LSW Programming Interview

14.06.2021

—

Mohammed Alaghbari
Sana'a . Yemen

Overview

This document is explaining How the LSW programming test system works .

Goals

1. Player Movement in 2D Top-Down View world.
2. Player interaction With The World.
3. Make a Cloth Shop System with a buying/selling feature.
4. Displaying shop items icons & prices.
5. Talking to the shopkeeper.
6. Equipping The Bought outfits & Showing them on the character itself.

How Goals was tackled

I. Player Movement in 2D Top-Down View world.

Player Movement is Handled in Player Controller class that is taking input

And move the player's rigidbody by changing its velocity.

It also handles the Player Animations & Sounds.

II. Player interaction With The World.

Player interaction Handled in “IntraeactWithWorld “ class that checks if the player collider trigger any object that have inherit the “IInteractable” interface .

If it triggers an object it saves it in collider var to check if the User pressed E key to interact with the object & Display the Interact Text on Screen & Highlight the object .

if the User pressed E key it calls a function in the object to open the interaction UI , also it Disable the Player Movement by calling a function in the “Game Manager” class , it get the Interact Text off the screen & turn of the Highlight on the object .

III. Make a Cloth Shop System with a buying/selling feature , Displaying shop items icons & prices With Saving & loading the Data.

Make a Cloth Shop Handled in 5 classes “ShopSystem” namespace.

First the “ClothShopData” Serializable class that has the “ShopItem” array in it.

Second the “ShopItem Serializable class that provide 4 variables :

A.
```
public string itemName;
```
to set and get the item name .

B.
```
public bool hasBought;
```
to check if the item has already been bought or not .

C.
```
 public bool isEquiped;
```

to check if the item is equipped or not .

```
D.  public int cost;
```

To set and get the item cost.

```
E.  public enum partType {Top, Pants};
F.  public partType type;
```

To Select the type of the cloth item. (used in changing players outfit by checking the type and changing the SpriteResolver).

Third the “ClothShopUI” class ,its responsible for:

A. Getting All the required data for every item from the data class and Displaying it in the shop panel(item name, item price , item icon..) .
B. Displaying the items icons by matching the index between the sprites array and the ShopItemUI array.
C. Handles the Buying and selling methods by changing the item data and calculating if the players has enough money to buy, item cost in selling method divided by 2. (the two methods use int Attribute to check the item id).
D. Calling the Save function from the “ClothShop_SavingLoadingData” class.

E. Handling the equipping outfits method by calling a function in the “ChangeClothes” Class also change the isEquipped bool to true for the equipped item and false for the rest items from the same type.

Forth the “ShopItemUI” class that reference the text, icon ...etc for each item to use it in “ClothShopUI” and change them.

```
    public TMP_Text itemName; // to display item's Name

     public TMP_Text itemCost; // to display item's Cost

     public Image icon; // to display set item's icon

    public Button sellButton;

    public TMP_Text buyButtonText;
```

Fifth the “ClothShop_SavingLoadingData” class that is responsible of converting the items data to a json file and save it when the save function is called and loading the json file when the load function is called.

IV. Talking to the shopkeeper.

A simple dialogue system was made to tackle this feature and to talk any NPC in the game.

6 classes that were handling this feature.

1. “DialougeSystem” Serializable Class that provide :

```
public string name;
```

To set and get the name of the NPC.

```
    [TextArea(1, 10)]

    public string[] sentences;
```

To set and get the sentences of the conversation.

```
    public Question question;
```

To set and get the questions of the conversation if needed.

2. “Question” Serializable Class that handle the texts and choices

```
[TextArea(2, 5)]

public string text;

public Choice[] choices;
```

3. “Choice” Serializable struct that handle the choice text and the next dialogue for this choice .

```
[TextArea(2, 5) ]

public string choice;

public DialougeSystem dialouge;
```

4. “NPC” class that is inherit the “IInteractable” interface and call the TriggerDialgue Function from the “DialogueTrigger” class
5. “DialogueTrigger” class handles the dialogue for the NPC character

By setting the “DialogueSystem” Class variables. And send it the “DialogueManager” Class on its called.

6. “DialogueManager” Class responsible of starting the conversation and showing the sentences in order , displaying the words in typing effect (letter by letter showing) and Ending the conversation when the sentences completes or showing the questions if there was a questions .

V. Equipping The Bought outfits & Showing them on the character itself.

Handled by ”ChangeClothes” class that reference the sprite resolvers for the bodyparts and the sprite asset library

It has the changeOutfit function that checks for the body part and target category on the sprite library asset to change them by those values.

```
 public  void ChangOutFit (string bodyPart,string targetCategory ,
string label)

  {

    switch (bodyPart)

    {

        case "Top":

    bodyTargetResolver.SetCategoryAndLabel(targetCategory, label);

    GetComponent<Player_Controller>().referenceCatogrey =
targetCategory;

        break;

            case "Legs":

    legsTargetResolver.SetCategoryAndLabel(targetCategory, label);


        break;


        default:

        return;

    }
}
```

Game Code Architecture

I used the singleton pattern for this game Architect because it’s a small project and the singleton pattern will be compatible and enough for it.

Theres three singleton class :

GameManager : for enabling and disabling players movement.(should be managing other game properties if three was).

ChangeClothes class: to call the chang outfit function that change the category and sprite in players body parts.

UI Manager : to reference and manage all the UI in the game.

Save Load data Class : to save shop items data and load it .

Pre-written code .

Almost all of the code was written during the interview task, the only pre-written code was the saving and loading code.

All of the art was made during the task except the Characters was downloaded and the UI.

My own opinion on how well I think I did

Well my opinion that i took a lot of time in making this task .because two halves of the time was in searching for a free and suitable art but i couldn't find so i had to do some art and thats took alot of time and didnt make me do all of my ideas into this task . but i think that i've accomplished the main goals of this task and nothing more, of course i can make something more polished with more features and better game architecture if i got some ready art .

In the end I'm happy with what I did because I enjoyed every minute of this task.

Thanks for the chance & it will be a great pleasure for me if i joined LSW developers team and make everything that needs giving this great game finished and published as well as i'm excited to this also .

wish you the best of luck for this game and the team behind it no matter whats the outcome.