Deliverable III

Task status

Task	Owner	Status	Comments
Main menu GUI (New game, Resume, Stats)	Joe	Completed	Could add Back button during setting
Board GUI	Chia-Chun	Completed	Could change deck image
Instructions	Gavin	On track	Right side message
Deck	Peter	On track	Show card info
Player AI	Gavin	Behind	Game has not been developed enough
Game Pieces and optional movement locations	Joe	On track	Draw cards and find possible moves
Stats	Peter	Behind	Not implemented yet
Saving game and resuming	Peter	On track	Debug/module
Pawn movement and Card draw animations	Chia-Chun	Behind	Fail to go home
Test	All	Behind	

Have done:

- Main menu GUI buttons
    - New game (on track)
    - Resume game (on track)
    - Instruction (completed)
    - Statistics (behind)
    - Quit
- New game
    - Pick a color
    - Decide how many computers
    - Decide how smart and how nice
    - Show the game

- Playing
    - Click a pawn -> draw a card (will change this)
    - Pawn can move/slide/bump
- Database
    - reading and writing (writing once game is over, so that is not finished)
    - Read to give statistics

Tasks to do

- Playing
    - AI setting
    - Computers draw and choose pawns automatically
        - Calculate score
    - Draw a card -> choose a pawn
        - How many pawns can be choose
        - How to implement card 7
        - Computers choose a pawn
        - Discard and draw a new card
    - Deck design (Draw button/Discard area/Show card info)
    - Add objects
        - Save button
            - Save the game - behind
        - I quit button
            - Abandon the game
        - Info button
    - Show info on right side
- Resume game
    - Load game information and draw the game - behind
- Instructions
    - Different buttons
- Statistics
- Test

Main menu

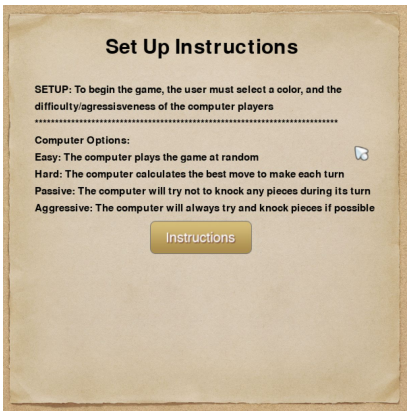
When the main program is executed, this main menu will be shown in a window. User can click either of five buttons. However, only “new game”, “instructions”, and “quit” work properly now. The feature of saving or resuming game has not been completed yet, and statistics will be implemented later.

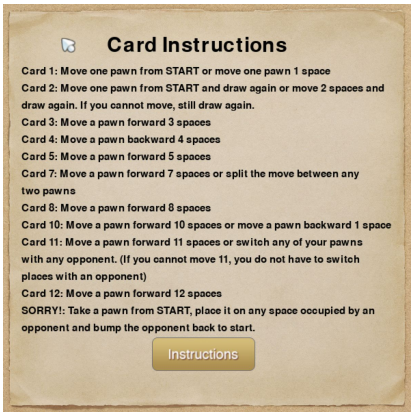
```
ip0af54525:cs205sorry chun$ python3 main.py
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    from menu import Menu
  File "/Users/chun/Documents/GitHub/cs205sorry/menu.py", line 4, in <module>
    from database import Database
  File "/Users/chun/Documents/GitHub/cs205sorry/database.py", line 2, in <module
>
    import MySQLdb
ModuleNotFoundError: No module named 'MySQLdb'
```

Also, we experienced problems with installing MySQL module on Macbook, so only the member using Windows system can save or resume a game now.

Instructions

In the instructions page, user can pick the instruction of a topic.

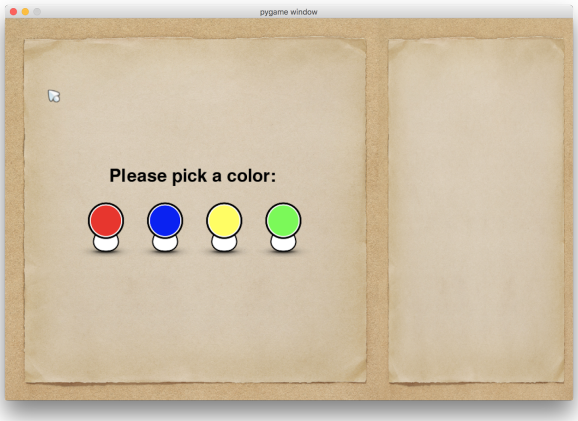




The description is shown in the square part of the window. There is also a button to go back to the instructions menu.

Game setting

After user clicks the new game button, user can decide a color, how many computer players, and how nice and how smart they are.

