

Assignment 4 Documentation

1. Introduction

- **Brief Overview:**

Pixel Jackpot is a 2D casino-themed game with a pixel art aesthetic. The player is given objectives from the start, following these objectives allows the player to explore the entire casino. The player gets to explore the casino completing challenges playing various slot machines, Blackjack tables, Roulette Tables, interacting with NPCs, and purchasing items from the shop.

- **Game Goal:**

- *Win*: Complete all objectives.
- *Lose*: No final Lose condition as the player is free to play games with their money if they run out they can earn more money, but each game has its own Lose conditions

- **Controls:**

- **Movement**: WASD
- **Interact with Machines/Tables/NPC/Objects**: Spacebar once in range
- **Pause Menu**: ESC

- **Progression and Features:**

- VIP Lounge
- Higher Stakes slot machines
- BlackJack Tables
- Roulette Tables
- NPCs
- Objectives
- Item Shop

2. Summary of Written Scripts

Provide a short description of each important script and its purpose in the game. For example:

Scripts Utilized in project 2 submission:

1. **Cashier.cs** – manages the interaction between the player and the in-game store. It opens the store UI when the player is nearby and interacts, allowing them to purchase VIP access if they have enough money. Once VIP access is bought, the script updates the player's status, removes the VIP barrier, disables the VIP button, and saves

progress. It also controls the player's movement and provides feedback for insufficient funds or successful purchases.

2. **GameInfoManager.cs** – handles the display of the game's rules panel. It starts with the panel hidden and uses button listeners to show or hide the rules when the respective buttons are clicked. The ShowRules method activates the panel, while the CloseRules method hides it.
3. **Interactable.cs** – abstract class that serves as a base for other interactable objects in the game. It defines an abstract Interact method, which must be implemented by child classes to specify their unique interaction behavior.
4. **MenuManager.cs** – manages the main menu functionality. It assigns a listener to the Play button to load the game scene when clicked and provides a QuitGame method to exit the game, which works only in the built version.
5. **MoneyDisplay.cs** - updates the on-screen display of the player's current money in real-time. It retrieves the player's money from the playerController and updates the TMP_Text UI element with the formatted value each frame.
6. **PauseManager.cs** - controls the game's pause menu, allowing players to pause and resume gameplay by pressing ESC. It manages the game's time scale to freeze or unfreeze the game and provides functionality to save progress, navigate to the main menu, open settings, and quit the game.
7. **PlayerController.cs** - manages the player's movement, animations, and interactions within the game. It tracks the player's money, VIP access status, and game over conditions. It allows the player to earn or spend money and saves progress using PlayerPrefs. The script also handles movement control, switching animations based on direction, and displaying objectives such as purchasing VIP access. Additionally, it includes methods for resetting the game and triggering a game over state when the player runs out of money.
8. **ReelManager.cs** - handles the logic for spinning slot machine reels, displaying symbols, and rewarding the player based on matching symbols. It deducts a spin cost, animates reel changes, and determines the outcome after the spin duration. If two or three symbols match, a payout is awarded based on defined multipliers. The script also updates the player's money and displays win messages when applicable. If the player lacks sufficient funds, the spin is prevented.
9. **SettingsManager.cs** – controls the display of the settings panel. It starts with the panel hidden and uses button listeners to show or hide the settings when the respective buttons are clicked. The ShowSettings method activates the panel, while the CloseRules method hides it.
10. **SlotMachineInteract.cs** - manages the interaction between the player and a slot machine. It opens the slot machine UI when the player is nearby and presses Space. The player's movement is disabled while the slot machine UI is active. The script triggers the reel spin via the ReelManager when the spin button is clicked and hides the UI when finished. It also uses collision and trigger events to detect when the player enters or leaves the slot machine's range, displaying interaction prompts accordingly.

Scripts Utilized in project 3 submission:

