Temasek Polytechnic

School of Informatics & IT

Diploma in Game Design and Development

{Buck$ Buddy} Technical Report

**Project Particulars**

| **Supervisor** | Dr Chek Yang Foo |
| --- | --- |
| **MP Project Title** | Buck$ Buddy For Special Needs |

**Project Team’s Particulars**

| **Admission Number** | **Student Name** |
| --- | --- |
| 2100441A | Ezry Irfan Bin Azni |
| 2100454D | Germaine Ong Si Jie |
| 2102565E | Domenic Ong Jun Yi |
| 2104454J | Padilla Carl Jameson Z. |
| 2100498A | Zidane Bin Zulkarnain |
|  |  |

**Contents**

[**Components Design**](#bookmark=id.1fob9te)

[**Components**](#bookmark=id.3znysh7)

[**Game Objects**](#bookmark=id.2et92p0)

[**Game Flow**](#bookmark=id.2s8eyo1)

[**Flow Chart Diagram**](#bookmark=id.17dp8vu)

[**Development information**](#bookmark=id.3rdcrjn)

[**Library**](#bookmark=id.26in1rg)

[**Version of software**](#bookmark=id.lnxbz9)

[**Hardware requirement**](#bookmark=id.35nkun2)

[**Setup of development platform**](#bookmark=id.1ksv4uv)

[**Detailed instruction on how to setup the development platform.**](#bookmark=id.44sinio)

***[Who has worked on every section of the MP final report must be clearly indicated. You must write Authored by: Xxx Yyy under the appropriate section headings.]***

# Components Design

## Components

***Description of all the components used in the game. Write all the component in the game and what is its purpose***

| ***Component Name*** |  |
| --- | --- |
| ***Description*** | *Give a basic description of the purpose of this component. What does it do, what can I modify from here.* |
| ***Variable*** | ***Only include variable that are important to the design, those variable that are use as a temporary variable need not be included e.g*** Int NumOfBullets - Use to keep track of the number of bullets currently in the game |
| ***Function*** | ***Only include important function, you may include common function like Awake, enable if there are important function being done inside***  *CreateBullet() - Help to create new instance of the bullet* |
| ***Game Object attached to*** | ***What is the game object it is attached to?*** |
| ***Author*** | ***Who created the script*** |

| *Component Name* | MapOpen Script |
| --- | --- |
| *Description* | This script opens the Map when it is clicked, and closes it when it is clicked again. |
| *Variable* | public GameObject panel – this GameObject is the Map, created as a Panel which displays the different aisles on it. |
| *Function* | *Void Start()- need to set the panel to be SetActive(false) at the beginning of the game.*  *OpenPanel()- used to check if the map (panel) exists or not, and if clicked, will set the bool to be true and the map to be active. If clicked again, the map will close.*    *ClosePanel()- used to close the Map if necessary* |
| *Game Object attached to* | *Map\_Btn* |
| *Author* | *Germaine Ong* |

| *Component Name* | AisleData (Removed) |
| --- | --- |
| *Description* | This component is to store the Data of what items are held in the Aisle. This acts as a datasheet. |
| *Variable* | The aisle data is created to track the food types within an aisle, hence the variables are Aisle Name and The food within the aisle |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  -GameManager |
| *Author* | *Who created the script*  -Ezry Irfan |

| *Component Name* | ItemData |
| --- | --- |
| *Description* | This component is to create Item data sets that can be applied to each food item for easier access. |
| *Variable* | The variables are:  Food ID (Int) - Establish Food ID  Food name (String) - Establish Food Name  Food Price (Float) - Establish Food Price  Food Icon (Sprite) - Establish Food Icon  Food Type (String) - Establish Food Type |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  Food Waypoints |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | LevelData |
| --- | --- |
| *Description* | This component is for the levelConfirm screen where the objective of the food, name, image and level number will be displayed. This is another DataSet |
| *Variable* | The variables are:  LevelNumber (Int) - Used to store Level Number  foodName (String) - Used to store the food name needed to be created  itemToMake (Sprite) - Image of the final item needed to be created  Ingredients (String[]) - The objective that the player needs to find |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  RandomEventHandler |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | NPCData |
| --- | --- |
| *Description* | This component is to create NPC Dataset for the random occurrence in the game. |
| *Variable* | The variables are:  Dialogue (string) - The dialogue that the NPC has  FirstImage (bool) - The first person talking image  SecondImage (bool) - The Second Person taking image  PlayerInteraction (bool) - Whether the player needs to interact with the NPC  FirstPersonName (string) - Name of the First Person  SecondPersonName (string) - Name of the Second Person  yesResponses (string) - The yes Responses  noResponse (string) - The no Responses |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  DialogueManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | SignboardData |
| --- | --- |
| *Description* | This component is to create Signboard Data so that the type of items in each isle can be displayed easily. |
| *Variable* | The variables are:  ItemName1 (String) - Item 1 in the aisle  ItemName2 (String) - Item 2 in the aisle  ItemName3 (String) - Item 3 in the aisle  Icon (Sprite) - The image of all the 3 items |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  Signboard |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | SignBoardHandler |
| --- | --- |
| *Description* | This component would display the signboard Data onto the actual signboard to modify all the sample text and default image with the respective needed signboard data. |
| *Variable* | This script relies on SignboardData for its variables. |
| *Function* | Update() – Update the placeholder signboard with the necessary new variables |
| *Game Object attached to* | *What is the game object it is attached to?*  Food Waypoints |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | TimeData |
| --- | --- |
| *Description* | This component is to create a dataset that would be storing the time taken by the player in each section of the game. |
| *Variable* | The variables are:  mainLevelTime (float) - Keeps track of the time in the main section  cashierTime (float) - Keeps track of the time in cashier section |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  GameManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | DialogueHandler (Removed) |
| --- | --- |
| *Description* | This component was to handle the original tutorial we planned to do however due to its limitation I have decided to rework it to allow it to have some functionality flexibility. This component would display the original dialogue text for the player to reach and understand the game’s mechanics. |
| *Variable* | The modifiable variables are:  -lines (String []) - Stores all the dialogue  -textSpeed (float) - Modifiable value of the text speed it displays |
| *Function* | Update – It would prompt the player to click once the sentences has completed. If the player clicked before the sentence completed, it would display the full sentence. Else, it would Display the next line once the player clicks and the text is indeed the full sentence displayed.  StartDialogue – It would begin the Coroutine  Typeline (Coroutine) – it would start a foreach loop for each of the characters in the sentence and slowly add it one by one into the text. This would give the spelling animation to make it more appealing.  NextLine – It would check if the list is empty to see the next sentence to be displayed, if it isn’t then it would display the next text in sequence. Else, it would disable the dialogue text bubble. |
| *Game Object attached to* | *What is the game object it is attached to?*  DialogueManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | GameOverScript |
| --- | --- |
| *Description* | This script would be handling the Game Over condition such as losing all your money… and also in charge of keeping track of how long the player takes in the level. |
| *Variable* | The variables are:  objectiveData (ObjectiveData) |
| *Function* | Update – Depending on the scene, it would start the timer and assign it to the correct dataset. It will also constantly be checking if the budget does reaches 0 which means that the player has failed prompting a Game Over screen.  Timer – It would take the current timer which is a float and convert it into minutes and seconds to simplify the float number. |
| *Game Object attached to* | *What is the game object it is attached to?*  GameManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | NPCRandomChance (Removed) |
| --- | --- |
| *Description* | It would control the spawning of the NPC and the movement relating to the NPC Interactions. |
| *Variable* | The variables are:  NPC (GameObject[]) - Holds prefabs of the different possible NPCs to spawn. |
| *Function* | NPCSpawner (Coroutine) – Depending on the scene it will spawn a certain NPC, at a random spawnpoint waypoint in the scene.  PlayerDetection – Check each object within the proximity collider on the NPC whether it is a player or not. If it is, return true.  GoToNextPoint – It checks whether the spawner is an odd number or even, and then it will make the NPC patrol between a set of point. If it detects a player, it will walk towards the player and prompt for a Y/N answer.  TestDialogue – This method will enable the dialogue box of the NPC, and will return the appropriate response. |
| *Game Object attached to* | *What is the game object it is attached to?*  RandomEventHandler |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | VisualizationScript (Removed) |
| --- | --- |
| *Description* | This was used to visualize the original collider for the NPC |
| *Variable* | N/A |
| *Function* | OnDrawGizmo() - It draws the collision hitbox |
| *Game Object attached to* | *What is the game object it is attached to?*  NPC(Removed) |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | Dialogue (Removed) |
| --- | --- |
| *Description* | This was the original dataset creator script for the creation of NPC dialogues |
| *Variable* | NPC (String) - Stores NPC’s name  NPCimage (Sprite) - Stores the NPC’s Image  Dialogues (string[]) - Stores the sentences the NPC will say  yesResponses (string[]) - the NPC response to the yes interaction  noResponses (string[]) - the NPC response to the no interaction |
| *Function* | N/A |
| *Game Object attached to* | *What is the game object it is attached to?*  N/A |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | DialogueManager |
| --- | --- |
| *Description* | This script handles the entirety of the NPC and Interaction with the player and what outcomes it will put out. |
| *Variable* | Queue<string> dialogues, dialogueNames,yesResponses,noResponses  Queue<bool> leftSideTalking,StartingOptions  Queue<bool> rightSideTalking  TextMeshProUGUI nameText, dialogueText  Image NPCImage1,NPCImage2  Animator animator  NPCData[] NPCDataList  NPCData npcData  GameObject yesButton, noButton, ContinueButton, LeftArrow, RightArrow, NPCDialogueBox  Button MapBtn, ShopBtn, arrowTest  NPCSpawning npcSpawning  bool talkingToNpc, NPCInteracted,Scammed - Check whether each of these player interaction has been triggered  int index; - Keeps track on how many player interaction there has been  GameOverScript gameOverScript |
| *Function* | NPCRandomChance() - This is to randomize a chance to trigger the NPC whenever you fast travel using the map.  StartDialogue() - It will activate the necessary UI elements and obtain the appropriate NPC dataset responses, actions etc. and put them in a queue.  DisplayNextSentence() - It would dequeue the responses everytime it displayed a new sentence. Or end the conversation if the responses count = 0.  TypeSentence() - Will ‘animate’ the text to make it look like it is typing by adding character by character.  StarterPlayerInteraction() - Display yes and no option.  LeftPersonTalking() - if the left person is talking then light up the image.  RightPersonTalking() - if the right person is talking then light up the image.  YesChoice() - Carry out the yes actions.  NoChoice() - Carry out the no actions.  CloseTimer() - Disable all the NPC related canvas.. |
| *Game Object attached to* | *What is the game object it is attached to?*  Dialogue Manager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | DialogueTrigger |
| --- | --- |
| *Description* | Button input for a random chance of triggering the NPC |
| *Variable* | N/A |
| *Function* | TriggerDialogue - Calls the random NPC chance method |
| *Game Object attached to* | *What is the game object it is attached to?*  BakeryAisle\_Btn  DrinksAisle\_Btn  SnacksAisle\_Btn  FruitsVegetableAisle\_Btn  FrozenFoodAisle\_Btn  DairyAisle\_Btn  CannedFoodAisle\_Btn  MeatSeafoodAisle\_Btn  RiceAisle\_Btn |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | NPCSpawning |
| --- | --- |
| *Description* | This handles the NPC Spawning and decides whether the player interacted. |
| *Variable* | TypesOfNpc (GameObject[]) - Stores the potential NPCs that can be spawned |
| *Function* | Update() - Checks if player has been interacted and whether it is vicinity before starting the dialogue.  FindClosestWaypoint() - Finds the closest waypoint to the player and removes it from the available spawn locations list.  SpawnNPC() - Checks what type of NPC is it from the dataset and spawns it at one of the available spawn points.  PlayerInVicinity() - Check if the player is in the vicinity of the NPC. |
| *Game Object attached to* | *What is the game object it is attached to?*  GameManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | TutorialScript |
| --- | --- |
| *Description* | This script handles the entire tutorial sequence. |
| *Variable* | DialogueBoxAssets (Sprite[]) - Stores the different dialogue box assets that can be used  CoinAssets (Sprite[]) - Stores the different coinAssets that can be used  CoinMascot (Image) - stores Coin Mascot Image  DialogueBox (Image) - Stores the dialogue Box Image |
| *Function* | Start() - Check if its level 1 and play the tutorial.  MapTutorialStart() - Check if the player has opened the map before and if he hasn't it will trigger the map tutorial sequence.  StartDialogue() - Adds each dialogue from the tutorial dataset into the Queue variable  DisplayNextSentence() - Will ensure that the player doesn’t skip through the tutorial without the message being fully written. If the player clicks mid-instruction it will display the full finished sentence. It would then dequeue the sentence to display the next one. It will also disable the tutorial once all sentences have been displayed.  CloseTimer() - It will disable all the tutorial UI Element after a few seconds.  TypeSentence() - ‘Animate’ the sentence by adding character by character  DialogueStarter() - To handle the dialogue positions and coin mascot position on the tutorial screen. This will change the transform position of the coin and dialogue to look ‘animated’. |
| *Game Object attached to* | *What is the game object it is attached to?*  GameManager |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | ButtonClicked |
| --- | --- |
| *Description* | This script is to check which level is selected |
| *Variable* | N/A |
| *Function* | WhichBtnClicked– Obtains the button that is clicked and input the level number into another script |
| *Game Object attached to* | *What is the game object it is attached to?*  Level Buttons |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | ButtonDataHolder |
| --- | --- |
| *Description* | Handles the level confirmation screen to modify the different UI elements as well as the transition between level selection and main game |
| *Variable* | The variables are:  starImages (Sprite[]) - Used to store the different star images |
| *Function* | Update – Handles the UI element to change to its respective level description. Handles the high score of the player as well. |
| *Game Object attached to* | *What is the game object it is attached to?*  RandomEventHandler |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | ButtonPressFunctions |
| --- | --- |
| *Description* | This script handles most of the scenemanagement related button outcomes. |
| *Variable* | N/A |
| *Function* | ConfirmSelection()– Obtains the level number from a different script and loads the respective level.  StartGame()– Loads the character selection screen.  StartLvl()– Loads the level selection screen.  Back()– Returns back to the level selection screen.  NextLevel()– Obtains the current level the player is on and adds 1 to move to the next level.  ReturnToMainMenu()– Loads the Main Menu screen.  ReturnToGame()– Returns back to the current level from the cashier scene. |
| *Game Object attached to* | *What is the game object it is attached to?*  Canvas |
| *Author* | *Who created the script*  Ezry Irfan |

