Justin Nordeman

Prof. Annexstein

CS 2021: Python Programming

8 Dec 2019

Final Project Report

**Introduction**

In this project, I developed a game using the pygame library. To begin designing my application, I begun by familiarizing myself with the pygame library by getting picture and text to show up on the screen as well as creating and moving sprites. After I familiarized myself with the library, I begun the development of my game by first planning out the type of game I wanted to create as well as the general flow of the game. I decided on making a football simulator loosely based on the infamous 2009 Cincinnati vs. Pittsburgh game.

To begin creating my project, I first made a screen that would display a game menu upon launch. On the menu, I displayed the game title, a picture, and a button that would launch the user into the game upon clicking it. From here, I knew that my game would be based upon the user clicking buttons, so I developed a button function that allowed me to set the button color, size, location and a message to display. In the button function, I included an argument that would decide what action would happen if the user clicked the button. Next, I developed the instruction screen with a button that would trigger the play calling screen when clicked. After this, I developed seven outcomes each for running and passing the ball which would then update the yardage, down ball location, and time remaining which is displayed at the top of the play call screen after each play is ran. To finish out the program, I developed a screen that would display if the user lost the game as well as putting pictures on the play calling screen and each play outcome screen. Finally, I developed a screen that would display if the user won the game. This screen also played audio from the actual game in 2009 that my game is based off. The general logic of my program can be seen in figure 1 below.

Please note, in order to successfully run the game, you will need to download the necessary audio and image files included in the [GitHub](https://github.com/nordemja/Python-Final-Project) repository.

**A close up of a map

Description automatically generated**Fig. 1. Flow Chart of Game Logic.

**Project Results**

In this project, I accomplished the development of an event-driven game. I successfully implemented the use of buttons and pygame in the development of the project. I also successfully managed the possibility of different outcomes in the game by implementing the use of multiple different screens in which certain events in the game would determine which screen to display. Additionally, I was also able to successfully implement the use of sounds

While developing the game, I ran into several pitfalls. First, I had an issue immediately after designing the game menu while trying to launch into the play calling screen. I unable to launch into the play calling screen because I was trying to perform the necessary action, corresponding to the event that the button would trigger, outside of a while loop. Putting the actions inside of a while loop was needed to get the screen to stay on the next event until the user clicked on the button to move to the next event. My next issue arose when I had an issue of buttons on separate screens being in the exact same location of the previous screen’s button. This caused the issue of running both button’s commands at the exact same time. I first noticed this issue while designing my game lost screen. I debugged this issue by moving the location of the buttons around to see why both button commands were running at the exact same time. Another issue that I encountered, was when I was implementing the use of audio in the project. Unlike images, I could not load both audio files at the same time at the beginning of the project. This caused the issue of only playing the most recently audio file. To solve this issue, I simply loaded each audio file when it was needed in their respective functions and included a stop command when I wanted the audio to quit playing. These issues and sample correct outputs can be seen in figures 2 – 7 below.

I would make several future improvements to this project. My main improvement would be implementing an actual countdown timer that would tick while the game is running and update in real time. My next improvement would be to add a sprite or a graphic that would display where the ball is on the field by following a simulated, marked up, football field. I would also possibly add functionality to let the user control the team and take control of a player and play out the down, rather than just return a random outcome.

Fig. 2. Error in code since the sound is playing in the while loop.

