Final Proposal

Project Description and Background

The goal of this project is to design an application that includes two parts on the smart phones or iPads. The first part is to predict the number of coronavirus infections in top 10 states that have the most coronavirus infected cases using the method of regression we learned in class so that people who download this app will have a better understanding of how epidemic situation will develop over time (maybe in the next few days or next few weeks) quantitatively. The second part, which is the main part of this project, is to design a board game that aims to fight against coronavirus, and the basic form of the game is similar to P6 assignment teeko_player. In particular, the goal of this game is to teach children and teenagers how the virus spreads and methods to prevent its spread. They can learn more about coronavirus at home while playing the game during the time of quarantine. However, since I have a few months during the summer to focus on this project, I will first do a toy version of the project to make sure I can test and fix every aspect of this game so that the users can play the game smoothly without encountering bugs. After doing this, the next step is to create a Graphical User Interface (GUI) and incorporate the basic game algorithm into the application. But for this proposal, I only focus on the toy example and will generalize later. I will explain more on these steps in the Project Details part.

Survey and Data Collection

At the very beginning, I created a survey that asks to choose top 3 most important things we need to have to protect ourselves during COVID-19. I gave this survey to more than 50 people, including my parents, my family members, my friends in different parts of the world. Based on the statistics, medical masks, alcohol (spray or pads), and medical gloves are the most popular choices. Furthermore, I searched on the Internet and found data from Centers for Disease Control and Prevention (CDC) on COVID-19 reported cases in each state in the United States. I will explain the purpose of doing the survey and finding the data later. Also, I interviewed about ten children about the format of the game they like.

Project Details and Descriptions about Preliminary works

The **stakeholders** in this project are children and teenagers because they will know more facts about COVID-19, understand the importance of some precautions, and learn how to protect themselves during this period after experiencing this app. Since medical mask is the most needed thing during the pandemic based on the survey, I will use ‘M’ and ‘V’ to represent medical mask and coronavirus in the gameplay instead of using ‘b’ and ‘r’ like in P6. Also, for the toy example or the proposal, I will just focus on New York which has the most reported cases.

Then, I will talk about **main technical steps**. For the first part which is the regression and prediction, I first downloaded the csv data about the reported cases in New York from March 3rd, 2020 till May 2nd, 2020 and revised them. The specific data format will be introduced in the Preliminary Results part. Then, I wrote a get_data() function to get that csv data. Next, I had a compute_betas() function to calculate and return the values of beta_0 beta_1 (intercept and slope of the regression) using the closed-form solution. Finally, I used a predict(diff) function, where the parameter equals the difference between the day you want to predict and March 3rd, 2020. For the game part, I first have a MaskVirusPlayer class. Inside this, I define my board and pieces. Since it is just the toy example, I will use a 10*10 board and pieces are [‘M’, ‘V’] as I described above. I specifically define ‘M’ to be my_player whereas ‘V’ to be opponent_player. The next step is to create a succ(state) function that takes in a board state and returns a list of the legal successors. After having a successor function, I will consider a heuristic function that aims to evaluate non-terminal states and return some float value between 1 and -1. Then, it comes to a minimax algorithm. Specifically, I will have min_value and max_value functions which both utilize the succ and heuristic functions and both take a board state and depth and every subsequent recursive call will increase the value of depth. To have a best next drop, I have a drop_piece(state) function that takes a state and returns the best possible next drop using the minimax algorithm. To define how to win, I wrote a game_value(state) function to check horizonal, vertical, and two directions of diagonal wins. I will also include other relevant functions, including opponent_drop(drop), place_pieces(drop), and print_board().

I want to briefly introduce the setting and rules of the game. As I mentioned before, it is a 10*10 board. ‘V’ goes the first and ‘M’ follows ‘V’ at every step. In the first 10 steps, both ‘V’ and ‘M’ can drop phase once at each step. For each step from step 11 to 13, ‘V’ has two chances to drop phase, but ‘M’ only has one chance to drop.