| *Component Name* | SettingMenuHandler |
| --- | --- |
| *Description* | Handles all the setting menu button outputs |
| *Variable* | N/A |
| *Function* | MenuButtonClicked()– Sets the menu button active/inactive.  ConfirmationPage()– Activates the confirmation page to allow Smart Shopper to be more error-proofing with extra confirmation.  Decline()– If the player declines in the confirmation page it will return back to the pause menu. |
| *Game Object attached to* | *What is the game object it is attached to?*  GameManager |
| *Author* | *Who created the script*  Ezry Irfan |

## 

| *Component Name* | MapDataList |
| --- | --- |
| *Description* | The script stores the different food types in each aisle. It has 2 functions, to return a string (aisle location, bakery etc) if needed to find where the food is located. Returns a string array filled with all the different food types in the specified aisle (if searching for dairy aisle, returns cheese, eggs, milk etc) |
| *Variable* | Arrays used to store what food types are in each aisle   * String[] bread * String[] drinks * String[] snacks * String[] fruits * String[] frozen * String[] dairy * String[] meat * String[] rice * String[] canned |
| *Function* | GetAisleTypes(string) - *Returns an array with all the different food types in the specified aisle*  GetAisle(string) - *returns a string to show what aisle the food type is in* |
| *Game Object attached to* | GameManager |
| *Author* | Zidane |

| *Component Name* | Objective |
| --- | --- |
| *Description* | The script used to display the objective on the 2 different shopping list. It is also used to see if the player has enough items to fulfil the level’s objective. |
| *Variable* | Public ObjectiveData objectiveData - used to get the data of the level from assets (drag and drop).  Public GameObject textPrefab - to get the text prefab to instantiate listed items in the main shopping list  Public Gameobject dynamicTextPrefab - to get a text prefab to instantiate list items in the dynamic shopping list  Public Gameobject shoppingListContent - where the textPrefabs would instantiate into.  Public Gameobject dynamicShoppingListContent - where the dynamicTextPrefab would instantiate into.  Public GameObject dynamicShoppingList - to enable or disable the dynamic shopping list if a menu opens.  Public List<KeyValuePair<string, int>> objList - a list that stores the number of food types needed for the level.  Public List<string> requiredAisleItems - stores what food type is required in the aisle  Public TexteshProUGUI levelBudget\_Txt - to display the budget for that level.  Public MapLocation mapLocationScript - used to hold the script that stores what item is in each aisle. |
| *Function* | Start() - Would instantiate text prefabs into the shopping list to show the objective of that level. Another function is to display the budget of the level.  Update() - Would sort the dynamic shopping list if the player changed into a different aisle.  Public List<string> GetCurrentAisleItems() - returns a list of strings, where the strings are what food type is in that specific aisle.  Public void SortShoppingList() - sorts the dynamic shopping list. Displays what food type is in that aisle while hiding the others.  Public void ShoppingListDisplay() - would instantiate all the text prefabs to both the main shopping list and the dynamic shopping list. Also adds the required amount of food types into the objList variable.  Public void ShowBudget() - shows the budget of the level in the shopping cart.  Public float GetBudget() - would return the budget of the level.  Public void ObjCheckOff() - would check off items in the dynamic shopping cart if the players have the required amount.  Public void ChangeItemDataBoolValue(string) - would get the item data according to the string parenthesis and switch the “sufficientAmount” bool to false if the quantity of that item in the shopping cart goes from 1 to 0 |
| *Game Object attached to* | GameManager |
| *Author* | Zidane |

