Hitman card game

Game Design Document

Lucas Workman

Overview

“Hitman card game” is a 2d game built using pygame and the SimpleGE module

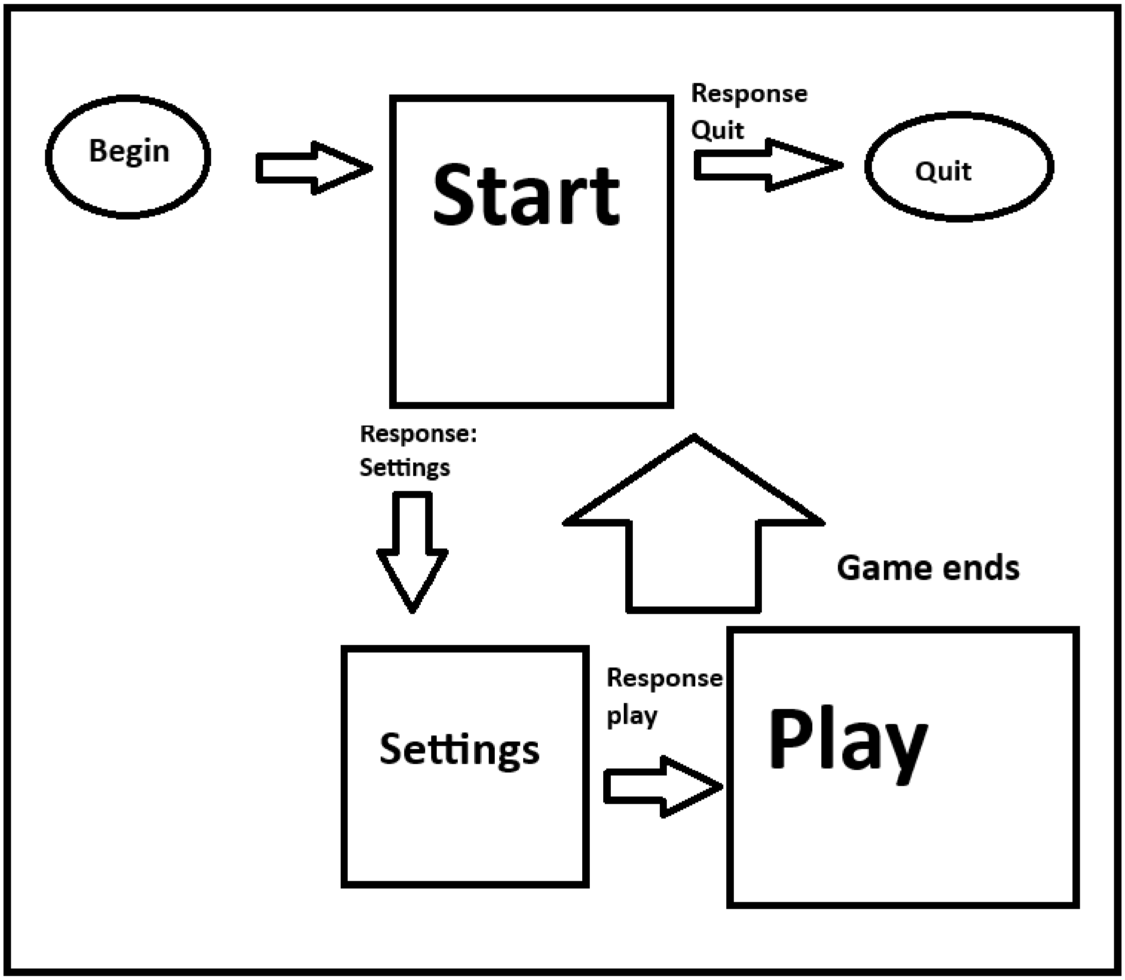
This is the idea behind the game: The game loads into a start scene where it tells you how to play. When you click play it takes you to a setting scene with settings to change the game. There will be a few parameters, such as cards being how many cards will be in the deck which is a slider element from 20 to 40 cards, how many hitman cards are in the deck, slider from 2 to 4 max, start with an angel card, which is a card you can play to survive a hitman card, and make that yes or no or a boolean variable. Once you finish changing the settings and want to start the Game scene, they click the play button to load that scene. Based on the settings it will be different, but the general idea for this scene is a table in a black room with the deck on the left and the played cards on the middle. You sit on one side of the table and your opponent (cpu) sits across from you. Your opponent will have a sprite to be visible.