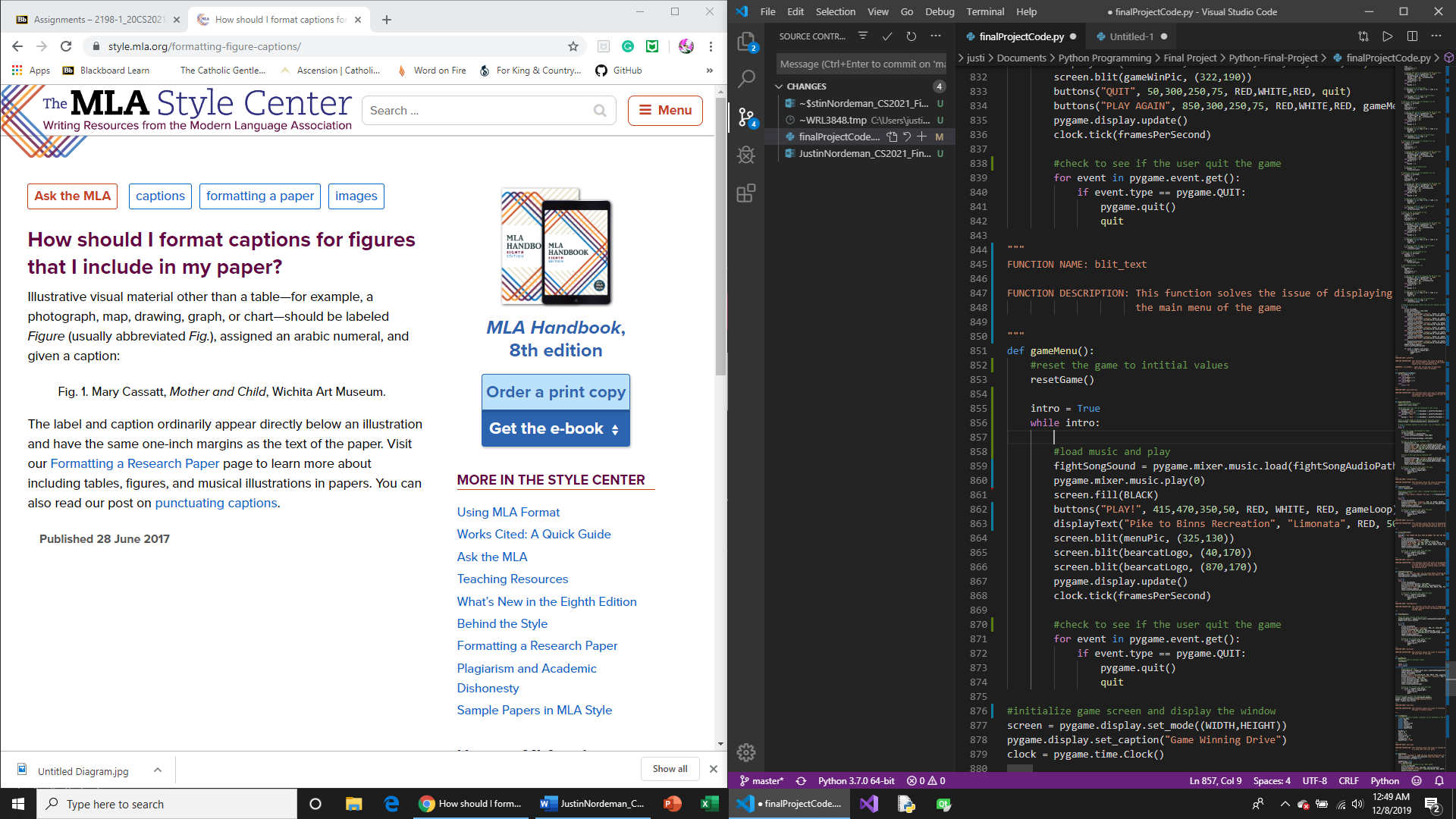


Fig. 3. Error in code since the screen is not constantly filling with black in the while loop.