| *Component Name* | ObjectiveData |
| --- | --- |
| *Description* | To create a data sheet for each level’s objective |
| *Variable* | Public class ItemObjListThing (has System.Serialiable)   * Public string ItemType - to save the type of food required * Public int amount - to save the amount required for the type * Public Sprite iconSprite - to save the sprite of the type of food   Public ItemObjListThing[] dynamicObjList - contains an array of all the objective requirements for that level  Public float budget - to save the budget for that level |
| *Game Object attached to* | Right click in project folder > Create > ObjectiveData |
| *Author* | Zidane |

| *Component Name* | ObjectiveDataHolder |
| --- | --- |
| *Description* | Used to store relevant data for the listed item in the shopping list, such as what is the item type of the listed item and how much is required. |
| *Variable* | Public string typeOfItem - stores what item type it is (Bread/Ham)  Public int quantity - stores how much of the item type is required for the level.  Public bool isItemHere - bool to see if the item is in the aisle the player is in.  Public int orderNumber - used for sorting the item in the dynamic shopping list.  Public GameObject line - used to cross out items in the shopping list is the player has enough items. |
| *Function* | *NA* |
| *Game Object attached to* | Prefabs   * ShoppingCartItems * DynamicShoppingCartItems |
| *Author* | Zidane |

| *Component Name* | InventoryItemController |
| --- | --- |
| *Description* | Attached to cart displayed item so they hold the correct ItemData. |
| *Variable* | Public ItemData item - used to hold the ItemData for displayed item.  Public Button removeButton - used to hide or show the button being displayed  Public TextMeshProUGUI quantityTxt - used to show the correct number of items in the cart. |
| *Function* | Public void RemoveOne() - used to remove the selected item in the shopping cart.  Public void AddItem(ItemData) - get the correct ItemData from another script |
| *Game Object attached to* | Prefab   * ItemListed |
| *Author* | Zidane |

| *Component Name* | InventoryManager |
| --- | --- |
| *Description* | Main script for that hold most functions for all the shopping list items |
| *Variable* | Public List<ItemData> itemList - list that holds all items added to cart  Public List <ItemData> itemSpawnedList - list  Public Dictionary<string, int> duplicateCounts - List that holds the quantity of each item (3 branded bread, 2 white eggs)  List<string> idList - used to find duplications in the list  //List<string> spawnedCartItems - list that holds which item has already been displayed in the shopping cart. Needed to it doesnt show duplicates.  Public InventoryItemController[] InvetoryItem - used to attach the correct item data to the item spawnedLIst.  Dictionary<string, int> cartObjDiction - list that hold the different quantity of food types (3 bread, 2 eggs)  Dictionary<string, bool> checkObj - used to check if the player has enough items for the level obj.  Public Gameobject textPrefabs - used to be instantiated into the shopping cart list  Public budgetReminder - used to hide or show the budget reminder if the players is close to using up all the budget.  Public transform ItemContent - where the shopping cart is, used as a parent for all the instantiated textPrefabs.  String missingItems - a string to return what the player is missing before continuing to the cash register.  Public float totalPrice - used to store the total cost for the entire shopping cart.  Public float budget - used to store the budget for that level.  Public sotreCartData cartData - script that would hold the cartListData and itemList. Used when going over to the cash register scene. |
| *Function* | Public void Add(itemData) - used to add the item to the list and changes values for budget related floats accordingly.  Public void Remove(itemData) - used to remove an item from the list and changes values for budget related floats accordingly  Public void ShowItem() - function that will display the correct information while opening the shopping cart.  Public Dictionary<string, int> CountDuplicates(List<string>) - used to count the total amount of individual items (3 branded bread, 2 fresh white eggs).  Public void CleanList() - clean up all list to help display the correct information to the shopping cart.  Void FindIds() - used to add the item names to the idList.  Public void SetInventoryItems() - used to set the correct itemdata to the correct shopping cart listed items.  Public bool CheckForObj() - returns a bool after it checks if the player has enough items to continue to the cash register and if the player has not went over the budget.  Public string FindMIssingItems() - returns a string that contains all the missing items the player needs for the shopping list.  Public string NPCFindMissingItems() - returns a string that contains one missing item  Public Dictionary<string,int> GetCartTypes() - returns a dictionary that shows the quantity of item types in the shopping cart. |
| *Game Object attached to* | InventoryManager |
| *Author* | Zidane |

| *Component Name* | ItemAddToCart |
| --- | --- |
| *Description* | The pop up food bubble would use this script to add items to cart |
| *Variable* | ItemController itemController - get the item data from the food waypoint.  ItemData itemData - would store the itemdata from the itemController.  ItemScript itemScript - stores the script from the food waypoint  TextMeshProUGUI tmp - edits the text on the food bubble pop up  GameObject dataToTake - stores food waypoint gameobject |
| *Function* | Void GetFoodItemData() - to get the food waypoint gameobject  Void DisplayFoodData() - after getting the food waypoint gameobject, get the components in the gameobject to display it on the text mesh pro.  Public void AddToCart() - button function that adds item to cart. |
| *Game Object attached to* | Prefab   * FoodBubbleCanvas |
| *Author* | Zidane |

| *Component Name* | ItemController |
| --- | --- |
| *Description* | Holds the itemData for the displayed items in the shopping list. |
| *Variable* | Public ItemData item - stores the ItemData for other scripts to use |
| *Game Object attached to* | Food waypoints in the game (SoyaMilkWP, FreshMilkWP etc.) |
| *Author* | Zidane |

| *Component Name* | storeCartData |
| --- | --- |
| *Description* | Used to store the relevant cart data to display it in the cash register scene |
| *Variable* | Public Dictionary<string, int> cartData - stores cartData from the inventory manager.  Public List<ItemData> data - stores the list of itemData from the inventory manager. |
| *Function* | Public void CleanList() - clears both variables |
| *Game Object attached to* | RandomEventHandler (don't destroy object in preload scene) |
| *Author* | Zidane |

| *Component Name* | CharacterScript |
| --- | --- |
| *Description* | Handles the character's movement and changes between the male or female model. Also handles the animation of the character |
| *Variable* | Float horizontal - gets a float of -1 or 1 depending if the player is moving left or right on the keyboard  Public float speed - used to change the speed of the player  Float timer - used in timers  Public bool isFacingRight - use to flip the player left or right  Bool isAFK - used to start the return to idle animation timer  Public bool movingLeft, movingRight - use to move the character left or right  Rigidbody rb - to add velocity  Public Animator anim - to change the animation of the player  Public Gameobject maleModel, femaleModel - hides the other model depending on the players preferred model.  Public Animator maleAnim, femaleAnim - gets both of the animator for each model. Would then be saved to the variable anim to change the animations of the character.  Public PlayerSelectOption selectedCharScript - to get which model the player chooses. |
| *Function* | Start() - hides either male or female player model depending on player choice.  Update() - plays the correct animation if the player is moving. Would run the timer function if the player is not moving  FixedUpdate() - where the movement gets updated  Void StartTimer() - if the player hasn't moved for longer than 2 seconds, play the correct animation  Void Flip() - flips the player to face left or right  Public void MoveLeftDown() & MoveRightDown() - both functions is to move the character by the on-screen buttons.  Public void MoveLeftUp() & MoveRightUp() - both functions are used to make the player stop moving if the on-screen buttons are released. |
| *Game Object attached to* | * Character |
| *Author* | Zidane |

| *Component Name* | ItemScript |
| --- | --- |
| *Description* | This script handles the spawning and displaying information on the food bubble pop-up |
| *Variable* | Public GameObject infoBubble - the get the prefab of the food bubble pop up  GameObject foodBubbleClone - the instantiated infoBubble prefab  Vector3 foodBubblePos - to set the location of the foodBubbleClone  Public bool didTextSpawn - check if the prefab got instantiated |
| *Function* | Private void OnTriggerEnter(Collider) - would run the spawn food bubble if the player is the one colliding.  Private void onTriggerExit(Collider) - would run the code to delete the food bubble if the player left the collider box.  Public void ShowBubble() - spawns the foodbubble  Public void DeleteBubble() - deletes the foodBubble |
| *Game Object attached to* | All the food waypoints in the levels (FullCreamMilkWP, OrganicEggWP etc) |
| *Author* | Zidane |

| *Component Name* | MapLoopTeleport |
| --- | --- |
| *Description* | Script to loop the level if the player goes past the last aisle, it teleports them to the start of the level. |
| *Variable* | GameObject leftBarrier, rightBarrier - the collider. If the player touches this it would teleport them.  GameObject leftTpPoint, rightTpPoint - a game object where the player will be teleported to. |
| *Function* | OnTriggerEnter(Collider) - if the player collides with one of the barriers, teleport them to the other side of the map. |
| *Game Object attached to* | Character |
| *Author* | Zidane |

| *Component Name* | AdjecentAisleScript |
| --- | --- |
| *Description* | Script to attached to the AdjecentAisles gameobject to display where the player currently is and what is the aisle left and right of the player. |
| *Variable* | Public Gameobject currentObject, leftObject, rightObject - to get gameobject of the childs in the adjacent aisle script to easily get the text and image.  Public gameobject go is the entire parent (the gameobject) to easily get the child that shows the current, left and right aisle.  Public TextMeshProUGUI currentAisle\_txt, leftAisle\_txt, rightAIsle\_txt - to display the name of the aisles.  Public image currentAisle\_img, leftAisle\_img, rightAisle\_img - to change the sprite of the current, left and right aisle.  Public GameObject[] aisles - to get the different aisle gameobject to find where the player is  [System.Serialiable] Public class MatchAisleIcons   * Public sprite icon * Public string aisle   -To save the image icon of the aisle. |
| *Function* | Void DisplayCurrentAisle() - sets the current aisle display to show the text and sprite of the aisle  Void DisplayRightAisle() - sets the right aisle display to show the text and sprite of the right aisle |
| *Game Object attached to* | AdjecentAisle gameobject |
| *Author* | Zidane |