Types of cards:

- Angel: You start with one of these cards unless your settings say otherwise. When you draw a hitman play this card to not die. 10% to draw a new one from deck

- Hitman: When you draw one of these cards its game over if you cant play an angel card.
- Arson: Randomly burn one of the other players cards
- Bottom Layer: When played, draw a card from the bottom of the deck instead of top
- Mine: Play this to randomly steal one of the players cards
- Shuffle: Play this to shuffle the decks order
- Seeing Eye: Play this to see the top 3 cards of the deck
- Copycat: Play this to get the same effect of the last played card

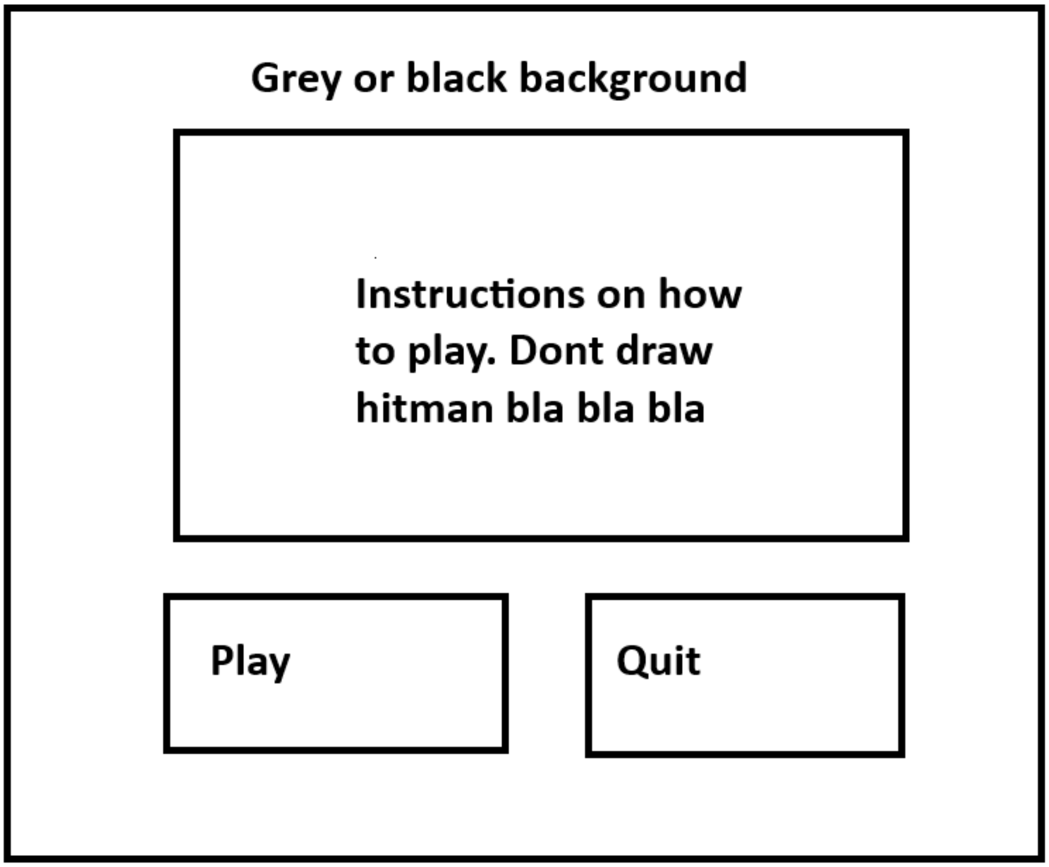
State Transition Diagram



You start at the start scene when loaded. You are then given the instructions of how to play on this scene by using a multilabel. There are two buttons, one for settings scene that will say “Play” to avoid confusion for the user, and one that will say Quit to quit the program. Based on the click save the variable for the while loop inside main and use that to change all the scenes