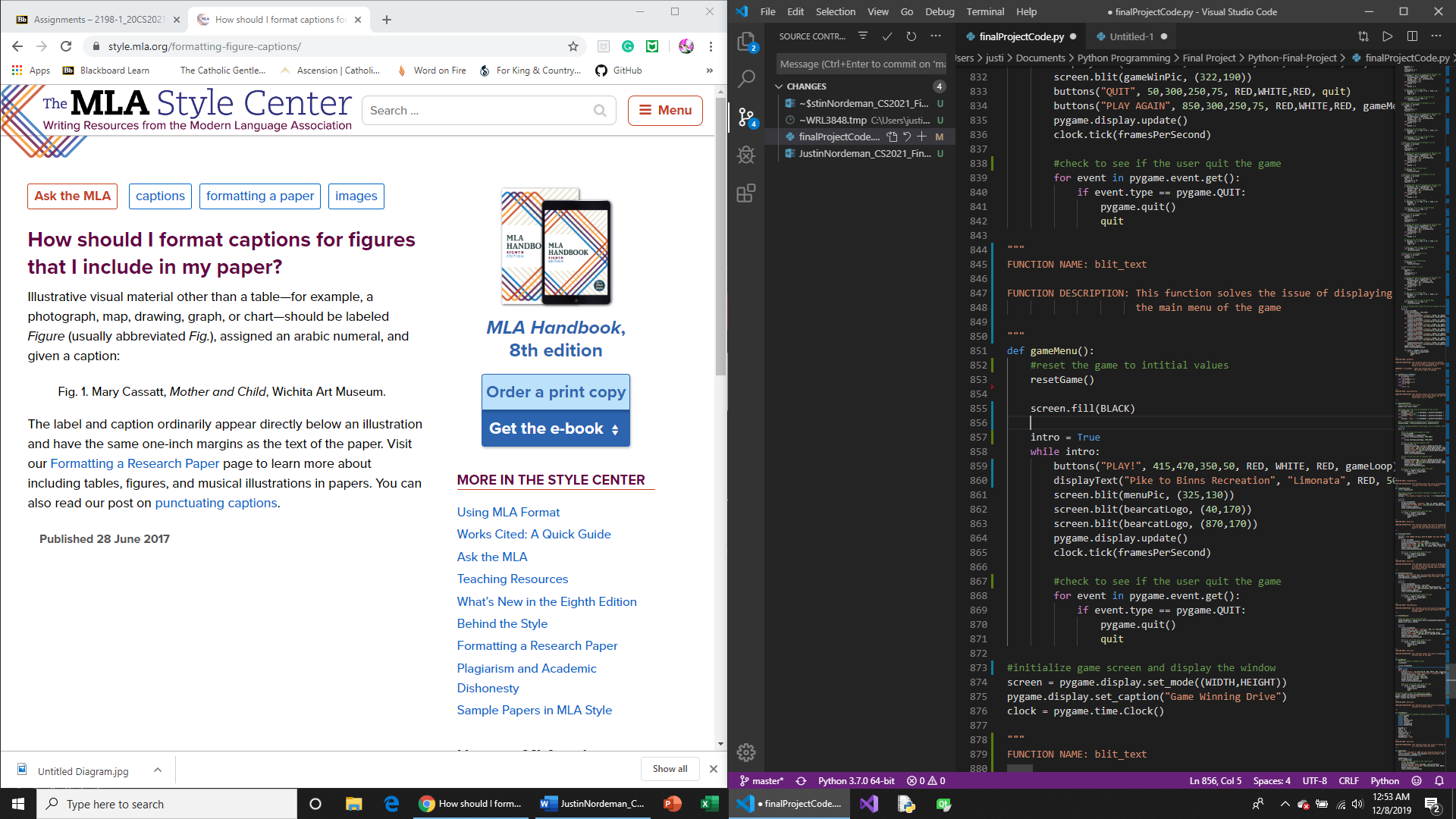


Fig. 4. Correct code that shows sound playing outside loop and screen being filled black.