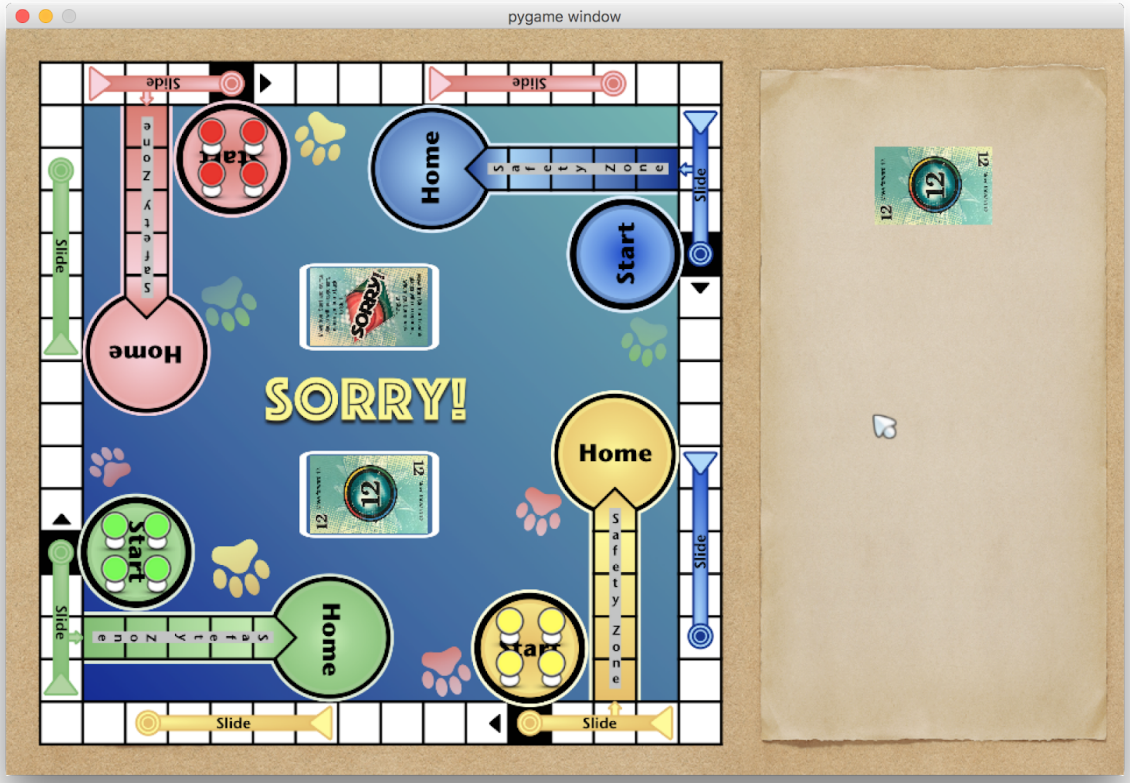


The board is rotatable in the beginning, so once the user chooses a color, the color will be placed at the bottom of the board.

If user selects 1 computer opponent, there will be only two colors of pawns drawn on the board. If user selects 2, there will be three colors of pawns. The order is from bottom, left, top, to right

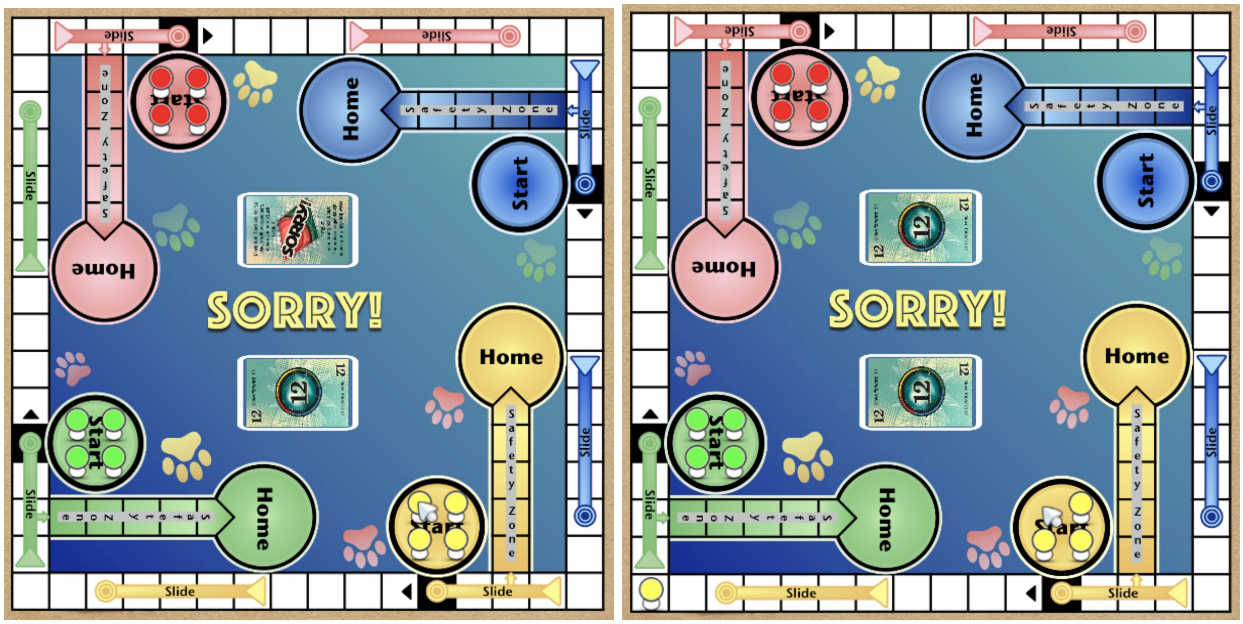
When deciding how hard the game should be, a checkmark will be drawn on the setting after user clicks it. Only when all computer settings are selected, user can click Done button to continue.