| *Component Name* | MapLocation |
| --- | --- |
| *Description* | Used to find which aisle the player is in. |
| *Variable* | Public GameObject[] borders - gets the start and end point of the aisle.  Public List<GameObject> sectionsClone - used to clone the borders array to be used to find adjacent aisles.  Public GameObject Player - to get player location  Public float closestDistance - used to find which aisle is closest to player.  Public Gameobject closestPoint - used to get the gameobject the player is closes to.  Public Gameobject currentAisleSection - used to compare to closestPoint gameobject to check if the player is still in the same aisle.  Public string currentAisle\_string - holds the current aisle the player is in. |
| *Function* | Public void RestCloneList() - used to reset all values of variables and clone list.  Public string FindPlayer() - returns a string that finds out which aisle the player is in  Public string FindLeftAdjacentAisle() - returns the string that is the aisle left of the player. If there is nothing to the left, return the furthest aisle.  Public string FindRightAdjacentAisle() - returns the string that is the aisle right of the player. If there is nothing there, return the furthest aisle  Public string FindFurtestAisle() - returns a string that contains the aisle that is the furthest from the player.  Public bool DidPlayerChangeAisle() - used to check if the player changed aisle. |
| *Game Object attached to* | GameManager |
| *Author* | Zidane |

| *Component Name* | MapOpen |
| --- | --- |
| *Description* | Script would handle most UI elements. |
| *Variable* | Public GameObject player - stores the player Gameobject.  Public Gameobject shoppingCartPanel - Hold the gameobject to open and close the shopping cart.  Public Gameobject shoppingList, dynamicShoppingList - both gameobjects would display the required item for that level.  Public Gameobject objPanel - pop up menu that shows what the players are missing before going to the cash register.  Public GameObject helpPanelCtnBtn - a button in the objPanel. Used to hide or show the button.  Public GameObject budgetRemainderPanel - to hide or show the panel.  Public GameObject[] waypoints - an array that stores the waypoint of each aisle around the level.  Public TextMeshProUGUI helpPanelBody\_Txt - to edit the body text in the budget objPanel  Public TexdtMeshProUGUI budgetReminder\_Txt - to edit the text in the budget remainder.  Public MapLocation mapLocationScript - to access the functions in the script to find where the player is.  Public sprite maleHead, femaleHead - used to switch the sprites on the map depending on what model the player chose.  Public GameObject selectedHead - the gameobject to move around the map to show where the player is  Public PlayerSelectOption selectedModelScript - gets the script to get which character model the player chose.  Public waypointString - to get a string from the waypoints array.  String[] aisles - stores strings for the different teleport buttons.  GameObject[] aislePoint - stores array filled with buttons on the map. Used to put the head on the correct aisle where the player is. |
| *Function* | Start() - has the function to change the image of the character's head on the map  Public void ShowMainShoppingList() - function would display the main shopping list only when the shopping CART or map is opened.  Public void CheckOut() - attached to the go to payment button in the map. Would check if the player is not over budget and if they have enough items for the objective.  Public void SaveCartItems() - would save the cart data from the inventory manager to the script attached to the dont destory on load gameobject  Public void GetHelpForMissingCart() - gets the required item the player needs for the objective and stores it in a string.  Public void TeleportToAisleDynamic() - function to teleport to the correct aisle after the player presses a button on the map.  Public void BudgetReminder() - would show or hide the budget reminder panel and show the player how close or over budget they are. |
| *Game Object attached to* | Map\_Btn |
| *Author* | Zidane |

| *Component Name* | PlayerSelectOption |
| --- | --- |
| *Description* | Script would store which character the player chose. |
| *Variable* | Public bool isMale - stores the bool to change the player model to what the player wants. |
| *Game Object attached to* | RandomEventHandler |
| *Author* | Zidane |

| *Component Name* | SelectCharacterScript |
| --- | --- |
| *Description* | Script is used in the PlayerSelect scene to pick between the male and female character |
| *Variable* | Public bool isMale - to save it to the dont destroy on load game  Color dim, bright - to change the player model to dim if the other model is being selected.  Public Button maleButton, femaleButton - buttons for players to select what gender they would like.  Public Button ctnBtn - button to continue to the next scene.  Public PlayerSelectOption playerSelectScript - the script where it would store the isMale bool to be used in later levels. |
| *Function* | Public void SelectMale() - would dim out the female character and brighten the male character  Public void SelectFemale() - would dim out the male character and brighten the female character  Public void continue - would continue to the next scene only if the player selects a model |
| *Game Object attached to* | Canvas (in playerSelect scene) |
| *Author* | Zidane |

| *Component Name* | ShowCartTotalAmount |
| --- | --- |
| *Description* | Script to show the total cost of items in the cart |
| *Variable* | Int State - used to switch between on/off eye  Image image - the image that would have sprites be switched  TextMeshProUGUI totalAmtTMP - text to display the total amountl.  Private Spritep[] switchSprites - hold the two sprites the script would switch between. |
| *Function* | Public void ShowOrHideTotal() - a function for a button to transition between showing or hiding the total amount in the cart.  Void Update() - gets the total price of the shopping cart and would show the total amount depending on the State int. |
| *Game Object attached to* | TotalAmountHelper (in canvas for levels) |
| *Author* | Zidane |

| *Component Name* | DisplayFoodItems |
| --- | --- |
| *Description* | Script used to display a list of items in the cash register scene. |
| *Variable* | public GameObject textDisplay - a prefab to show the items  Public GameObject smallerTextDisplayer - prefab but for the smaller screen  Public Gameobject listContent - the location where the textDisplay prefab will instantiate into.  Public GameObject listSmallerContent - the location where the smallerTextDsiplayer prefab will instantiate.  Public List<string> spawnedItemBubble - to keep track of which item already has been displayed in the listContent  Sting itemTxt - used to hold the name of the item. |
| *Function* | Void ShowItems() - to display the items on the cash register screen |
| *Game Object attached to* | Prefab   * ItemListed |
| *Author* | Zidane |

| *Component Name* | Payment Drop |
| --- | --- |
| *Description* | This is the Payment Drop Script; the purpose of this component is to take in notes and coins and add in the assigned values to it. It also updates the progress bar that is a visual representation of the remaining cash needed to drag in for the player. |
| *Variable* | * Public GameObject[] placement – It is used for the placements of each note and coin. A invisible gameobject with the Tag “Waypoint” is used in order for the variable to find these gameobjects. * Public List<GameObject> activeMoney – it is a list that keeps track of the money that is dragged into the payment gameobject. * Public List<GameObject> twoDollar   Public List<GameObject> fiveDollar  Public List<GameObject> tenDollar  Public List<GameObject> fiftyDollar  Public List<GameObject> hundredDollar  Public List<GameObject> oneDollar  Public List<GameObject> fiveCent  Public List<GameObject> tenCent  Public List<GameObject> twentyCent  Public List<GameObject> fiftyCent – The purpose of these Variables is that Each of them contains the specific notes and coins of the GameObjects. It is used to keep each note and coins sorted into a list in order to check that they have been dragged in.   * Public Text moneyGenerator & public float randomNumber – The purpose of this text is to display the total money that the player needs to pay. * Public Text progressBarText – The purpose of this text is to display the total amount of money the player has dragged into the payment system in order to keep track of how much left is needed to pay. * Public Button confirm – This is for the confirm payment button which the player clicks if he/she thinks the amount to be paid Is sufficient * Public Image mask – This is for to use for the progress bar and is needed to fill up the progress bar according to whatever the player adds to the board and is a visual representation to show the player how much he/she has filled up. * Public float fillAmount – Amount was needed to keep track of the bar and set a total amount for it to fill up accordingly. |
| *Function* | Start() – Function needed to set and call different components to the script for later use. Sets some parts of the canvases that do not need to appear first to false.  OnDrop(PointerEventData event Data) – Function used for the drag and drop system.  Checker() – Purpose of this is the scoring system of the game, the scoring system is based on how off the player is to the exact amount. Eg. If player is off by 3 dollars, he/she gets a 2 star on that level.  GetCurrentFill() – Purpose of this method is for the progress bar to indicate how much the player has paid.  ResetCash() – It is a method created for when the player presses the reset money button and it puts all the cash back into its original place  ClearList() – Clears the list I the storedData function, its initially for the reset button. |
| *Game Object attached to* | *What is the game object it is attached to?*  The Gameobject it is attached to is the Payment GameObject. It is an invisible 2D box around the NTUC Cashier and it makes it seem that you need to give money to the NTUC Cashier in order to pay for the food. |
| *Author* | *Padilla Carl Jameson Z.* |