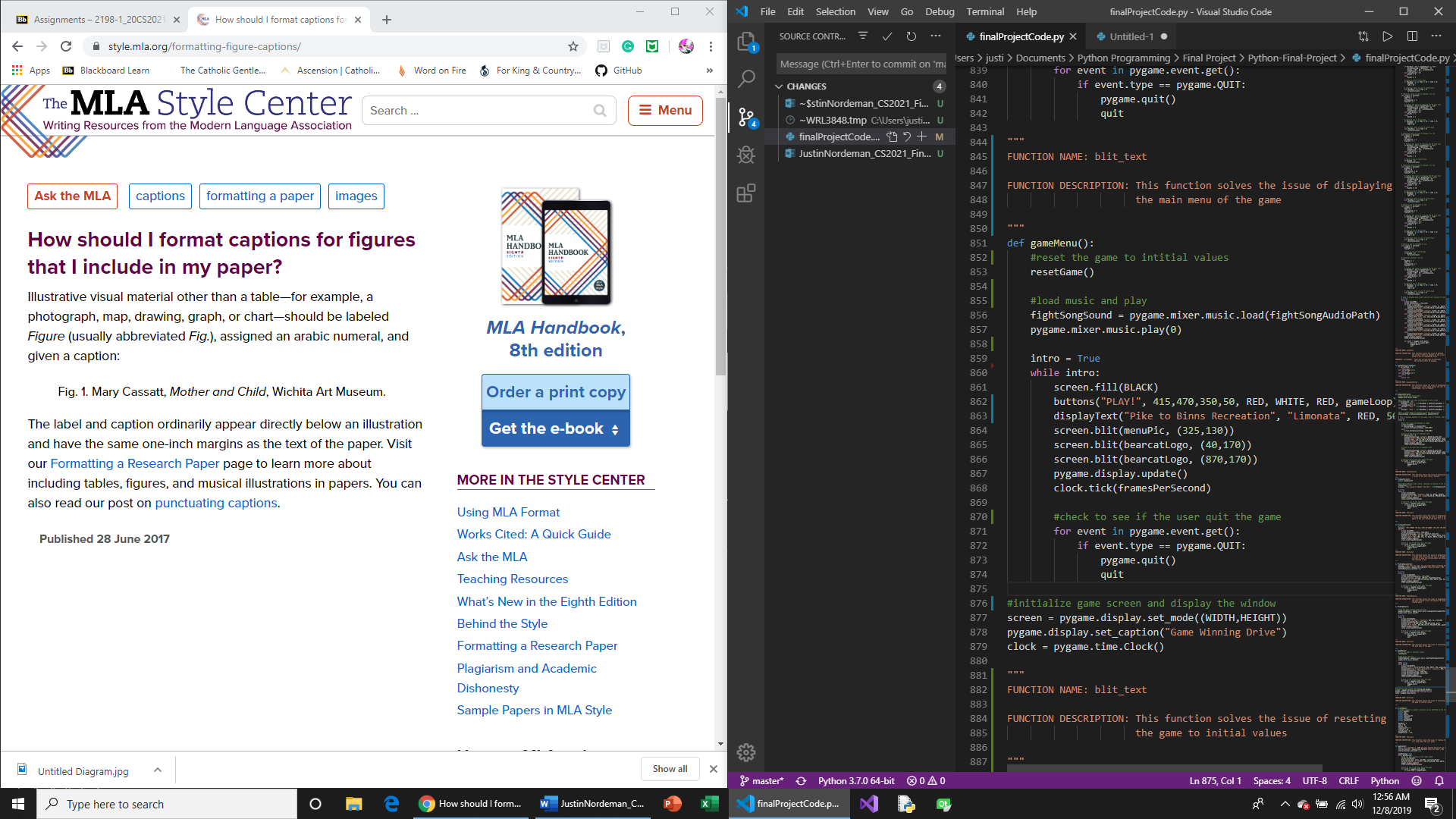


Fig. 5. Correct sample output of the user running the ball.