Playing

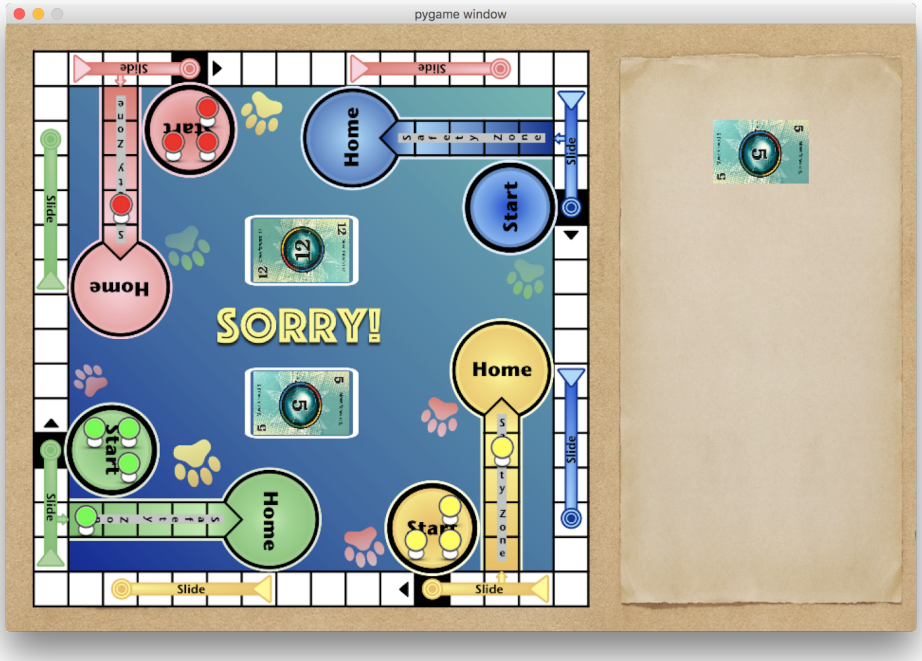


Before this game starts, we selected yellow, 2, and both Nice&Smart. As shown in the image, yellow is at the bottom. There are two computer opponents located at the left and the top. The colors of computers is decided according to the position. The positions always follow the order of bottom, left, top, and right.

Once the game setting is done, this board is shown in the window. The upper rectangle in the middle is the card deck, and the lower one is the discard deck. After a card is drawn, it will be placed in the discard deck and players can see the number.



Currently, the game starts when user clicks a pawn and the program draws a card directly. The pawn will move steps of whatever number it is on the card. This will be changed afterwards, because we did this only for easy implementation. Therefore, when the yellow pawn is clicked, and the program draws a card 12, the pawn successfully moves 12 steps from start to the corner.