| *Component Name* | Cash Spawner, Cash Spawner Five, Cash Spawner Ten, Cash Spawner One, Cash Spawner FiftyC, Cash Spawner TwentyC, Cash Spawner TenC, Cash Spawner FiveC |
| --- | --- |
| *Description* | This is the Cash Spawner Scripts, I put them altogether because they all have the same exact code in each gameobject, the purpose of these components is to spawn a random amount of notes and coins for the player to pay to the counter. These are randomized based on the budget. |
| *Variable* | * public GameObject cash – Needed to call cash gameobject component to * public Vector3 cashPos – To determine the cash position (where the cash will be placed when it spawns) * public float offset – the offset needed to move the cash to show that it is a stacked amount. (For Aesthetics) * public float offsetCount – needed to increment the amount and move the cash accordingly when dragged in (For Aesthetics) |
| *Function* | FiveDollarSpawner() (All the scripts have different names on the method but all have mostly the same function) – Used for creating a new instance of the cash and coins. |
| *Game Object attached to* | * 2DollarSpawner * 5DollarSpawner * 10DollarSpawner * 1DollarSpawner * 50cSpawner * 20cSpawner * 10cSpawner * 5cSpawner |
| *Author* | *Padilla Carl Jameson Z.* |

| *Component Name* | DragDrop |
| --- | --- |
| *Description* | This is the Dragdrop script responsible for the main mechanic of the cash register which is the dragging of the notes and coins in order to make payment. |
| *Variable* | Private RectTransform rectTransform – Variable used to make dragging of gameobject smoother.  Private CanvasGroup canvasGroup – Variable used to make gameobjects alpha, translucent for aesthetic purposes. (To let player know that he/she is dragging the gameobject) |
| *Function* | Private void Awake() – used to call all the components such as rectTransform and canvasGroup for later use in the code  Public void OnBeginDrag() – This is used for when the player STARTS to drag the gameobject.  Public void OnDrag() – This is used DURING the dragging of the gameobjects, then the code below will be executed.  Public void OnEndDrag() – This method is used to when the player stops dragging the gameobject. |
| *Game Object attached to* | 50 Dollar  100 Dollar |
| *Author* | *Padilla Carl Jameson Z.* |

| *Component Name* | DontDeleteAudio |
| --- | --- |
| *Description* | *Allows audio to continue playing even through scenes* |
| *Variable* | *transform.gameObject* |
| *Function* | *Awake()* |
| *Game Object attached to* | *BGM and AudioSources* |
| *Author* | *Domenic* |

| *Component Name* | AudioSource |
| --- | --- |
| *Description* | *Contains audio* |
| *Variable* | *Add Item, Choose Character, Click Button, Hover Button, Lvl Complete, Open Map, Shop Cart, Electrodoodle, Wallpaper* |
| *Function* | *Play()* |
| *Game Object attached to* | *Add Item, Choose Character, Click Button, Hover Button, Lvl Complete, Open Map, Shop Cart, Electrodoodle, Wallpaper* |
| *Author* | *Domenic* |

| *Component Name* | EventTrigger |
| --- | --- |
| *Description* | *Plays audio when player hovers or clicks a button* |
| *Variable* | *AudioSource* |
| *Function* | *PointerEnter, PointerClick* |
| *Game Object attached to* | *BGM and AudioSources* |
| *Author* | *Domenic* |

# Game Objects

List all the game objects in the game.  
For each game object list the following information  
- Object Description  
- Which scene is it in  
- Which other game objects it contains  
- What components it contain

· Map Button

- *Description:* Displayed as a Map Icon, it allows players to click on it to display a map of the supermarket.

- *Scenes it is in:* In Level 1-15.

- *Other GameObjects it contains* : none.

- *Components*: Image, Button and MapOpen script.

· Map Panel

- *Description:* A panel, displayed with the respective Aisles as buttons on it which can be clicked.

- *Scenes it is in:* In Level 1-15.

- *Other GameObjects it contains* : Bakery, Canned Food, Drinks, Snacks, Fruits & Vegetable, Frozen Food, Dairy, Rice, Meat & Seafood, Cashier and Close Map Buttons.

- *Components*: Image.

| *GameObject Name* | FoodWaypoints  OrganicEggWP  LargeEggWP  FreshMilkWP  OatMilkWP  SoyaMilkWP  FullCreamMilkWP  DeluxeCheeseWP  StandardCheeseWP  CheddarCheeseWP  CreamCheeseWP  StandardCreamWP  ABCCreamWP  DeluxeCreamWP  WhippingCreamWP  FreshWholeChickenWP  CheeseSausageWP  DeluxeSmokedSausageWP  ChickenSausageWP  SmokedSausageWP  HoneybakedHamWP  HamWP  MincedChickenWP  ChickenCubeWP  BonelessChickenWP  ChickenThighWP  ConeIceCreamWP  JumboIceCreamWP  StandardPizzaDoughWP  ABCPizzaDoughWP  MiniPizzaDoughWP  DeluxePizzaDoughWP  FrozenFriesWP  StandardTomatoPasteWP  StandardTomatoPasteWP2  ABCTomatoPaste  ABCTomatoPaste2  ChocolateChipCookieWP  VanillaCreamCookieWP  OatCookieWP  PlainChipWP  BBQChipWP  SpicyChipsWP  StandardChocolateBarWP  DeluxeChocolateWP  CaramelChocolateWP  DarkChocolateWP  FineGrainBreadWP  MilkBreadWP  WhiteBreadWP  WholemealBreadWP  WhiteNoodlesWP  YellowNoodlesWP  EasyCookNoodlesWP  RiceNoodlesWP  RiceWP  StandardSoySauceWP  DeluxeSoySauceWP  WintermelonWP  ChickenSoupWP  ABCSoupWP  StandardFlourWP  ABCFlourWP  DeluxeFlourWP  CornFlourWP  SmokyBBQSauceWP  JapaneseBBQSauceWP  KoreanBBQSauceWP  StandardBBQSauceWP  StandardTomatoSauceWP  DeluxeTomatoSauceWP  OrganicTomatoSauceWP  JapaneseTomatoSauceWP  StandardSugarWP  ABCSugarWP  DeluxeSugarWP  CaneSugarWP  FineSaltWP  SeaSaltWP  OrganicSaltWP  PremiumSaltWP  StandardRedColouringWP  PremiumRedColouringWP  NaturalRedColouringWP  ABCRedColouringWP  StandardGreenColouringWP  PremiumGreenColouringWP  NaturalGreenColouringWP  ABCGreenColouringWP  StandardBlueColouringWP  PremiumBlueColouringWP  NaturalBlueColouringWP  ABCBlueColouring  FreshTomatoWP  CherryTomatoWP  JumboTomatoWP  OrganicTomatoWP  StandardLettuceWP  BabyLettuceWP  MixedLettuceWP  PremiumLettuceWP  FreshBroccoliWP  AustralianBroccoliWP  BroccoliCauliflowerWP  ReadyToCookBroccoliWP  MiniMushroomWP  StandardMushroomWP  BrownMushroomWP  ButtonMushroomWP  BagOf3BigOnionsWP  BagOf5SmallOnionsWP  Bagof10BigOnionsWP  BagOg10SmallOnionsWP  StandardCucumberWP  JapaneseCucumberWP  MiniCucumberWP  OrganicCucumberWP  StandardGingerWP  MalaysianGingerWP  OrganicGingerWP  ReadyToCookGingerWP  FreshMintLeafWP  MalaysianMintLeafWP  OrganicMintLeafWP  ChinaMintLeafWP  FreshPineappleWP  HoneyPineappleWP  SlicedPineappleWP  JumboPineappleWP  StandardOrangeWP  AfricanOrangeWP  JumboOrangeWP  BagOf5OrangesWP  FreshLemonWP  SouthAfricanLemonWP  JumboLemonWP  ABCLemonWP  GreenTeaWP  OrangeJuiceWP  LycheeTeaWP  LemonTeaWP  MiloPacketWP  RibenaPacketWP |
| --- | --- |
| *Description* | All these Game Objects act as waypoints for the food waypoint to spawn at. |
| *Scene* | All Levels |
| *Children* | N/A |
| *Components* | Item Controller  Item Script  Box Collider |

| *GameObject Name* | Floor |
| --- | --- |
| *Description* | This is for the flooring in Smart Shopper |
| *Scene* | All levels |
| *Children* | temporaryFloor@4x |
| *Components* | N/A |

| *GameObject Name* | Game Manager |
| --- | --- |
| *Description* | This gameObject handles most of the game processes that occur. |
| *Scene* | All levels |
| *Children* | N/A |
| *Components* | Objective  Tutorial Script  Map Location  NPC Spawning  Game Over Script  Setting Menu Handler  Map Data List |

| *GameObject Name* | Dialogue Manager |
| --- | --- |
| *Description* | Handles the NPC Dialogue from the NPC Interaction in the game |
| *Scene* | All Levels |
| *Children* | N/A |
| *Components* | Dialogue Manager |

| *GameObject Name* | World |
| --- | --- |
| *Description* | Holds all the asset in the level that is used for background design purposes |
| *Scene* | All levels |
| *Children* | DairyAisle (Respectively each contains a shelf and the food assets)  MeatSeafoodAisle (Respectively each contains a shelf and the food assets)  FrozenFoodAisle (Respectively each contains a shelf and the food assets)  DrinksAisle (Respectively each contains a shelf and the food assets)  CannedFoodAisle (Respectively each contains a shelf and the food assets)  SnacksAisle (Respectively each contains a shelf and the food assets)  RiceAisle (Respectively each contains a shelf and the food assets)  FruitsVegetableAisle (Respectively each contains a shelf and the food assets)  aisleSignboardLonger@4x |
| *Components* | aisleSignboardLonger@4x contains:  Sign Board Handler |