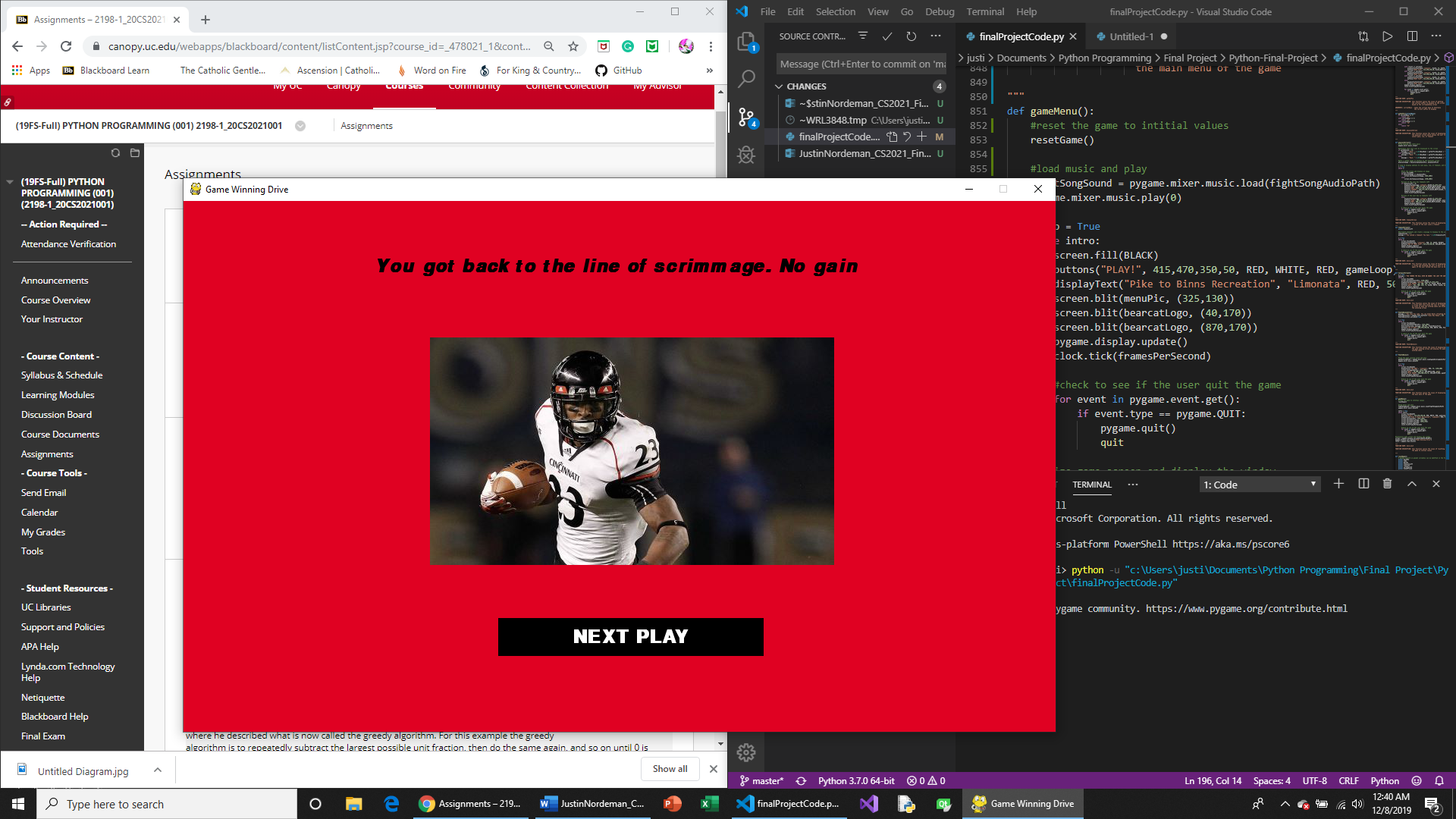


Fig. 6. Screen that is displayed if the user lost the game.