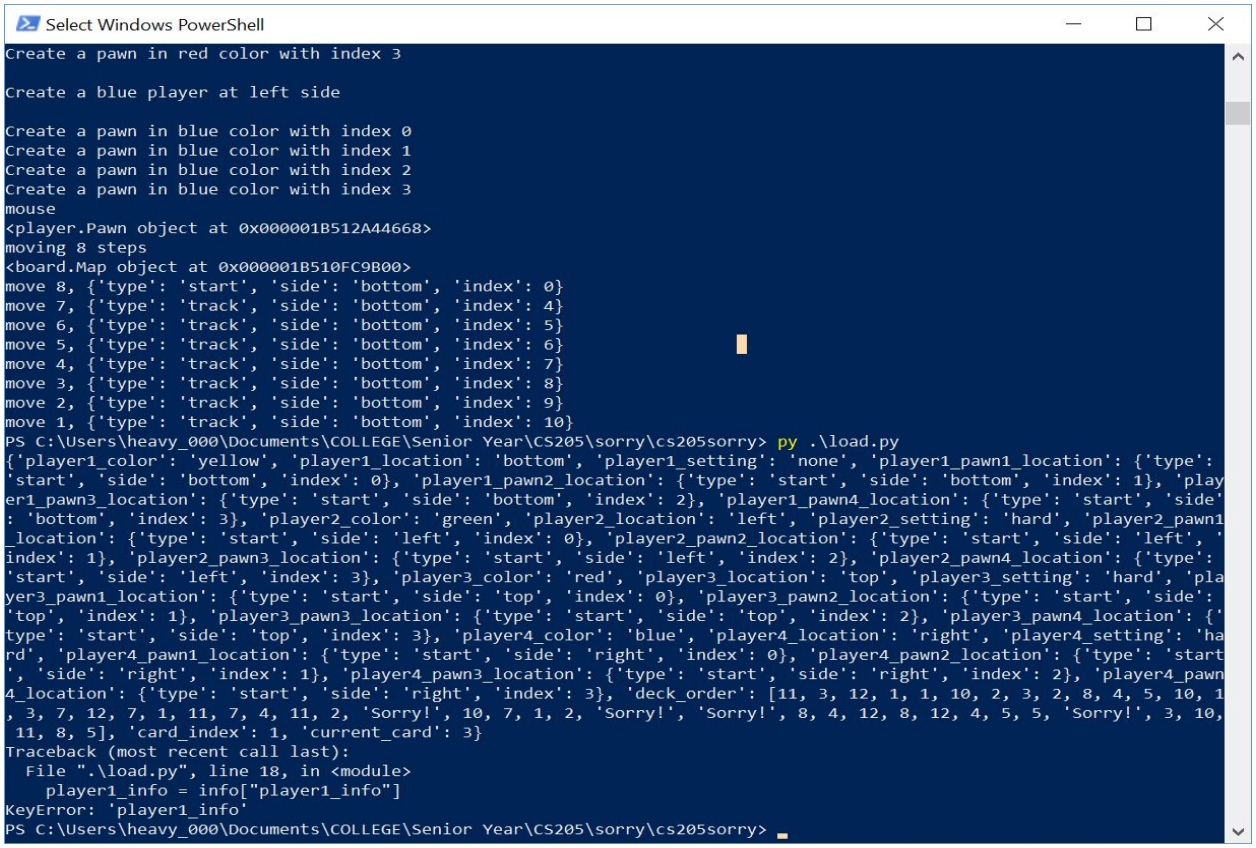
However, after testing the program for a while and letting any pawn stand in safety zone, we found that even the green pawn can move for 5 steps, it did not move.

```
def onClick(self):
    """
    When a pawn is clicked, this function will be called and move the pawn if possible
    """
    if (self.playerPosition is self.main.game.turn):
        #Draw a card and decide how many steps to move
        self.moveStep = self.main.game.drawCard()

        #Check if this pawn can move to the destination
        #status = moving/sliding/safe/home/notAllowed
        self.status = 'moving'
        destination = self.position
        for i in range(abs(self.moveStep)):
            destination = self.getNext(destination)
        self.status = self.checkCollision(destination, self.status)
        if self.status is 'notAllowed': #If this pawn cannot move, ignore
            self.moveStep = 0
        self.status = 'moving'

        #Change to next turn
        self.main.game.nextTurn()
```

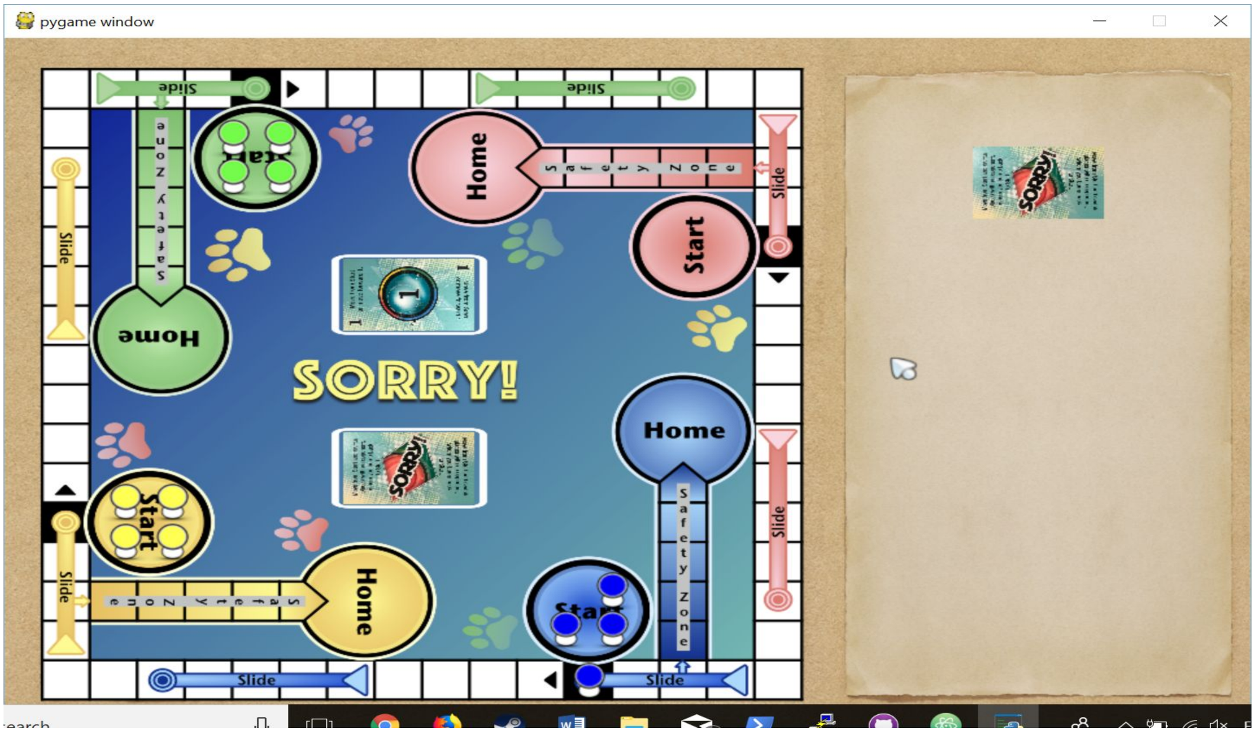
There should be something wrong before self.checkCollision() is called, because there's where showed error message.