## 

| *GameObject Name* | Canvas |
| --- | --- |
| *Description* | Holds all the different UI Elements |
| *Scene* | All Levels |
| *Children* | Map  Cart\_Panel  ShoppingList  DynamicShoppingList  Map\_Btn  Cart\_Btn  MoveLeft  MoveRight  DialogueBox (unused and reworked)  NPCDialogueBox  LevelObj\_Panel  AdjacentAisles  SettingButton  TimerClock  TutorialBox  BudgetReminder  GrayedBackgroundMenu  GameOver |
| *Components* | Canvas  Canvas Scaler  Graphic Raycaster |

| *GameObject Name* | Setting Button |
| --- | --- |
| *Description* | Once the button is pressed, it would display the pause menu |
| *Scene* | All Levels |
| *Children* | N/A |
| *Components* | Onclick method taken from the game manager to make the pause menu active. |

| *GameObject Name* | TimerClock |
| --- | --- |
| *Description* | This gameObject is what holds the time and displays it in the scene. |
| *Scene* | All Levels |
| *Children* | -Timer (Text) |
| *Components* | N/A |

| *GameObject Name* | TutorialBox |
| --- | --- |
| *Description* | This gameObject holds all the different UI element needed for the tutorial |
| *Scene* | Only Level 1 |
| *Children* | * BakeryAisleBtn * TutorialMap2 * TutorialCart * TutorialMap * TutorialMovement * SpeechBubbleWaypoints * CoinWaypoints * CoinHolder * Dialogue * TutorialMessage * ContinueBtnTutorial |
| *Components* | N/A |

| *GameObject Name* | GrayedBackgroundMenu |
| --- | --- |
| *Description* | This gameObject is for the Menu UI where it would grey out the background and display the menu screen |
| *Scene* | All Levels |
| *Children* | * Image * ReturnToMenu * ResumeButton * CrossButton * MenuTxt * DeclineBtn * ConfirmBtn |
| *Components* | N/A |

| *GameObject Name* | GameOver |
| --- | --- |
| *Description* | This is the UI for the game over screen |
| *Scene* | All Levels |
| *Children* | * Image * MenuTxt * ConfirmBtn (1) * ConfirmBtn (2) * ConfirmBtn (3) |
| *Components* | N/A |

| *GameObject Name* | RandomEventHandler |
| --- | --- |
| *Description* | This gameObject will not destroyOnLoad and will be carried out through every scene. This game Object carries all the data needed to progress to the next level, restart etc. |
| *Scene* | All Scenes |
| *Children* | N/A |
| *Components* | Button Data Holder  Player Selection Option  Store Cart Data |

| *GameObject Name* | GoodHelperNPC |
| --- | --- |
| *Description* | This is a prefab for the good NPC in the game |
| *Scene* | All Levels |
| *Children* | * NPC (Sprite) |
| *Components* | N/A |

| *GameObject Name* | Scammer |
| --- | --- |
| *Description* | This is a prefab for the bad NPC in the game |
| *Scene* | All Levels |
| *Children* | * NPC (Sprite) |
| *Components* | N/A |

| *Game Object* | Cart\_Panel |
| --- | --- |
| *Description* | The panel to show or hide the shopping cart. Displays items the player has added to cart, the total cost of items and have the ability to remove unwanted items. |
| *Scene* | Game Levels ( Level1, Level2) |
| *Child* | Cart   * CartTitle\_Txt * CartItems * TotalAmountHelper * Close\_Btn * LevelBudget\_Txt |
| *Components* | NA |

| *Game Object* | CartTitle\_Txt |
| --- | --- |
| *Description* | To display “Cart” at the top of the panel |
| *Scene* | Game Levels |
| *Components* | TextMeshPro |

| *Game Object* | CartItems |
| --- | --- |
| *Description* | Where the displayed items prefab will be childed to. |
| *Scene* | Game Levels |
| *Child* | * Viewport   + Content     - Text Prefabs (instantiated) |
| *Components* | * Scroll rect (CartItems)   + Mask (ViewPort)     - Grid Layout group (Content)     - Content Size Fitter |

| *Game Object* | TotalAmountHelper |
| --- | --- |
| *Description* | Able to toggle the text to show the total cost of cart or hides it |
| *Scene* | Game levels |
| *Child* | * TotalAmt\_Txt * Show\_Btn |
| *Components* | * ShowCartTotalAmount (script)   + TextMeshPro (TotalAmt\_Txt)   + Button (Show\_Btn) |

| *Game Object* | Close\_Btn |
| --- | --- |
| *Description* | Closes the shopping cart panel |
| *Scene* | Game Levels |
| *Components* | Button |

| *Game Object* | LevelBudget\_Txt |
| --- | --- |
| *Description* | Shows the budget on that level |
| *Scene* | Game Levels |
| *Components* | TextMeshPro |

| *Game Object* | ShoppingList |
| --- | --- |
| *Description* | Displays the times required for that level |
| *Scene* | Game Levels |
| *Child* | * Viewport   + Content     - Prefabs to show items needed (instantiated) |
| *Components* | * Mask (Viewport)   + Content Size Fitter (Content)   + Grid Layout Group |

| *Game Object* | DynamicShoppingList |
| --- | --- |
| *Description* | Displays the items required for the level in the smaller and dynamic shopping list. If the item is in the aisle, show the food item at the top of the list. If there is nothing in the aisle needed, hides the list |
| *Scene* | GameLevel |
| *Child* | * Viewport   + Content |
| *Components* | * Mask (Viewport)   + Content Size Fitter (Content)   + Grid Layout Group |

| *Game Object* | Cart\_Btn |
| --- | --- |
| *Description* | Open or closes the shopping cart |
| *Scene* | GameLevels |
| *Components* | Button |

| *Game Object* | MoveLeft |
| --- | --- |
| *Description* | Button used to move left if player presses and holds it |
| *Scene* | GameLevel |
| *Components* | Button |

| *Game Object* | MoveRight |
| --- | --- |
| *Description* | Button used to move right if player presses and holds it |
| *Scene* | GameLevel |
| *Components* | Button |

| *Game Object* | LevelObj\_Panel |
| --- | --- |
| *Description* | Pops up the panel if the player tries to continue to the cash register when they do not have enough items for the objective or they went over budget |
| *Scene* | GameLevel |
| *Child* | * BodyHelpPanel\_Txt * Buck Buddy * Continue\_Btn * Hint\_Btn |

| *Game Object* | BodyHelpPanel\_Txt |
| --- | --- |
| *Description* | Used to inform the player if they either went over budget or are missing items |
| *Scene* | Game Levels |
| *Components* | TextMeshPro |

| *Game Object* | Buck Buddy |
| --- | --- |
| *Description* | An image to show the mascot of the company |
| *Scene* | Game Level |
| *Components* | Image |

| *Game Object* | Continue\_Btn |
| --- | --- |
| *Description* | To close the objective panel |
| *Scene* | Game Levels |
| *Components* | Button |

| *Game Object* | Hint\_Btn |
| --- | --- |
| *Description* | If the players need help on what items they need, the function will show the players what they are required to find. |
| *Scene* | Game levels |
| *Child* | Text (TMP) |
| *Components* | Button |

| *Game Object* | BudgetReminder |
| --- | --- |
| *Description* | A pop up that would remind players that they have come to close to the budget or they have already went over. |
| *Scene* | Game Levels |
| *Child* | * SpedGrowCoin\_img * Reminder\_Txt |
| *Components* | * Image (SpedGrowCoin\_img) * TextMeshPro (Reminder\_Txt) |

| *Game Object* | Character |
| --- | --- |
| *Description* | The game object that holds the components for the main character |
| *Scene* | Game Levels |
| *Child* | * Main Camera * MaleModel * FemaleModel |
| *Components* | * Capsule collider * Rigidbody * CharacterScript (script) * MapLoopTeleport (script)   + Animator (MaleModel and FemaleModel) |

| *Game Object* | Teleport Points around the map |
| --- | --- |
| *Description* | In the world gameobject, there will be the different aisles around the game. Inside those aisle there would be a empty child game object of the aisle name in there. This empty child is the teleport point. Used when the players presses a button on the map to teleport, teleports the player to these teleport point. |
| *Scene* | Game Levels |
| *Where* | World   * Aisle (Dairy Aisle, Drinks Aisle etc)   + Teleport point (will be the same name as parent) |
| *Components* | Tagged with “Waypoint” |

| *Game Object* | Border points |
| --- | --- |
| *Description* | Border points are used at the start and at the end of the aisle. Used to find where the player is located. |
| *Scene* | Game Levels |
| *Components* | Tagged with “Border” |