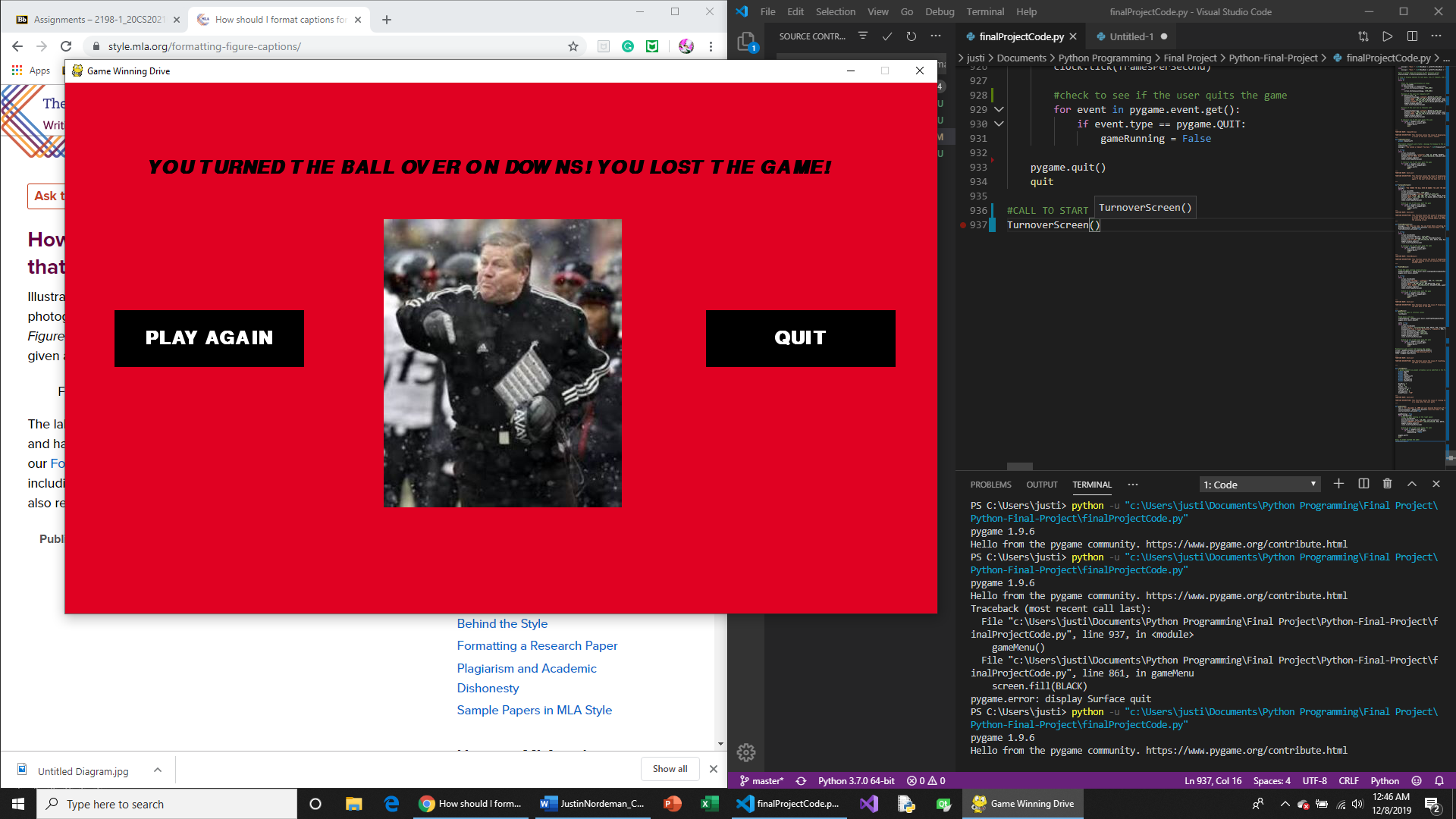