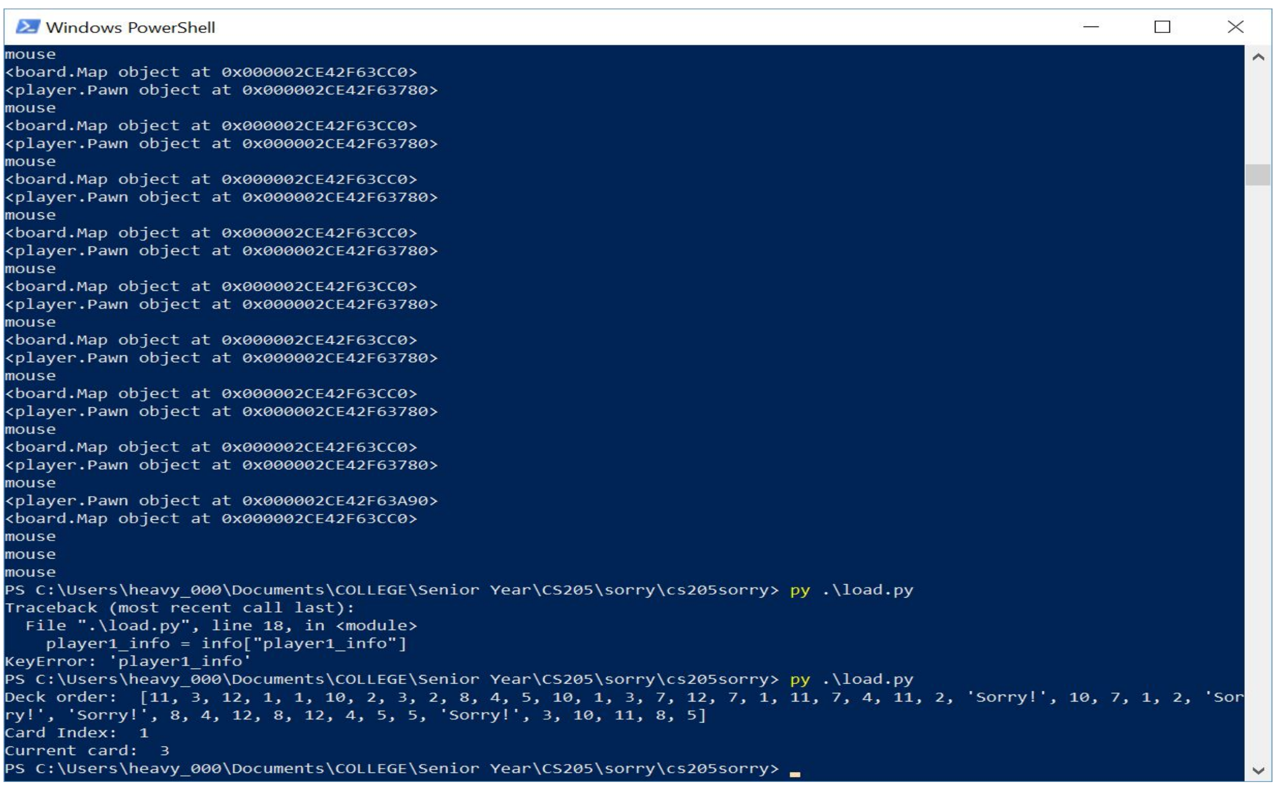
Saving/Resuming:

Working with saving and resuming has been a somewhat frustrating task overall. Working with files that others wrote leads to a longer time to understand what is going on with everything and searching through many lines of code for the correct variables and functions. A lot of errors I have had were key errors from using a dictionary coupled with the pickle module in python that allows a large dictionary to be saved into a text file then loaded back in as a dictionary. It is a very helpful tool to do saving and loading and smooths thing out a lot instead of doing it another way. As you can see in the screenshot below, there is a key error on the player1_info. At first glance it appears that there shouldn't be a key error problem, but I am assuming there is no value there as something is wrong with when and where I took the value from. I currently have quite a few key errors after this one so I can assume it is not just an error with 1 variable, and that it is a problem with my saving function.