| *Game Object* | InvisBarrier |
| --- | --- |
| *Description* | Colliders at the start and the end of the level. Used to teleport players to the other side of the map if they walk far enough. |
| *Scene* | Game Levels |
| *Child* | * LeftBarrier   + LeftTpPoint * RightBarrier   + RightTpPoint |
| *Components* | * Collider (Barriers) |

| *Game Object* | InventoryManager |
| --- | --- |
| *Description* | Gameobject holding the inventory manager script. Used to handle most of the shopping cart and things relating to items. |
| *Scene* | Game Level |
| *Components* | InventoryManager (script) |

| *Game Object* | GameManager |
| --- | --- |
| *Description* | Empty gameobject that holds multiple scripts for the level |
| *Scene* | Game Levels |
| *Components* | * Objective (script) * MapLocation (script) * MapDataList (script) |

| *Game Object* | DisplayObject |
| --- | --- |
| *Description* | Prefab to instantiate to the cash register display to list the items there. |
| *Scene* | Used in “CashRegister”. In prefab folder |
| *Child* | * Quantity * FoodName * Price |
| *Components* | * Text (Quantity, FoodName, Price) |

| *Game Object* | DisplayObject |
| --- | --- |
| *Description* | Prefab to instantiate to the cash register display to list the items there. |
| *Scene* | Used in “CashRegister”. In prefab folder |
| *Child* | * Quantity * FoodName * Price |
| *Components* | * Text (Quantity, FoodName, Price) |

| *Game Object* | DisplaySmallerScreenCR |
| --- | --- |
| *Description* | Prefab to instantiate into the smaller display in the cash register |
| *Scene* | Used in “CashRegister”, in prefab folder. |
| *Child* | * Price * ItemName |
| *Components* | Text (Price, ItemName) |

| *Game Object* | DynmicShoppingCartItems |
| --- | --- |
| *Description* | Used to display food items in the dynamic shopping list. Has a line to cross item out after getting enough items |
| *Scene* | Used in game levels, a prefab. |
| *Child* | * QuantityCart\_Txt * FoodType\_Img * Line |
| *Components* | * TextMeshPro * Image (FoodType\_Img, Line) |

| *Game Object* | ItemListed |
| --- | --- |
| *Description* | Used to display the items added to cart in the shopping cart panel |
| *Scene* | Used in game levels, a prefab. |
| *Child* | * QuantityCart\_Txt * FoodType\_Img * Line |
| *Components* | * TextMeshPro * Image (FoodType\_Img, Line) |

| *Game Object* | ShoppingCartItems |
| --- | --- |
| *Description* | A prefab that shows what items is needed in the shopping list. |
| *Scene* | Used in game levels, a prefab. |
| *Child* | * QuantityCart\_Txt * FoodNameCart\_Txt * FoodType\_Img * LineCross |
| *Components* | * TextMeshPro (Quantity * Image (FoodType\_Img, LineCross) |

CASH REGISTER SCENE

| *Game Object Name* | Main Camera |
| --- | --- |
| *Object Description* | Used to show the visuals of the game |
| *Which scene is it in* | CashRegister |
| *Which other game objects it contains* | - |
| *What components it contain* | Camera Component, Audio Listener |

| *Game Object Name* | Canvas |
| --- | --- |
| *Object Description* | It is to show the main components of the Cash Register Scene. All the main functions and mechanics are contained in this game object. It is the parent of a lot of GameObjects. |
| *Which scene is it in* | CashRegister |
| *Which other game objects it contains* | * CashRegisterBG * Monitor * NTUC Worker * Icon * Title * Money Generated * Payment * 1DollarSpawner * 50cSpawner * 20cSpawner * 10cSpawner * 5cSpawner * 50 Dollar * 100 Dollar * DragText |
| *What components it contain* | Rect Transform, Canvas, Canvas Scaler, Graphic Raycaster, Display Food Items Script |

| *Game Object Name* | Display 1 |
| --- | --- |
| *Object Description* | Used to display the items the player has purchased |
| *Which scene is it in* | Cash Register |
| *eWhich other game objects it contains* | * Scroll View * Viewport * Content |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Default UI Material |

| *Game Object Name* | Display 2 |
| --- | --- |
| *Object Description* | Used to display the items the player has purchased for further verification |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | * Scroll View * Viewport * Content |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Default UI Material |

| *Game Object Name* | CPs |
| --- | --- |
| *Object Description* | Used as set positions for the cash and coins when it is dragged into the player system. Its basically a sorting system for cash and coins |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | * 2DollarCP * 5DollarCP * 10DollarCP * 50DollarCP * 100DollarCP * 5cCP * 10cCP * 20cCP * 50cCP * 1DollarCP |
| *What components it contain* | Rect Transform |

| *Game Object Name* | 2DollarSpawner |
| --- | --- |
| *Object Description* | A prefab that is used to spawn a random amount of 2 Dollar Notes according to the budget. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | 2 Dollar |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Cash Spawner Script |

| *Game Object Name* | 5DollarSpawner |
| --- | --- |
| *Object Description* | A prefab that is used to spawn a random amount of 5 Dollar Notes according to the budget |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | 5 Dollar |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Cash Spawner Five |

| *Game Object Name* | 10DollarSpawner |
| --- | --- |
| *Object Description* | A prefab that is used to spawn a random amount of 10 Dollar Notes according to the budget |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | 10 Dollar |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Cash Spawner Ten |

| *Game Object Name* | Confirm |
| --- | --- |
| *Object Description* | It is a Confirm button used to confirm the payment after dragging a certain amount of money to pay for the items. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Pay Text, Fill Amount |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button, Animator |

| *Game Object Name* | Reset |
| --- | --- |
| *Object Description* | It is a Reset button if the player wants to restart the canvas if they think they paid too much for the items. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Reset Money Text |
| *What components it contain* | Rect Transform, Canvas Renderer, Imager, Button, Default UI Material |

| *Game Object Name* | Progress Bar |
| --- | --- |
| *Object Description* | A Visual representation of how much the player needs to pay the correct amount or whether the player has paid to little or too much for the items that has been purchased |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Mask, Fill, Fill Amount |
| *What components it contain* | Rect Transform, Canvas Renderer, Imager, Default UI Material |

| *Game Object Name* | ReturnToGame |
| --- | --- |
| *Object Description* | It basically returns the player to the scene if he/she has missed to purchase an item. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Return Text |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button, Button Press Functions Script |

| *Game Object Name* | TimerClock |
| --- | --- |
| *Object Description* | Used to keep track of how long a player takes to purchase and pay for items in that level. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Timer |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Game Over Script, Default UI Material |

| *Game Object Name* | ScoreBoard |
| --- | --- |
| *Object Description* | It is to show the player score, in this case we show the score in stars, where 1 star is the lowest score you can get and 3 stars is the highest score you can achieve. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | Star, 2nd Star, 3rd Star, Message, Timer Report |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button Press Functions |

| *Game Object Name* | Next Level |
| --- | --- |
| *Object Description* | A button used to bring the player to the next level of the game |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | NextLevelTxt |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button (Scoreboard Script and Payment Script) |

| *Game Object Name* | Restart |
| --- | --- |
| *Object Description* | A button used to restart the game |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | RestartTxt |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button (Scoreboard Script and Payment Script) |

| *Game Object Name* | Menu |
| --- | --- |
| *Object Description* | A button used to bring the player back to the main menu |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | MenuTxt |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button (Scoreboard Script and Payment Script) |

| *Game Object Name* | Event System |
| --- | --- |
| *Object Description* | A Game Object that is needed to make the canvas function properly |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | - |
| *What components it contain* | Transform, Event System, Standalone Input Module |

| *Game Object Name* | Budget Manager |
| --- | --- |
| *Object Description* | It is used for the budgeting system of the game according to each level. |
| *Which scene is it in* | Cash Register |
| *Which other game objects it contains* | - |
| *What components it contain* | Transform, Cash Budget Script |

LEVEL SELECT TEST SCENE

| *Game Object Name* | Main Camera |
| --- | --- |
| *Object Description* | Used to show the visuals of the game |
| *Which scene is it in* | Level Select Test |
| *Which other game objects it contains* | - |
| *What components it contain* | Camera Component, Audio Listener |

| *Game Object Name* | Canvas |
| --- | --- |
| *Object Description* | It is to show the main components of the Level Select Scene. All the main functions and mechanics are contained in this game object. It is the parent of a lot of GameObjects. |
| *Which scene is it in* | Level Select Test |
| *Which other game objects it contains* | BG, Level Select Txt, Levels |
| *What components it contain* | Canvas, Canvas scaler, Graphic Raycaster |

| *Game Object Name* | Level 1 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 1 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, Ham Sandwich |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 2 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 2 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, EggCheeseSandwich |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 3 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 3 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, FriedNoodles |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 4 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 4 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, ChickenSoup |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 5 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 5 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, CheeseBurger |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 6 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 6 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, MushroomBurger |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 7 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 7 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, BBQChickenPizza |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 8 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 8 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, MushroomPizza |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 9 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 9 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, ChocolateCake |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 10 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 10 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, PineappleCitrusJuice |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 11 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 11 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, SausageEggSandwich |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 12 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 12 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, VegetableFruitSalad |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 13 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 13 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, Chicken Sandwich |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 14 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 14 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, Spaghetti |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

| *Game Object Name* | Level 15 |
| --- | --- |
| *Object Description* | It’s the button to indicate level 15 and the food you are buying for. |
| *Which scene is it in* | LevelSelectTest |
| *Which other game objects it contains* | LvlTxt, RainbowCake |
| *What components it contain* | Rect Transform, Canvas Renderer, Image, Button |