1. **BJInteract.cs** - manages player interaction with a Blackjack table, displaying prompts when the player is nearby. If the player presses Space while in range, it opens the Blackjack UI panel, disables player movement, and completes a specific objective if conditions are met. The script also includes methods to toggle an information panel and close the Blackjack panel, re-enabling player movement afterward.
2. **CardScript.cs** - manages individual card properties and visuals in a card game. It stores the card's value and image, allowing the value and image to be set or retrieved. The ResetCard method resets the card to its back image and default value, referencing the deck's back card sprite.
3. **deck.cs** - handles the deck of cards in the game, including shuffling and dealing cards. It initializes card values, assigns face values (with face cards set to 10), and shuffles the deck by swapping cards randomly. The DealCard method assigns the next card in the deck to a specified card object, reshuffling if all cards have been dealt. The GetCardBack method provides the card back image.
4. **GameManager.cs** - controls the main logic of a Blackjack game, managing player and dealer actions, bets, and game state. It initializes and updates UI elements for player balance, bet amount, and hand totals, and handles button interactions for actions like hitting, standing, doubling down, and betting. The game flow includes shuffling the deck, dealing cards, and determining the winner, with balance adjustments based on game outcomes and messages displayed for results.
5. **GameScript.cs** - manages an individual Blackjack player's hand and balance, including drawing and resetting cards. The StartHand method initiates the hand with two cards, while GetCard draws additional cards, updating the hand value and tracking aces for value adjustment. The CheckAce method optimizes ace values between 1 and 11 as needed, and ResetGame clears the hand for a new round by resetting card visuals and values.
6. **GameAudioManager.cs** - manages background music and sound effects, including volume control via sliders and saving settings with PlayerPrefs. Ensures only one instance exists and plays game music on startup, with methods to adjust and play sound effects.
7. **MenuAudioManager.cs** - controls menu music volume, initializing from saved settings and allowing adjustments through a slider. Plays menu music on startup if not already active.
8. **SlotMachineFree.cs** - manages interaction with a free slot machine, enabling the slot machine UI and disabling player movement upon interaction. Displays prompts when the player is nearby and updates objectives if specific conditions are met.
9. **Objective.cs** - defines an objective with a title, description, reward amount, and completion status. Provides a method to mark the objective as complete. These are utilized in the player controller to handle displaying objectives.

New Scripts for Project 4:

1. **DataSaving.cs** - defines a serializable structure to save item data, including the item's name and remaining time.
 2. **NPCInteract.cs** - manages interactions with an NPC, displaying relevant panels such as messages, game info, and game UI. It enables or disables player movement during interactions and shows prompts when the player is within range. Methods handle opening and closing specific panels as needed.
 3. **CoinToss.cs** - implements a coin toss game where players can bet on heads or tails. It manages bets, simulates a coin flip with animations, and calculates payouts for correct guesses. The script updates UI elements such as the bet amount, result, and payout messages, and resets the game for the next round.
 4. **RouletteInteract.cs** - handles player interaction with the roulette table, enabling or disabling the roulette UI and player movement during interaction.
 5. **RouletteLogic.cs** - manages the logic for a roulette game, allowing players to place bets on various outcomes, spin the wheel, and calculate payouts based on the winning number. It handles UI updates for bets, results, and payouts, includes animations for the spinning wheel, and resets the game after each round.
 6. **VIPGateInteract.cs** - manages player interaction with the VIP gate, allowing players to purchase VIP access if they have sufficient funds.
 7. ***Updated CashierInteract.cs** - added new functionality to handle the new items.
 8. ***Updated PlayerController.cs** - added item functionality, includes 3 new items with timers that save when the game is quit.
-

3. Important Game Objects and Their Interaction

- **Player GameObject**: Moves around the casino, interacts with slot machines, tables, and the cashier, and has prefs saved to store money, item status, and location in the game.
- **Slot Machines(Free, Starter, and VIP, Elemental)**: Interact with the player when touched, reel manager to handle the spinning of the symbols and payout, and UI components like win_text to display if the slot machine won.
- **Black Jack Manager and BlackJack Table**: Interact with the player when touched, has a script derived from interactable, and utilizes CardScript, deck, GameScript, and GameManager to handle logic and UI.
- **Roulette Manager and Roulette Table**: Interact with the player when touched, has a script derived from interactable, and uses the RouletteLogic script to handle all game logic and payouts.
- **Objectives**: Holds the UI elements for the objective bar and the objective complete popup. Connects with interactables and the player to handle the logic for completing and rewarding objectives.
- **ReelManager**: Interacts with slot machine to spin symbols and player controller to give payouts

- **PauseCanvas:** Allows the player to save the game, return to the main menu, or adjust settings.
 - **Cashier:** Interacts with player to open buy menu and to adjust values in the players pref, adjusts players speed, costs of items and payouts depending on what items they bought from the shop.
 - **Tile Maps:** Tilemaps interact with the player and other objects depending on if they should have colliders/be interactable.
 - **NPC:** Interacts with the player and displays a message and options to play a game with them using the CoinToss.cs script.
 - **CoinToss:** new game canvas that is interactable by player.
 - **AudioManager:** One inside menu scene and one inside game scene, managers the sources for all the audio in the scene and adjusts the volumes based on sliders. Interacts with the player by storing the saved audio in prefs.
-

4. Project Requirements and How They Are Met

Explain how each of the required functionalities is implemented:

- **Player Input that is explained to the player:**

The first time loading the game an information panel displays explaining the objectives of the game. Each casino game has an information panel that explains the game, the payouts, and the cost. There are objectives that walk the player through the entire casino and all of the games, each objective has a description and a reward to incentivize progression.