There is also a problem with saving the deck, which was working before, but after I changed around some things with the save function to make it more usable with other code, I have not been able to save. For example you can see in the two below screenshots that the Sorry! Board game first drew 12, then 1, then Sorry!, but in the function that should be loading the deck, it prints the loaded deck but as we can see, it starts off with 11, 3, 12, which is not in the correct order. So something overall is wrong with saving, maybe just not getting to the part to save the deck or just not saving at all. After commenting out the parts without saving the deck, it still does not work, so it is something wrong with the saving function overall.

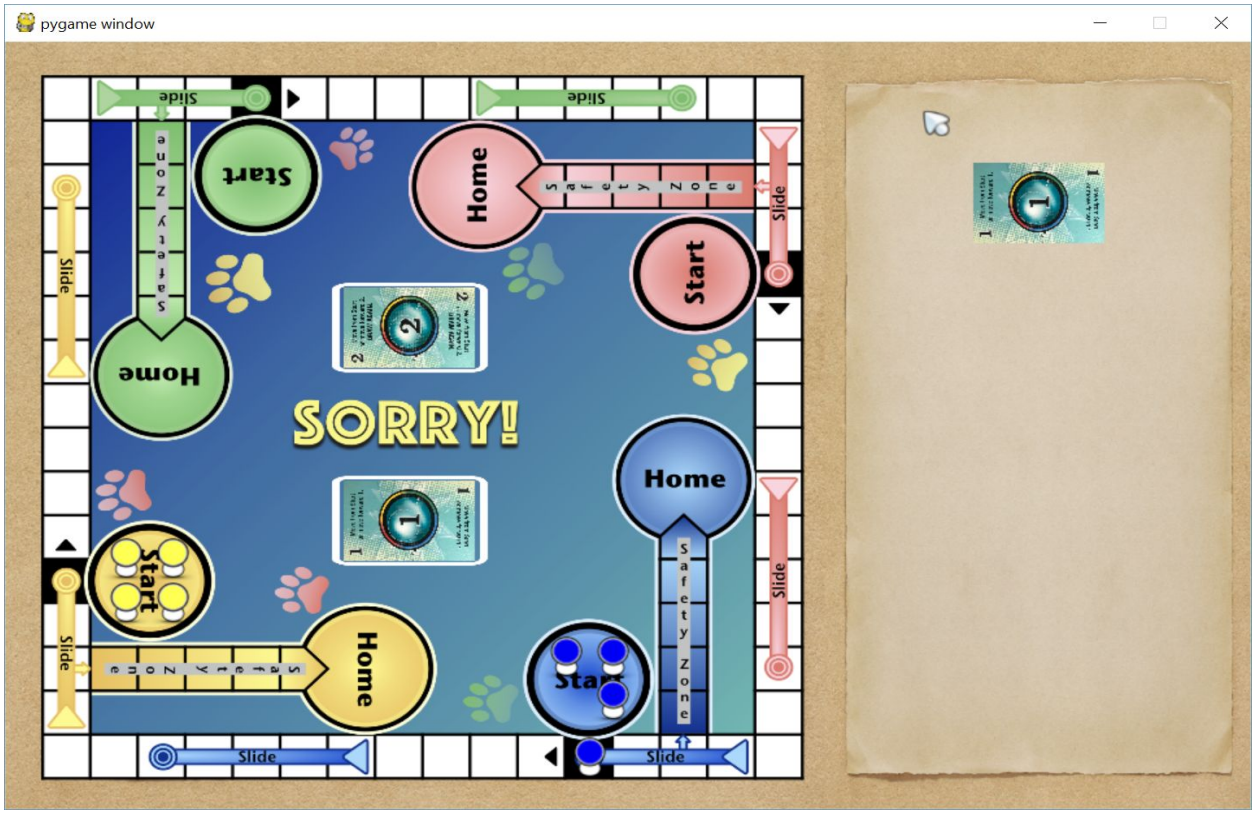


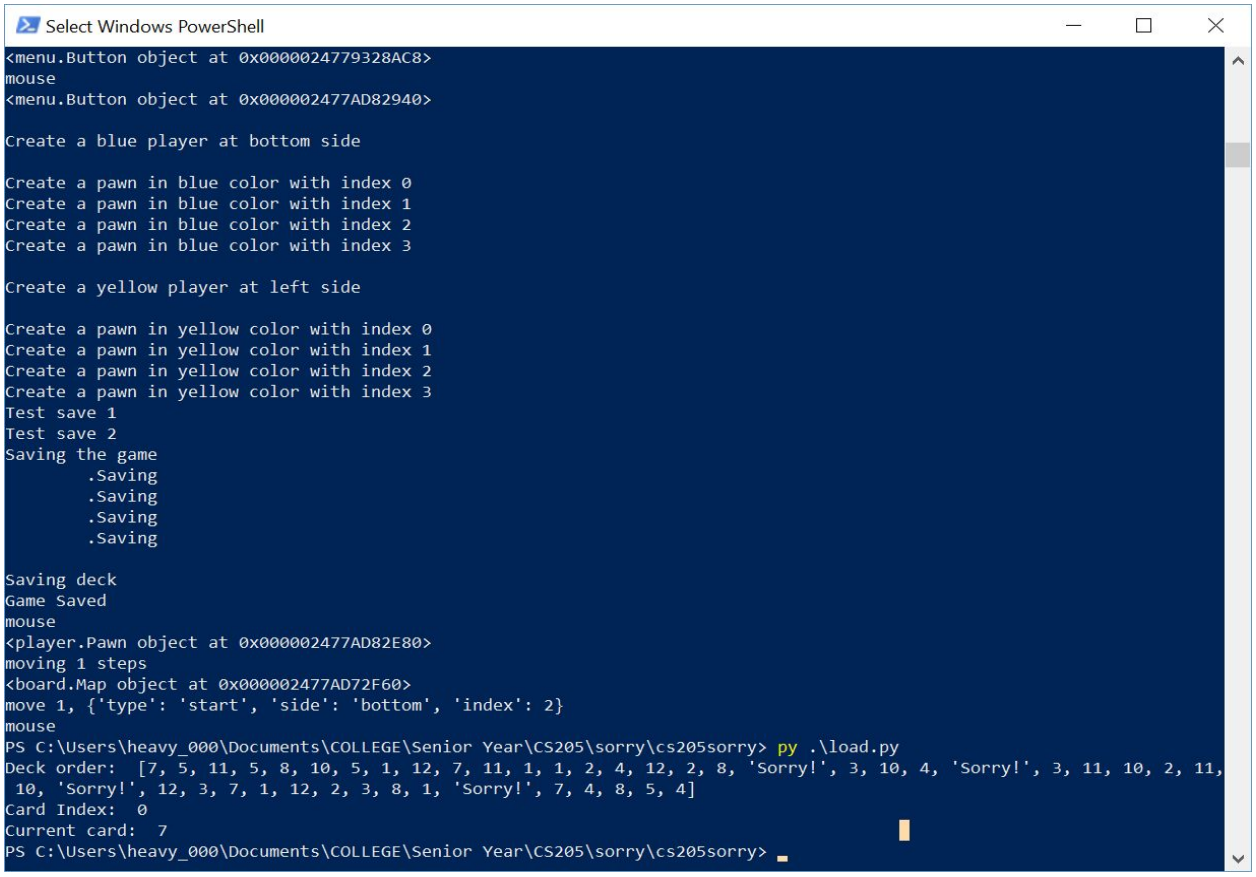



```
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<board.Map object at 0x000002CE42F63CC0>
<player.Pawn object at 0x000002CE42F63780>
mouse
<player.Pawn object at 0x000002CE42F63A90>
<board.Map object at 0x000002CE42F63CC0>
mouse
mouse
mouse
PS C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry> py .\load.py
Traceback (most recent call last):
  File ".\load.py", line 18, in <module>
    player1_info = info["player1_info"]
KeyError: 'player1_info'
PS C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry> py .\load.py
Deck order:  [11, 3, 12, 1, 1, 10, 2, 3, 2, 8, 4, 5, 10, 1, 3, 7, 12, 7, 1, 11, 7, 4, 11, 2, 'Sorry!', 10, 7, 1, 2, 'Sor
ry!', 'Sorry!', 8, 4, 12, 8, 12, 4, 5, 5, 'Sorry!', 3, 10, 11, 8, 5]
Card Index:  1
Current card:  3
PS C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry> _
```

However, I found that part of the save function was commented out due to errors it caused (probably because I left it with a game-killing error after pushing to Git), so I fixed a few of those errors and uncommented it, which now works again and saves.

However, the function to save the deck order from the board is not the same as in the save function, I.E. the deck that shows up in the game is not the same as the deck that gets loaded back in. This is shown in the screenshots below.

