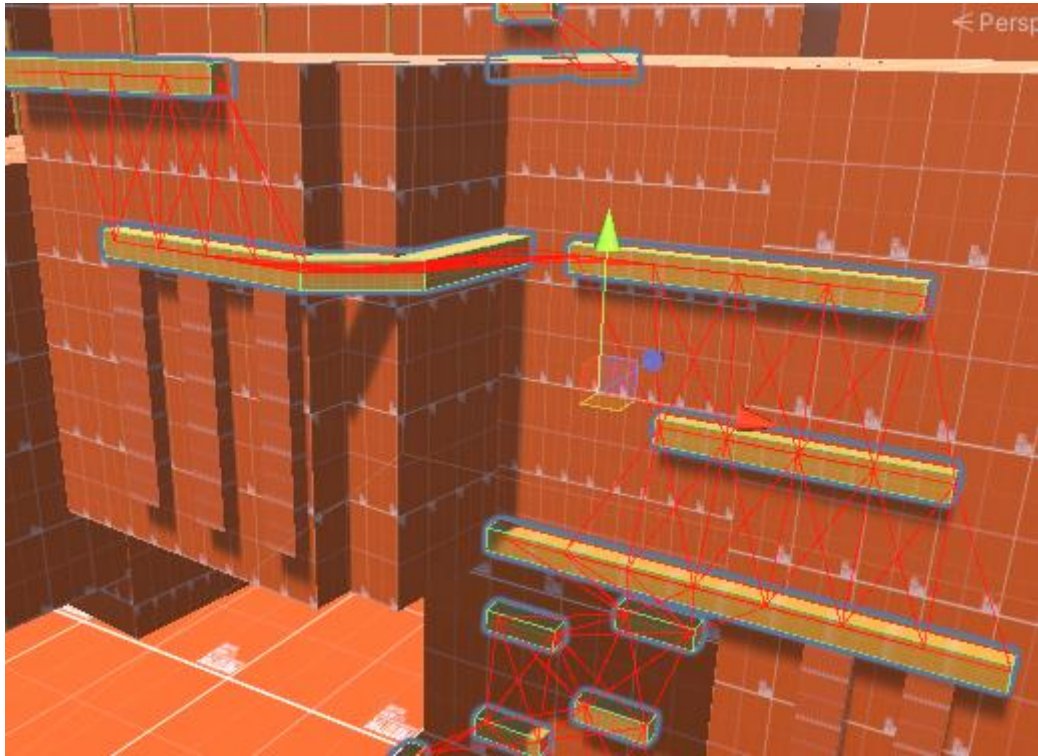


- 5) Click on Delete All -> Create Indicators -> Set Interval of Points -> Update Points. This will generate a new map of points on the Ledge.

**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**



- 6) Once configured the ledges just click on Climbable Ledges Manager -> Reset Connections -> Update Points -> Refresh on DrawLine Script. You'll see a map of points like this.



### 3.3 Setting Jump Landing Points (Pole)

Follow the instructions on how to create multiple points on the ledge in point 4 of the “**Setting Climbable Points (Ledges)**” section.

(These Landing points need to be attached to a GameObject with a Mesh)

### 3.4 Setting Jump Landing Points (Wall)

You can create custom jump landing Points by dragging the prefab “Jump Points”, this prefab contains 8 points in circular shape that you can reposition anywhere you need, to allow the player reach to that position.

## 4. Contact Me

If you need to contact me for support, you do it via email [knela96@gmail.com](mailto:knela96@gmail.com).

In case of having any bug via Issues Page of the Official Repository:

<https://github.com/knela96/Dynamic-Parkour-System>

Twitter: <https://twitter.com/knela96>

**\*This package needs the use of external packages like Cinemachine and the New Input System. (All FREE from Unity Asset Store)**