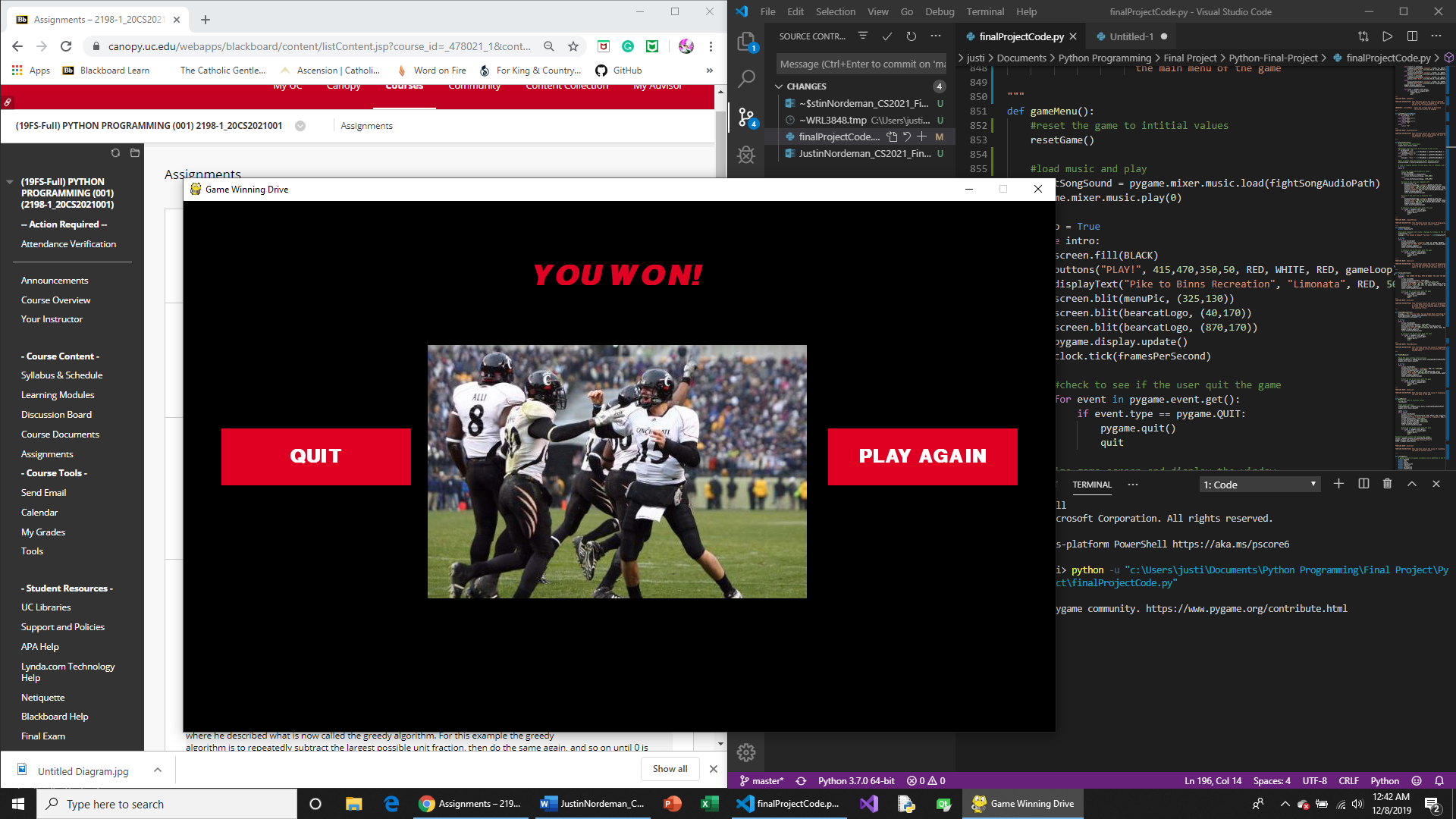