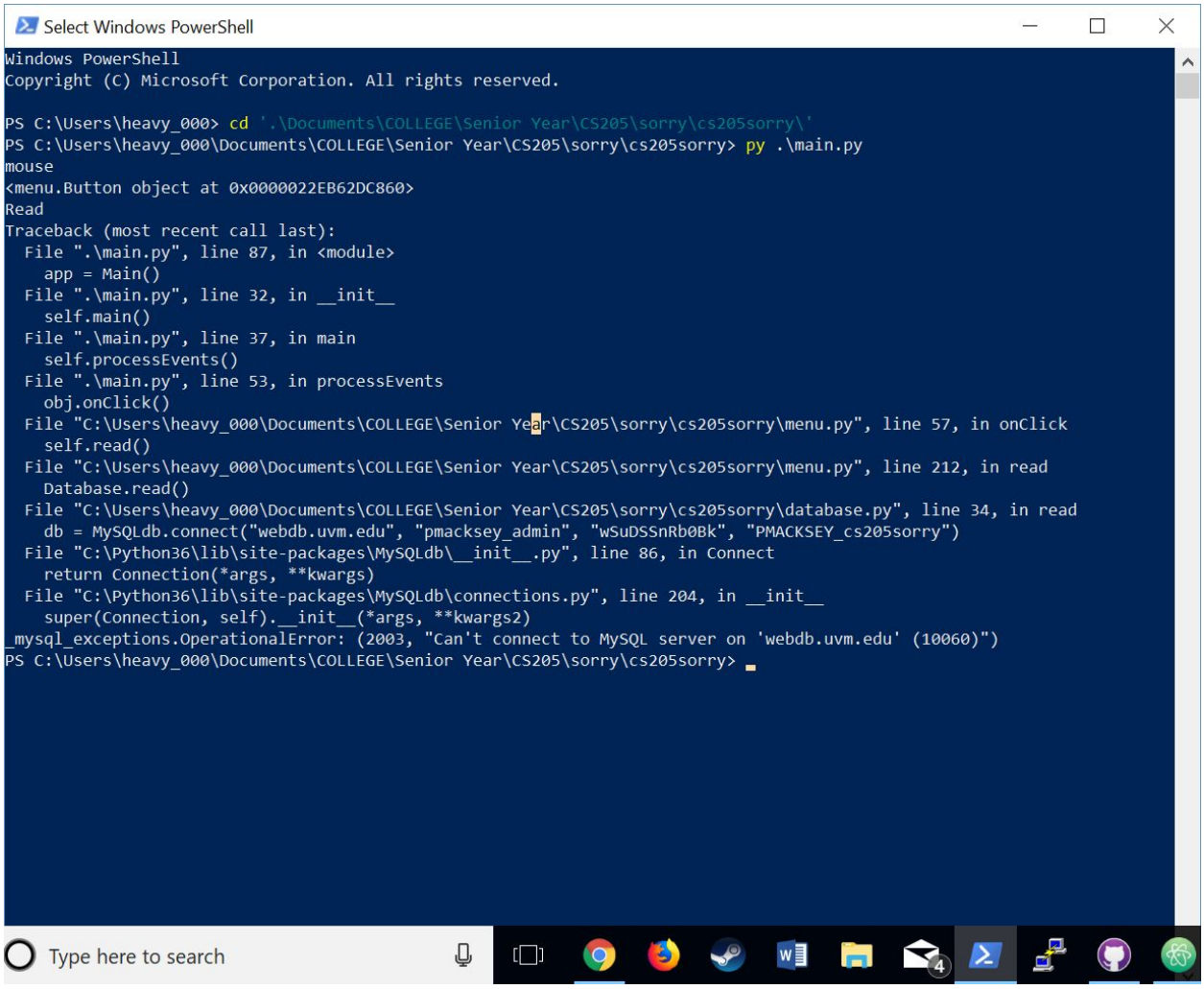




The loading function seems to be working fine, as once you get the variables into the save function and save to the save.txt file, they will all load as long as I put the variables in to load them, so the only problems there would be the issues of me not implementing them.

Database

Working with the database has presented some issues. First of which is getting the python package to work on everyone's computer. In addition, you must be connected to the UVM wifi to read the database at the moment, and I will try to find a way to be able to use it without a VPN, as I would like any game played to be able to update to the database. Below is what happens when you are not connected to the UVM network.


Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\heavy_000> cd '.\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry\'
PS C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry> py .\main.py
mouse
<menu.Button object at 0x0000022EB62DC860>
Read
Traceback (most recent call last):
File ".\main.py", line 87, in <module>
app = Main()
File ".\main.py", line 32, in __init__
self.main()
File ".\main.py", line 37, in main
self.processEvents()
File ".\main.py", line 53, in processEvents
obj.onClick()
File "C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry\menu.py", line 57, in onClick
self.read()
File "C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry\menu.py", line 212, in read
Database.read()
File "C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry\database.py", line 34, in read
db = MySQLdb.connect("webdb.uvm.edu", "pmacksey_admin", "wSuDSSnRb0Bk", "PMACKSEY_cs205sorry")
File "C:\Python36\lib\site-packages\MySQLdb__init__.py", line 86, in Connect
return Connection(*args, **kwargs)
File "C:\Python36\lib\site-packages\MySQLdb\connections.py", line 204, in __init__
super(Connection, self).__init__(*args, **kwargs2)
_mysql_exceptions.OperationalError: (2003, "Can't connect to MySQL server on 'webdb.uvm.edu' (10060)")
PS C:\Users\heavy_000\Documents\COLLEGE\Senior Year\CS205\sorry\cs205sorry>
Type here to search