

Considerations

First of all, thank you all for the opportunity. It was really fun to make this prototype and I really enjoyed the process. I must declare that I have never made a Top Down Unity game, my experience with the genre comes from years ago using Adobe Flash and AS3. The tools unity provide are much more efficient in making this type of game, I remember the pain it was to make various layers and sort all tiles every frame to see if the player had to be in front or in the back of each sprite. Such a pain! But unity makes this a lot more fun, so you can worry about real stuff that matters in the game.

Nevertheless I did use some help, as I never had used unity tile mapper before, so I did watch this tutorial a bit:

https://www.youtube.com/watch?v=b8YUfee_pzc

Sometime ago I purchased this asset:

<https://assetstore.unity.com/packages/2d/characters/customizable-pixel-art-character-kit-13984>

So the prototype was built around what I could gather from using this particular set of characters. Most of the time In the first couple of days was spent preparing and choosing the assets to be used in the game.

For the shop, I used some asset from <https://itch.io/>

I cant find the exact one I got (I had a bunch of then downloaded already).

As a final result I'm proud of what I could manage to build. If I had more time of course I would implement more pieces of clothes, gender (male female), dialogs and animations to the whole project.

What I'm not particularly proud of, is the way I loaded the resources, which is a bit slow. I was focusing on meeting all the requirements from the link you guys gave me. So performance was not on my priority list this time. I needed to make things work!

Please take in consideration that I am not a very good designer, so the way I constructed the shop is very simple and not as aesthetic pleasing as It could be. I did my best to build something cool with the assets I had.

Overview

Classes:

`class CameraController : Singleton<CameraController>`

Moves the camera to follow the player.

`class Constants : MonoBehaviour`

Just a bunch of strings, constants and helper functions. It all is related to asset naming and folder structure.

`class InventoryManager : Singleton<InventoryManager>`

It handles the inventory of the player, managing all owned items, equip and selling items.

`class NotificationCenter : MonoBehaviour`

This one is a script it got years ago from a unity wiki that does not exist anymore. It handles notification and observers to call functions without having the need to access a instance of the actual receiver object.

`class NPCOpenStore : MonoBehaviour`

Handles click on NPCS to open store, all parameters are configured on editor.

`class Player : MonoBehaviour`

Handles player movement and collision detecting.

class PlayerSkin : MonoBehaviour

Handles player equip/unequip. Also handles animations.

class ResourceManager : Singleton<ResourceManager>

Used to load sprites from resources folder.

class Singleton<T> : MonoBehaviour where T : MonoBehaviour

Base class to facilitate the process of making a Singleton

class Wallet : Singleton<Wallet>

Handles the player funds (add/remove)

UI

class UIPopupLayer : Singleton<UIPopupLayer>

Handles every popup in the game.

class UIPopup : MonoBehaviour

Base class to all popups, contains title, close button, confirm button and cancel button.

class UIHudFunds : MonoBehaviour

Just a displayer for the player funds.

class UIInventory : UIPopup

Popup that shows player inventory with tabs for every item slot.

class UIInventoryItem : MonoBehaviour

This is the actual item that is displayed on UIInventory

class UIMenu : UIPopup

Pressing ESC you see the menu to close the game or add funds if you need more.

class UIPopupAlert : UIPopup

Simple customizable alert that can be used all around. It is only used to notify the user has no funds left.

class UIPopupConfirmation : UIPopup

Popup displaying product and price to confirm purchase.

class UIPopupShop : UIPopup

This is the shop window to see all products the npc has to offer.

class UIProductCell : MonoBehaviour

The actual product that is displayed on UIPopupShop.

Tools:

class SliceRenamer : MonoBehaviour

class SliceRenamerEditor : Editor

Used to rename all slices within a spritesheet to have the same name followed by a index number. This is necessary to make all animations compatible between all spritesheets.