Instructions

How the game works



First scene shown when loading the game. It has a few key aspects to it

Background- just a basic fill background, preferably grey or black for ambiance. I may switch to a dark alley background depending on how it looks.

Instructions- Multilabel in the middle of the screen explaining how to play the game

playBtn - When clicked loads the Settings scene

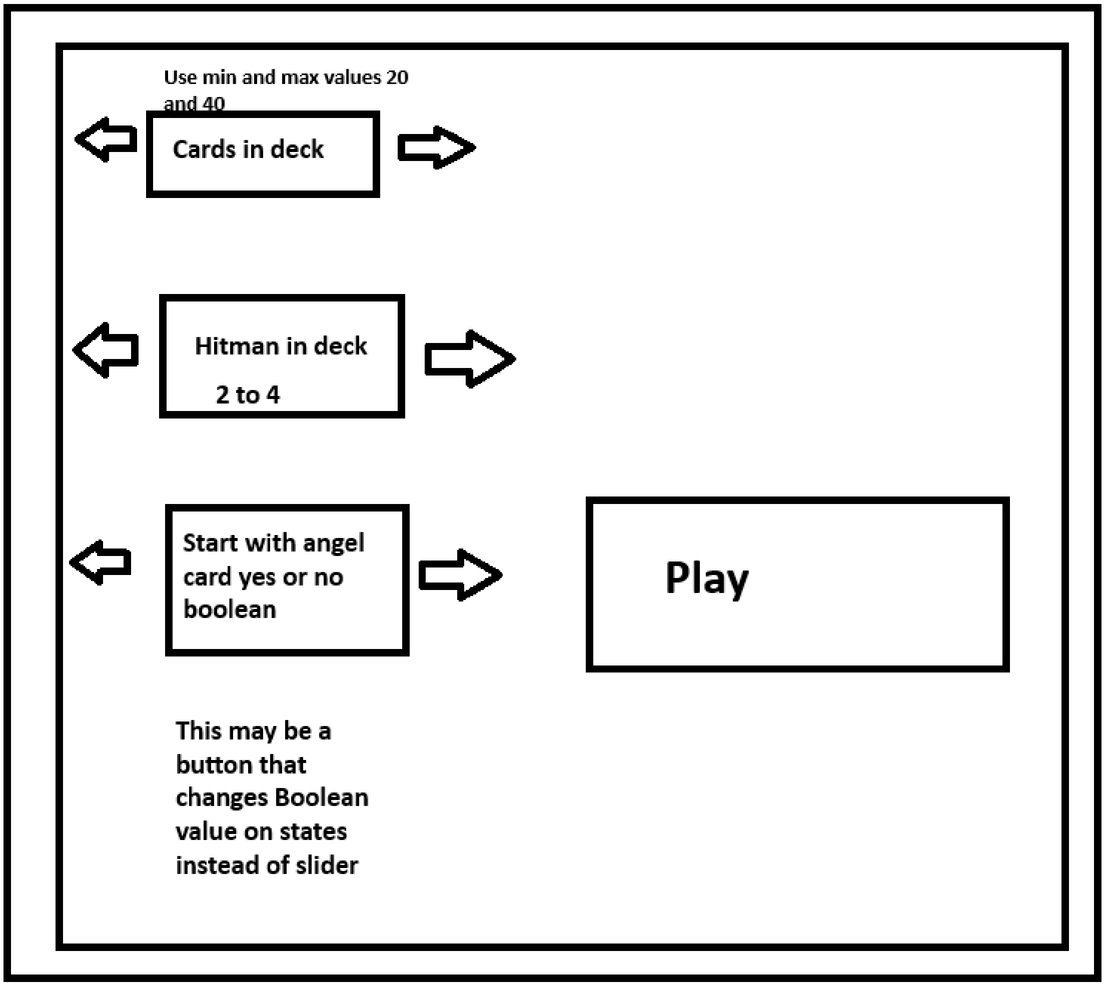
quitBtn - when clicked quit the program by exiting the main loop

```
init():
    Set the background to a grey or black color
    Make a response variable
    Create the instructions text
    Center the instructions text

    Make playButton
    Make the playButton text say “Play”
    Position it to where it is in the image

    Make quitButton
    Make the quitButton text say “Quit”
    Position it where it is in the image

process():
    If the playButton is clicked:
```

```
    Response is play and will run the game scene and quit the current scene
If the quitButton is clicked:
    Response is quit and quit the scene
```