Fig. 7. Screen that is displayed if the user won the game.



**Division of Work**

During the development of this project, I chose to work alone and developed most of the program on my own. I did use two functions that were found online. The first function I used was from Stack Overflow contributed by Ted Klein Bergman. This function was useful in my program when I needed to allow for multi-line text wrapping. The second function that I used was from Harrison Kinsley on pythonprogramming.net. This function was used for creating my buttons used in the game. I did slightly modify this function by adding an argument that would also allow the text on the button to change if the mouse was hovering over it. Additionally, the images used in the game were saved from google images and are not my own.

The rest of the program was developed by me. The first two functions developed was a function to display the game menu and a function to display the instruction screen. This includes the displayText function which I developed to allow the program to display text on the screen without having to retype all the necessary text commands. I also developed a simple function called Results which returns a random element from a list that is passed in as an argument. The two main functions in the program were developed to handle the event of running or passing the ball. I also developed a function that would display the play calling screen to allow the user to choose to pass, run, or call a timeout. The three game ending functions were developed by me to run based on if the user won or lost the game. Additionally, I also developed a function that would display a screen if the user called a timeout. Most of this program was developed independently, with two functions used from online which is noted in the program code.

**Code Appendix**

The code and necessary image and audio files can be accessed by following the provided GitHub link below.

<https://github.com/nordemja/Python-Final-Project>

Works Cited

“Armon Binns Game Winning Catch - Dec. 5th, 2009.” *Youtube*, CincinntiBearcats, 13 Dec. 2013, www.youtube.com/watch?v=F0pyYwuyG90.

Bergman, Ted Klein. “Rendering Text with Multiple Lines in Pygame.” *Stack Overflow*, 2 Feb. 2017, stackoverflow.com/questions/42014195/rendering-text-with-multiple-lines-in-pygame.

“Brian Kelly Snow.” *Seattle Times*, The Associated Press, 10 Dec. 10AD, encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTVGCiiAv6a9jSFWQh4gKuaqNBv-36igVrzhfdQuMUe5PjKiUq16A&s.

Conpboyle. “Pike to Binns.” *247sports*, 24 Apr. 2017, gobearcats.com/images/2017/8/29/isaiah\_pead\_111616\_getty\_ftr\_jxzilqiauwbt1wjswozryrph1.jpg?width=1023&quality=80&format=jpg .

Egger, Mo. “Tony Pike.” *ESPN 1530 On Demand*, 8 May 2019, gobearcats.com/images/2017/8/29/isaiah\_pead\_111616\_getty\_ftr\_jxzilqiauwbt1wjswozryrph1.jpg?width=1023&quality=80&format=jpg .

“Isaiah Pead.” *Go Bearcats*, University of Cincinnati Athletics, 23 May 2018, gobearcats.com/images/2017/8/29/isaiah\_pead\_111616\_getty\_ftr\_jxzilqiauwbt1wjswozryrph1.jpg?width=1023&quality=80&format=jpg .

Kinsley, Harrison, director. *Game Development in Python 3 With PyGame - 15 - Button Events*. *Youtube*, 10 Oct. 2014, www.youtube.com/watch?list=PLQVvvaa0QuDdLkP8MrOXLe\_rKuf6r80KO&v=P-UuVITG7Vg&feature=emb\_title.

Smith, Thomas E. “Brian Kelly Expresses His Unhappiness with the Performance of His Offense. At Left Is Quarterback Tony Pike and Wide Receiver Mardy Gilyard.” *Cincinnati.com*, 8 May 2019, Brian Kelly expresses his unhappiness with the performance of his offense. At left is Quarterback Tony Pike and wide receiver Mardy Gilyard , https://www.gannett-cdn.com/presto/2019/05/06/PCIN/c43f2f35-82e5-412d-a937-b97d340991c1-1205\_UC\_PITT-55.jpg?width=2560

Sullivan, Chuck. “Armon Binns Catching Football.” *The American*, 9 Sept. 2019, theamerican.org/images/2019/9/4/CINCINNATI\_BINNS\_ARMON\_2032.jpg?width=1416&height=797&mode=crop.

“University of Cincinnati Bearcats Fight Song ‘Cheer Cincinnati.’” *University of Cincinnati Bearcats Fight Song "Cheer Cincinnati"*, redlegsfan21, 12 Oct. 2019, https://www.youtube.com/watch?v=m0gPFqFYA0E.

Williams, Justin. “Tony Pike Passing.” *The Athletic*, 1 Aug. 2019, cdn.theathletic.com/app/uploads/2019/07/23232133/GettyImages-94023308.jpg.