

SHADOWJACK

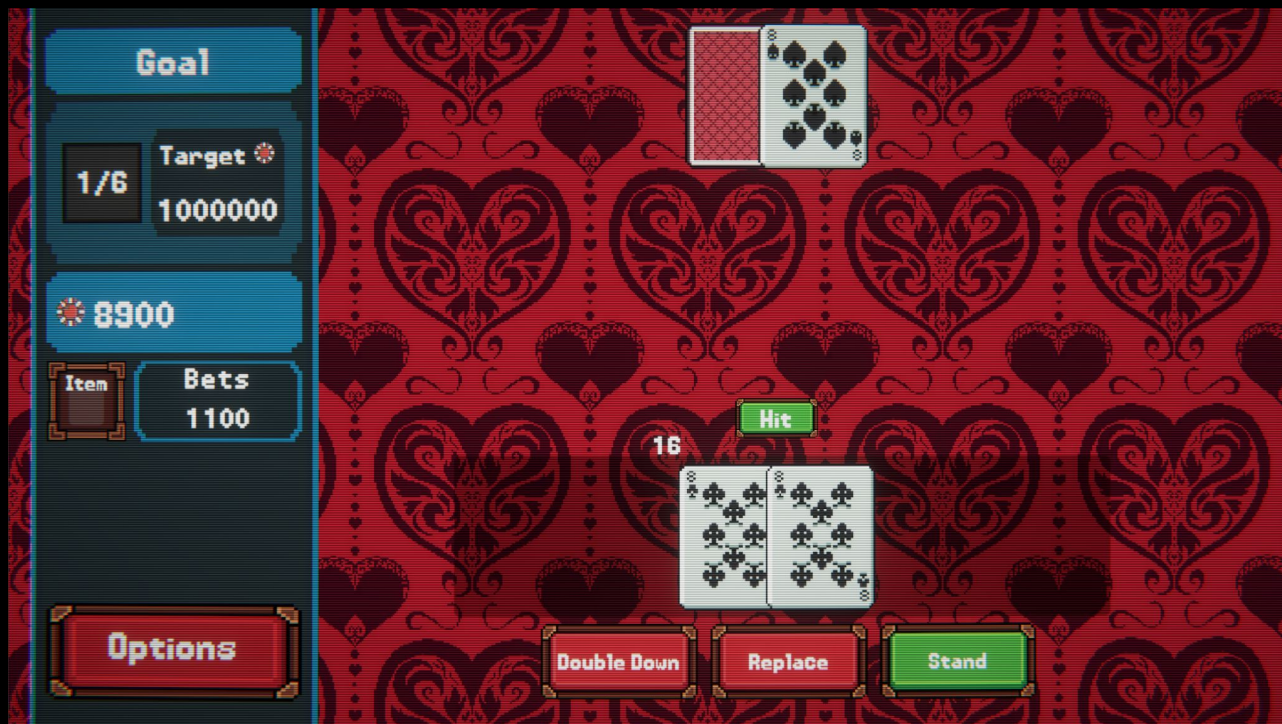


Object of the Game

Each participant attempts to beat the dealer by getting a count as close to 21 as possible, without going over 21.



Your
currency,
also
doubles as
a score,
reach the
target
before the
rounds
reach 6



Enemy's hand

Your hand



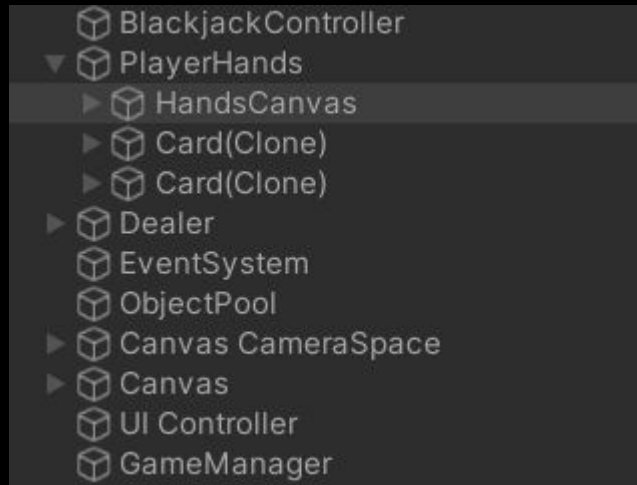
Hit: Draw one card to add to your total. But beware—if your total exceeds the Burst Limit, you **lose the round**.

Replace: Call for a replacement! Discard one of your cards, and draw a random card from the deck. Note: Replacement cards are guaranteed to have at least 5 copies still remaining in the deck.

Double Down: Double your bet and draw one final card. Your turn ends automatically, and your result is compared with the dealer's.

The game has a lot of empty game objects that are entirely focused on taking care of different aspects of the gameplay independently. They constantly update each other as well.

Being a card game, it makes use of Lists and Arrays frequently

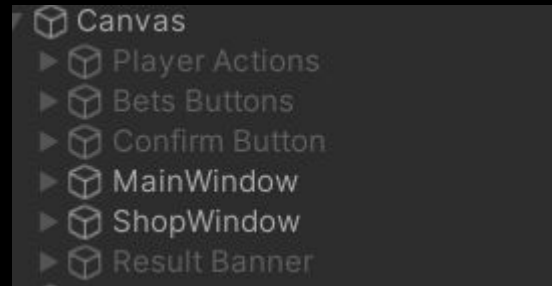
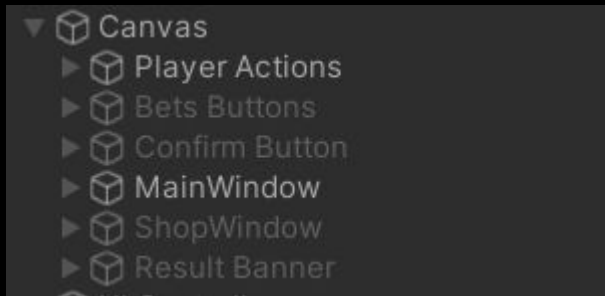


Script that handles the player and dealer hand

Script that handles the player and dealer hand

Events like clicking on buttons or cards

Game manager script that contains all the game info such as the tokens and the rounds



The game barely uses different scenes, the shop and the game itself and the lose screen are all in the same scene, just waiting to be activated or deactivated accordingly

Array and List mod



Its hard to know what the enemy's cards are during debugging, so I added code that allows us to do so!

Make an array version of the card List (we won't change the list because its better to keep the dynamic size)

```
public Card[] CardsArray => cards.ToArray();  
#endregion
```

Add the actual method at the bottom, itll just be using the premade card structure like rank for the names

```
public void DebugPrintCardsArray()  
{  
    // we bring the newly added array here  
    Card[] cardArray = CardsArray;  
  
    // make a string thats just the card rank so we can clearly read the enemy cards from the debug  
    string output = string.Join(", ", cardArray.Select(card => card.rank.ToString()).ToArray());  
    Debug.Log("Cards Array: " + output);  
}
```

Call it anywhere! here we put it in the addCard function to auto call it everytime the hand gets a new card

```
DebugPrintCardsArray();
```



Cards Array: Jack, Queen



Cards Array: Four, Three



Cards Array: Jack, Two

Difficulties:

- Code is impossible to penetrate at my current level, i couldnt make sense of anything, it is a miracle i was able to make a debugging tool
- The game itself was well contained, as in i didnt really have room to think of things to add or bugs to solve, it served its purpose

QNA