## 

## AudioSources

Empty GameObject containing gameobjects with audio sources

In: MainMenu

Contains (GameObject): BtnHover, BtnClick

Contains (Component): DontDestroyAudio

## BtnHover

GameObject containing AudioSource

In: MainMenu

Contains (GameObject):

Contains (Component): AudioSource (Hover Button)

## BtnClick

GameObject containing AudioSource

In: MainMenu

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## BGM

GameObject containing AudioSource

In: MainMenu

Contains (GameObject):

Contains (Component): AudioSource (Electrodoodle), DontDestroyAudio

## StartGame

Button with EventTrigger

In: MainMenu

Contains (GameObject):

Contains (Component): EventTrigger

## Settings

Button with EventTrigger

In: MainMenu

Contains (GameObject):

Contains (Component): EventTrigger

## QuitGame

Button with EventTrigger

In: MainMenu

Contains (GameObject):

Contains (Component): EventTrigger

## AudioSources

Empty GameObject containing gameobjects with audio sources

In: PlayerSelect

Contains (GameObject): Hover Button, Click Button, Choose Character

Contains (Component): DontDestroyAudio

## Hover Button

GameObject containing AudioSource

In: PlayerSelect

Contains (GameObject):

Contains (Component): AudioSource (Hover Button)

## Click Button

GameObject containing AudioSource

In: PlayerSelect

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## Choose Character

GameObject containing AudioSource

In: PlayerSelect

Contains (GameObject):

Contains (Component): AudioSource (Choose Character)

## Male

Button with EventTrigger

In: PlayerSelect

Contains (GameObject):

Contains (Component): EventTrigger

## Female

Button with EventTrigger

In: PlayerSelect

Contains (GameObject):

Contains (Component): EventTrigger

## Continue\_Btn

Button with EventTrigger

In: PlayerSelect

Contains (GameObject):

Contains (Component): EventTrigger

## AudioSources

Empty GameObject containing gameobjects with audio sources

In: LevelConfirm

Contains (GameObject): Hover Button, Click Button, Choose Character

Contains (Component): DontDestroyAudio

## Hover Button

GameObject containing AudioSource

In: LevelConfirm

Contains (GameObject):

Contains (Component): AudioSource (Hover Button)

## Click Button

GameObject containing AudioSource

In: LevelConfirm

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## Back\_Btn

Button with EventTrigger

In: LevelConfirm

Contains (GameObject):

Contains (Component): EventTrigger

## Play\_Btn

Button with EventTrigger

In: LevelConfirm

Contains (GameObject):

Contains (Component): EventTrigger

## AudioSources

Empty GameObject containing gameobjects with audio sources

In: Level1

Contains (GameObject): Hover Button, Click Button, Open Map. Shop Cart

Contains (Component):

## Hover Button

GameObject containing AudioSource

In: Level1

Contains (GameObject):

Contains (Component): AudioSource (Hover Button)

## Click Button

GameObject containing AudioSource

In: Level1

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## Open Map

GameObject containing AudioSource

In: Level1

Contains (GameObject):

Contains (Component): AudioSource (Open Map)

## Shop Cart

GameObject containing AudioSource

In: Level1

Contains (GameObject):

Contains (Component): AudioSource (Shop Cart)

## BGM

GameObject containing AudioSource

In: MainMenu

Contains (GameObject):

Contains (Component): AudioSource (Wallpaper), DontDestroyAudio

## Map\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## Cart\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## SettingButton

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ReturnToMenu

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ResumeButton

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## DeclineBtn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ConfirmBtn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ConfirmBtn (1)

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ConfirmBtn (2)

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## ConfirmBtn (3)

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## BakeryAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## DrinksAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## SnacksAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## FruitsVegetableAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## FrozenFoodAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## DairyAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## CannedFoodAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## MeatSeafoodAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## Cashier\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## RiceAisle\_Btn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## YesBtn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## NoBtn

Button with EventTrigger

In: Level1

Contains (GameObject):

Contains (Component): EventTrigger

## AudioSources

Empty GameObject containing gameobjects with audio sources

In: CashRegister

Contains (GameObject): BtnHover, BtnClick

Contains (Component): DontDestroyAudio

## Hover Button

GameObject containing AudioSource

In: CashRegister

Contains (GameObject):

Contains (Component): AudioSource (Hover Button)

## Click Button

GameObject containing AudioSource

In: CashRegister

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## LvlCompleteSFX

GameObject containing AudioSource

In: CashRegister

Contains (GameObject):

Contains (Component): AudioSource (Click Button)

## Confirm

Button with EventTrigger

In: CashRegister

Contains (GameObject):

Contains (Component): EventTrigger

## Reset

Button with EventTrigger

In: CashRegister

Contains (GameObject):

Contains (Component): EventTrigger

## ReturnToGame

Button with EventTrigger

In: CashRegister

Contains (GameObject):

Contains (Component): EventTrigger

## NextLevel

Button with EventTrigger

In: CashRegister

Contains (GameObject):

Contains (Component): EventTrigger

## Restart

Button with EventTrigger

In: CashRegister

Contains (GameObject):

Contains (Component): EventTrigger

## Menu

Button with EventTrigger

In: CashRegister

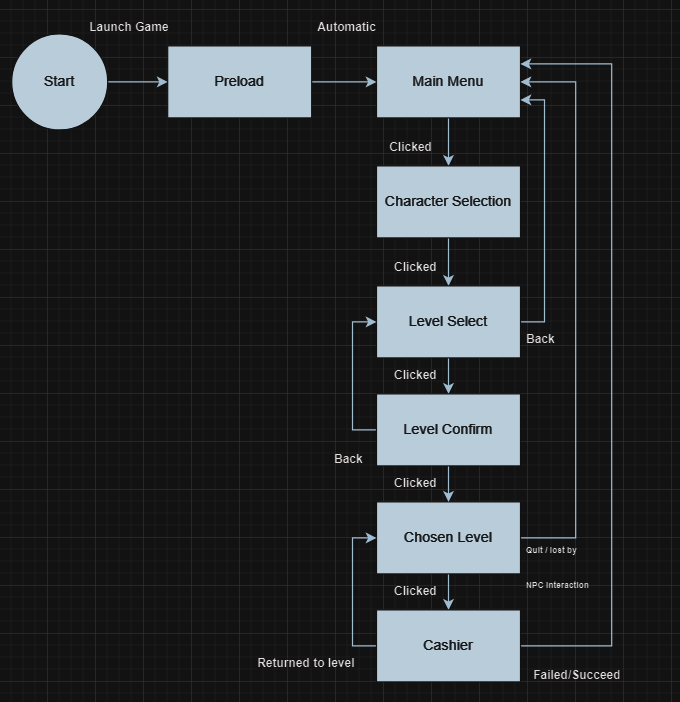
Contains (GameObject):

Contains (Component): EventTrigger

# Game Flow

## Flow Chart Diagram

***Have a flow chart that indicate from the start of the game which scene it being with.  
For each scene what event can happen and each event will lead to what other actions or scene.***



# Development information

## Library

Lists all external library that you have use. What the licensing agreement and where is the source

WebGL

## Version of software

Lists all the software and its version used to develop this game. For example which Unity version is being used etc

Unity 2022.3.3f1 <DX11>

## Hardware requirement

List all the hardware and their specification to run the game

PC

Tablet/Mobile

## Setup of development platform

## Detailed instruction on how to setup the development platform.

To setup the development platform you must launch Unity Hub and Github desktop and import the project from there and launch the project from unity hub.

**APPENDIX: Code base location**

*Where is each of the scripts being located in the project*

| *Scripts* | AisleData (Removed)  ItemData  LevelData  NPCData  SignboardData  SignBoardHandler  TimeData |
| --- | --- |
| *Location* | Assets/Assets/Scripts/Data |

| *Scripts* | DialogueHandler (Removed)  GameOverScript  NPCRandomChance (Removed)  VisualizationScript (Removed) |
| --- | --- |
| *Location* | Assets/Assets/Scripts/Gameplay |

| *Scripts* | Dialogue (Removed)  DialogueManager  DialogueTrigger  NPCSpawning |
| --- | --- |
| *Location* | Assets/Assets/Scripts/NPCDialogue |

| *Scripts* | TutorialScript |
| --- | --- |
| *Location* | Assets/Assets/Scripts/TutorialScript |

| *Scripts* | WorldButtonClicked  ButtonDataHolder  ButtonPressFunctions  SettingMenuHandler |
| --- | --- |
| *Location* | Assets/Assets/Scripts/UI Scripts |

Cash Register Scripts – Located under the folder called Scripts and under the folder called “Cash Register Scripts” are the cash register scripts

Level Select Script – Located under the folder called "Assets” and under the “UI Scripts” Folder is the ButtonClicked script used for level Select.

*DontDeleteAudio*

*Assets\Assets\Scripts\Audio Scripts*