- **Saving and Loading used for settings and gameplay:** Save game system stores current money, unlock status, item status, objective status, and player position; loading restores the saved state. Settings are also saved from the volume of different audio sources.
- **Multiple sounds:** Menu music, game music, and sound effects for different games(slots, black jack, roulette, and coin toss).
- **Settings (Volume):** Adjustable menu music volume, game music volume, and sound effects volume.
- **Multiple animated objects:** Player uses an animation controller for walking, the Slot machine uses animations for spinning, Roulette uses an animation for switching numbers, and coin toss uses an animation for flipping the coin.
- **Main Menu & Gameplay Scene, Pausing:**
 - Main menu with "Start Game", "Settings", "Game Info" and "Exit Game"
 - Casino gameplay scene where the player interacts with slot machines, tables and the shop, uses items, navigates the casino, and completes objectives.
 - Pause menu accessible during gameplay.
- **A more complicated scope of gameplay features than Assignment 3:**

New Features:

1. **Player items** - The item shop now has 3 purchasable items, Cost Reduction, Speed Boost, and Bonus Rewards. Each item has a timer and can't be purchased again until the timer expires. Each item has its own unique feature, Cost Reduction cuts all Casino's costs in half for 120 seconds, Speed Boost increases the player's speed by 50% for 120 seconds, and Bonus Rewards gives an extra 50% bonus payout for each win.
 2. **Roulette Game** - Brand new Casino game Roulette, features its own gameObject that is interactable, a game panel, an info panel, and all the logic that goes into Roulette. The player can bet on 1st, 2nd, 3rd, red, black, or green.
 3. **NPC** - Interactable NPC, provides a message with different response options.
 4. **Coin Toss Game** - Accessible through the NPC, new game has its own game panel, logic, and info panel. The game allows users to bet against the NPC in a simple coin toss game.
 5. **New Sound Effects** - Both new games, Roulette and Coin Toss feature new sound effects.
 6. **Advanced Saving** - Saving is now done using JSON for items, this allows me to save the item's status and timer and load it accordingly when the game is opened.
 7. **More Objectives** - With the addition of new games, the objectives were updated with objectives for each game.
 8. **New Slot** - Inside the VIP Lounge features a new slot machine, Elemental. This machine has a custom panel and new symbols.
 9. **Bug fixes/UI Changes** - Fixed many bug issues and updated some UI elements to look more sleek and less clumpy.
- **Emergent Gameplay/Progression:**

The game has objectives that navigate you through the entire casino. The next objective is locked until the completion of the following one. The objectives have rewards and progress you to different areas of the casino. The game also has items that can help you progress faster through the game which includes, speed boosts, cost cuts, and bonus rewards.

- **Win/Lose Conditions:**

Each game has its own win-and-lose conditions

Slot Machines you either get 3 symbols in a row and win or 2 for some symbols to get a small win or you lose.

Blackjack Table you either win the hand or you lose the different conditions including dealer/player busting, player/dealer blackjack, player higher hand, and player lower hand.

In roulette, you either win depending on what bet options you chose(you can bet on multiple items at once) or lose if none of your bets hit.

Coin Toss you either picked the right side(heads or tails) resulting in a win otherwise a loss.

5. List and Summary of Scenes

Describe each scene in your game:

- **Main Menu:** Provides options to Start/Load the game, view game info, settings to adjust volume, and exit the game.
 - **Casino Gameplay Scene:** Contains the main casino tilemap, decorations tilemap, slot machines, blackjack table, roulette tables, NPCs, CoinToss, player character, other sprites in the game, other UI elements, and the VIP lounge.
-

6. Referenced Material

List any assets, tutorials, or guides used, and how they were integrated:

- **Assets Used:**
 - Casino Tilemap:
<https://gamebetweenthelines.itch.io/2d-top-down-pixel-art-tileset-casino>
 - PixelArt Characters: <https://pixel-boy.itch.io/ninja-adventure-asset-pack>
- **Music & Sound Effects:**
 - Local Forecast - Elevator by Kevin Macleod: [Local Forecast - Elevator](#)
 - Slot Machine Reel Spin:
<https://www.zapsplat.com/music/fruit-machine-sound-win-or-spin-tone-2/>
 - Deal Card:
<https://www.zapsplat.com/music/large-thick-playing-card-set-down-single-deal-2/>
 - Complete Objective:
<https://www.zapsplat.com/music/game-sound-award-hit-win-coins/>
 - Roulette Sound:
 [Sounds Effects - Roulette game \(Roleta jogo\)](#)
 - Coin Toss Sound:
 [Fear and Hunger Coin Flip/Toss Sound effect](#)
- **Tutorials & Scripts:**
 - [How to Create 2D Tilemaps in Unity 2024](#) (Referenced to design tilemap)
 - [How to Create 2D Top Down Movement in Unity 2024](#) (Slightly modified his movement script)

- [!\[\]\(eae20f1adff742df783f6f7e3bbe72d1_img.jpg\) Adding Collisions For Your Tilemaps and 2D Character - Unity Tutorial \(Guide\)](#)
 - [!\[\]\(43c6e08c5a1618d745b54da5c843274e_img.jpg\) EVERYTHING you need to know to SAVE and LOAD your game in Unity \(Guide\)](#)
-

7. Source Code

Linked as a ZIP inside deliverables.