

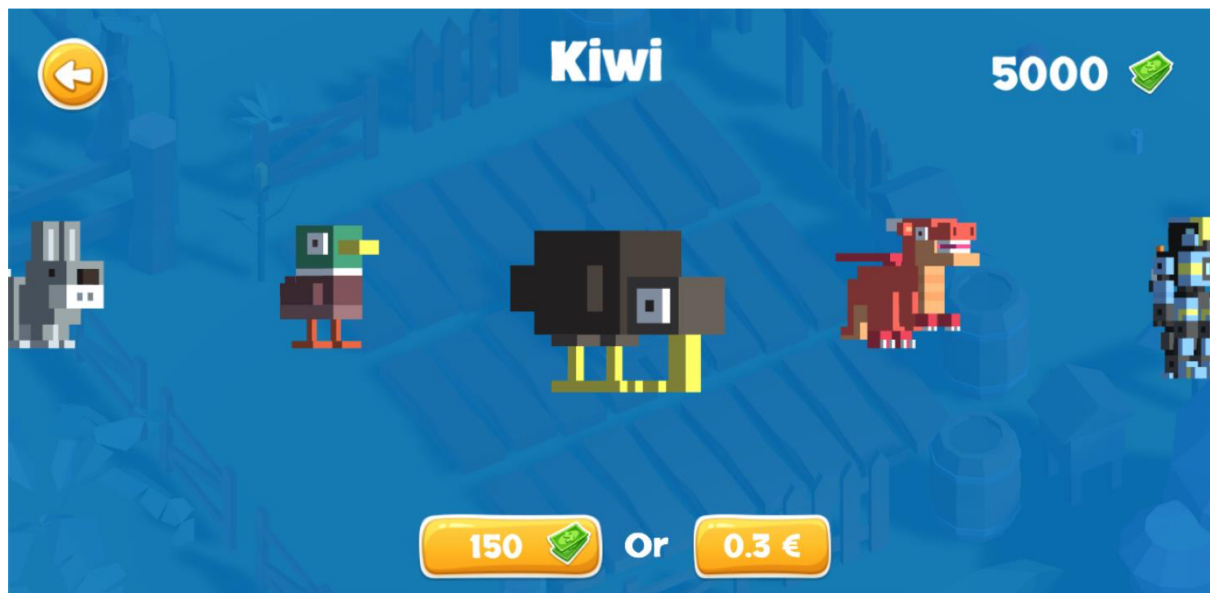
Character Selector Pro

Introduction

The Character Selector Plus package gives you an easy way to integrate a selection of a character.

The characters can be unlocked and purchased, it is a great way to retain players or earn some money (there are some sample scenes that show in detail the features)

The package also deals with saving purchases and manage coins.



Setup

It is recommended to watch the online tutorial:

<https://www.youtube.com/watch?v=Vej159o0T6k&feature=youtu.be>

- Drag the Prefab CharacterSelectorPlus/UI Camera and add it as a child to your Main Camera (3:20)
- Add the prefab CharacterSelectorPlus/CanvasCharacterSelector to your scene (4:13)
- From the CanvasCharacterSelector inspector, in the canvas component drag the previously added UI Camera into the Render Camera field (4:28)

- Create your prefab for the selector and add them to the Prefab list of CanvasCharacterSelector/Panel/CharacterSelectorController (it is recommended to duplicate an existing prefab, in the CharacterSelectorPlus/_Example/Prefab folder, and add the model as a child of the Cube GameObject within the prefab) (9:51)
- Add a button that opens the menu, using the ClickCanvas() function of the ManagerSelector script present on the CanvasCharacterSelector GameObject (4:46)
- Make sure an event system is present in the scene, otherwise add it:
Game Object -> UI -> Event System

Usage

For a usage live example, watch the video tutorial at [16:34](#).

The plugin is now configured and events can be monitored.

The plugin uses the Delegate pattern, you can then define observer methods by registering on the CharacterSelector class.

The following is the code of a possible script listening for the character selection and purchase character events.

```

1  using UnityEngine;
2
3  public class CharacterSelectorClient : MonoBehaviour {
4
5      private void OnEnable()
6      {
7          CharacterSelector.OnCharacterSelected += CharacterSelected;
8          CharacterSelector.OnPurchaseCharacter += PurchaseCharacter;
9      }
10
11     private void OnDisable()
12     {
13         CharacterSelector.OnCharacterSelected -= CharacterSelected;
14         CharacterSelector.OnPurchaseCharacter -= PurchaseCharacter;
15     }
16
17     private void CharacterSelected(string characterName)
18     {
19         foreach (Transform child in transform)
20         {
21             child.gameObject.SetActive(child.gameObject.name == characterName);
22         }
23     }
24
25     private void PurchaseCharacter(CharacterProperty characterProperty)
26     {
27         Debug.Log("You must pay " + characterProperty.name + ", for the character: " + characterProperty.nameObj);
28         GameObject.Find("CharacterSelectorController").GetComponent<CharacterSelector>().paymentConfirmed(characterProperty);
29     }
30 }
31

```

Key functions

Observe selection events:

Add in the OnEnable method of your script:

```
CharacterSelector.OnCharacterSelected += CharacterSelected;
```

And in the OnDisable method:

```
CharacterSelector.OnCharacterSelected -= CharacterSelected;
```

Where CharacterSelected is a function with the following signature:

```
private void CharacterSelected(string characterName)
{
}
```

Observe purchase events:

Add in the OnEnable method of your script:

```
CharacterSelector.OnPurchaseCharacter += PurchaseCharacter;
```

And in the OnDisable method:

```
CharacterSelector.OnPurchaseCharacter -= PurchaseCharacter;
```

Where PurchaseCharacter is a function with the following signature:

```
private void PurchaseCharacter(CharacterProperty characterProperty)
```

Get the number of coins:

```
PlayerPrefs.GetInt ("money", 0);
```

Set the number of coins:

```
PlayerPrefs.SetInt ("money", 500);
```

Unlock a character (Ex: after a payment is confirmed)

```
GameObject.Find("CharacterSelectorController")  
    .GetComponent<CharacterSelector>().paymentConfirmed(characterProperty);
```

Support

For every question and support request, do not hesitate to write to the following email
thundershotgames@gmail.com