Settings Scene Class

Changes parameters for the game

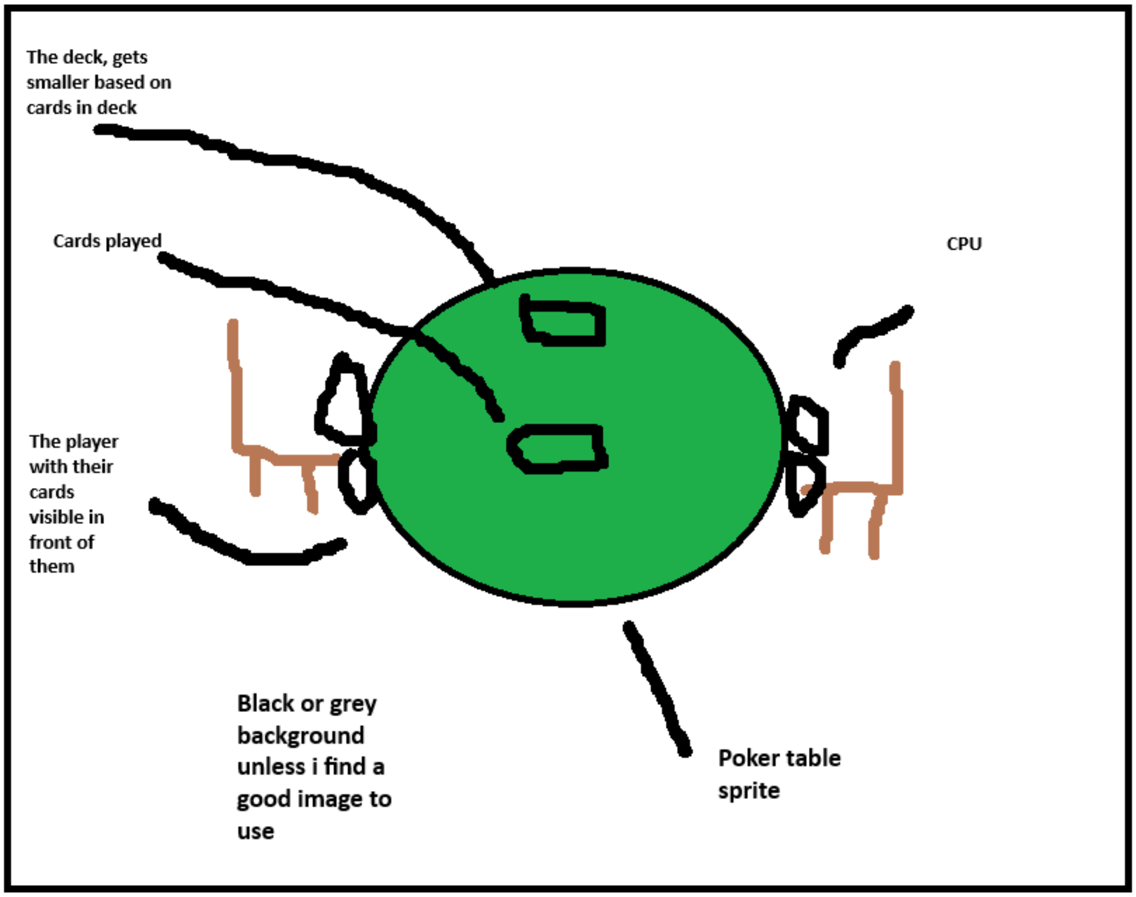

Use min and max values 20 and 40
Cards in deck
Hitman in deck
2 to 4
Start with angel card yes or no boolean
Play
This may be a button that changes Boolean value on states instead of slider

There are 4 main sprites in here, all being sliders except the angel one may change in the future Theres also the play button. These sliders change parameters that alter the game. When the play button is clicked it loads the actual game scene

```
Init():
    Set the background to black color or grey
    Make a variable called “cardsInDeck”
Make a variable called “hitmanInDeck”
Make a variable called “startWithAngel” starts as true
Add the play button

process():
    If play button clicked:
        Send the parameters
        Response is game
```

GAME CLASS

The actual game


The deck, gets smaller based on cards in deck
Cards played
CPU
The player with their cards visible in front of them
Black or grey background unless i find a good image to use
Poker table sprite

The main sprites in this scene are the table, the cards (8 different cards), the other player, maybe your hands on the table, the deck, the other players chair

```
Init(cardsindeck, hitmanindeck, startwithangel):
        Background is either grey or black again
        Make a dictionary that contains all cards including angel and hitman
        Make a list that contains all the cards except angel and hitman
        Position the table and other player in front of the player
        Deck will be a button so we can check if it gets clicked or not
```

```
Make a list for player cards
Make a list for cpu cards

process():
    For i in range hitmanindeck:
        deck.append(hitman)
        Cardsindeck -= 1
    For i in range cardsindeck:
        Rng = rand.randint(0,1)
        For key, value in cardDict:
            If value == Rng
