Salad Chef: Development documentation

The salad chef game comprises of four major modules namely gamemanager, customer spawn manager, customer and player.

GameManager :

The gamemanager controls the game flow of what will happen and when will happen. The functionalities that gamemanager control are

1.Aloting a seat to the customer - This functionality allocates a seat to the spawned customer. Once a customer object is spawned, one of the available positions from the list of transform positions **(List<Transform>)** near the customer table is given as the target position to the customer so that he navigates to that position.

2.Remove reserved seat - This functionality releases the seat from the customer when he leaves and adds them back to the list of free seats available for allocation. Once the customer object leaves the target position given to it is removed and added back to the available**(List<Transform>)** positions list, so that it can be allocated back.

3.Checking the delivered combo - This functionality checks, if the combo that is delivered to the customer is correct(is that he asked for). The chopped veg textmesh pro value is maintained with the player and a string var with the customer to save the requested salad combo. Once the player delivers the chopped vegetables, checking the combo method in the gamemanager is called to check the combo. The text value of the textmesh pro component of the player for chopped vegetables is checked with the requested salad combo saved in the salad combo var.

4.Reseting the combo of the player and the customer - This functionality resets the combo the player prepared and the customer requested, once the delivery is made correctly. Once the correct delivery is made the text value of the textmeshpro component for the chopped vegetables in the player and the salad combo variable in the customer is cleaned and set to string.empty respectively.

5.Check if the game is over - This functionality checks if the game is over. Created a flag for each player. Once the time runs out this flag automatically gets true stating that the current player's game has ended. Then a method in the gamemanager is called to check if other player has completed the game, if so game over UI is popped up showing the results.

6.RestartGame - Restart the game. UI button event system for triggering the event.Scenenmanager.loadscene to load the current scene.

7.Saving the high score - Once the game is over and the high score is saved in a class and later then serialized and parsed into a json format and write in a text file in to show the high scores later if needed.

8.Loading the high score - Once the game is loaded, the saved high score text file is read from its path and loaded into a class and sorted according to the top score order.

Customer Spawn Manager :

This manager is responsible for spawning the customer in regular intervals, and also allots the customer a seat.

1.Initiate The Customer Spawn - This method initiates the customer spawning functionality. A coroutine method responsible for customer spawning is called and the customer object is instantiated at a random interval of 1 to 5 seconds.After the instantiation is done the gamemanager is called for allocating a seat to the customer object(**transform position**) and the retrieved position is stored in the customer script target position for the object to navigate to.

2.Assign A salad Combo - This method is responsible for assigning a salad combination for the customer. The salad combination will be a maximum of three vegetables from A to F. First a random no is generated to set the salad combination length(**maybe 1 or 2 or 3**). Later for loop is used to for no of salad combination length to get the vegetables, for example if the length of the salad is 3, the for loop is iterated for 3 times and a random string from A to F is fetched and saved in the salad combination var in the customer script.

Customer Script :

This is the behaviour script of the customer.

1.Go To The Seat - This method navigates the customer to its allotted seat. Lean tween is used to move the customer object to the position. Vector3.lerp can also be used to move the player to the desired location, but lean tween gives callbacks once the current operation is completed. Once the player reached the seat location the leantween throws back a callback stating that the customer object has reached.

2.Changing The Current Seat To Occupied - This method changes the state of the current seat to occupied so that it cannot be assigned to another customer causing a bug in the game. A list of free seat positions and occupied positions seat positions are maintained. Once the customer script receives a seat position, that position is removed from the list of free positions and saved in the list of occupied seat positions. Then the customer spawning call is made to

spawn another customer. The customer wait time calculations are carried out and displayed in the UI above the customer. Leantween.value method is used to iterate through the wait time. Here normal update calculations can also be used instead of leantween, but i find it comfortable for me and it throws callbacks for every value update, so that the UI can be updated at regular intervals. Scroll rect UI is used for the display of the wait time bar.

3.Customer Leaves - When the time runs out the customer leaves. When the wait time becomes zero, the customer leave method is called. The salad combo is resetted and the occupied seat is released and added back to the list of available seats in the gamemanager. Again leantween is used to move the player out of the screen. Once the customer moves out of the screen the customer is destroyed.

Player Script :

This script is responsible for the behaviour of the player.

1.Move - Moving the player according to the input axis. The input axis in declared in the inspector so that different axis can be used for different players. The keycode for interaction with the vegetables, chop table and customer is made public so the players can have the flexibility to choose what key they want to interact with the objects.

2.Add Vegetables To Plate - When the player comes in contact with the vegetables plate and pressed the interaction button the vegetable is added to the player. Accomplished using the OnTriggerStay and when the interaction button is pressed.

3.Adding The Vegetables To The chop table - Used OnTriggerStay and when the interaction button is pressed the vegetable from the player is added to the chop table. The text value of the vegetable from the player is assigned to the text value of the chop table that has the textmesh pro component attached.

4.Adding the Chopped Vegetables Back To The Player - Once the vegetables that are added to the chop table from the player, the chop table takes some time to chop it and once done is added back to the chopped vegetables list of the player. Used Time.delta time for the countdown for chop time.

Functionalities Pending :

1.Speed Power up
2.Showing the high score - there are some issues with the saving and showing the high score list.

3.Adding the vegetables to the plate near by the chopping table.