                deck.append(value)
                (ok im not sure if this works as i just made the structure, but basically the dictionary acts as a way to change the chance of something, because with random everything has the same chance but i want angel to not have the same chance)

    If startwithangel:
        playerlist.append([wherever the angel is indexed])
        cpulist.append([wherever the angel is indexed])
    For i in range 5():
        Cardchosen = random.randchoice(cardlist)
        playerlist.append(Cardchosen)
        cpulist.append(Cardchosen)
```

Table Sprite

Poker table image

Size is probably around 125x125 enough to fit in the middle of screen

Set position to middle

Deck sprite

This sprite has an image of a deck of cards, positioned to the left side of the player on the table

It will be a button to check for click

```
process():
    If clicked:
        Chosencard = random.choice(deck)
        deck.remove(Chosencard)
        playercards.append(Chosencard)
```

Cpu Sprite

This sprite is an image of some type of player, TBD

(i will also add a turn variable that is a boolean to determine if it's their turn to go or not, changes based on if they played a card or not)

```
process():
    If turn:
            Chosencard = random.choice(cpucards)
            cpucards.delete(Chosencard)
```

TextLabelCardsLeft

Textlabel above the deck that tells you how many cards are left in the deck

TextlabelCardsCPU

Textlabel by the cpu cards to tell you how many cards they have

```
main():
    The main loop of the program
    Use keepgoing boolean and set to true

    While keepgoing:
        Load instruction scene
        When instruction scene ends check for a response
        If response is “Settings”:
            load settings scene
            If response is “Game”:
                and load Game scene
        Else:
```

Keepgoing = False

assets

hitmancard

Poker table

Hitman (CPU)