I hope this can mimic the real situation of the faster spread of virus after some time. Beginning from step 14, it goes back to the original rule in which both ‘V’ and ‘M’ can only drop phase once. This means after wearing masks and having other protection methods, we slow down the speed of infection. Notice that, starting from step 11, players have opportunities to randomly choose one empty grid in any column to place ‘A’ (alcohol) at each round. However, each column can have and only have one alcohol. Since we have 10 columns, players will place 10 alcohol in 10 steps. These ‘A’ pieces are playing the role of barriers that prevent the virus from spreading. Next, I specify how to win the game. If ‘M’ gets six in a row horizontally, vertically, or diagonally, ‘M’ wins. Conversely, the opponent ‘V’ wins. Also, if there is not a winner after 30 steps, we will let ‘M’, the player, win. Since we have 33 grids with virus, it means that we control the virus ratio to about 30% and prevent the virus from spreading fast and widely. Hence, by letting players win, we give them idea that if more people wear masks and have a better sense of self-protection during the pandemic, the spread of coronavirus can be controlled.

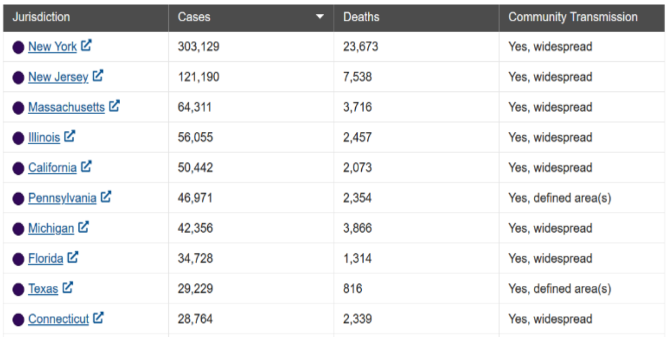
After introducing my preliminary coding steps (necessary functions) and details of the game, I will discuss the GUI. On the front page of GUI, it will show the title of my project which is “Fighting Against Coronavirus” together with a cartoon picture that shows a cartoon character kills several cartoon viruses. Below that, I decide to have two buttons that will link users to other pages. In details, the first one links users to the prediction of coronavirus infection cases, and the other links to the game. Moreover, in GUI, I will change the pieces ‘M’ and ‘V’ to the real images of medical mask and virus, so users will actually get a feeling of stopping the spread of the virus while playing. Another feature of the GUI I want to mention is that there will be some pop-up windows. For example, after step 10, the virus will have two chances to drop phase in each step. At this time, the pop-up window saying “ATTENTION: Not enough people wearing the masks, the spread of the viruses is becoming faster and faster!” After step 13, the pop-up window will show the message “NICE! More people paid attention to this, wore masks and gloves, and disinfected their living spaces using alcohol”. Other choices of information on the pop-up window can be some precautions during COVID-19 at the beginning or the end of the game.

Then, I will explain **verifiable milestones** for each of these steps. For prediction part of the toy example, the milestone to verify the closed form solution of betas is using normalized gradient descent. However, when using normalized gradient descent, I need to choose appropriate T which is the number of iterations of gradient descent and eta which is the parameter in the iterate gradient calculation. I will try different T, from small to large, and different values of eta to see how small the mean squared error (MSE) can be. I also want to confirm if the MSE converges. If MSE converges to the MSE result we got when we compute betas in the closed form, the prediction part will be verified. For the game part, since I first plan to have a toy example, I want to spend time playing the game by myself to see if there are problems in the code. For instance, whether pieces are dropped correctly in each step, and whether the rules of my game are incorporated well without any bugs. I will spend at least about a week to play and write some test codes to test the toy version of my game. If everything goes correctly, I will say that I achieve this step. Finally, to verify if I succeed in incorporating the game and the prediction in the app, similarly, I will play with the GUI and design test codes to test it. Then, I will invite several of my friends who are CS major to play and test it. Finally, I will invite three or four stakeholders (children or teenagers in my family) to play the game. If everything works fine, this step is verified.

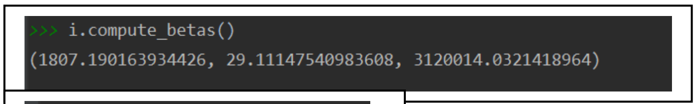
For my toy example, I will give about three weeks in total, including 4 days to write and test the prediction part, a week to design the game, and a week to test the game. For the real GUI, I will leave two weeks to generalize the toy example, including increasing prediction for more regions in the United States and generalizing the game to be closer to the reality. Then, I will leave about two weeks to test GUI, using the milestone I mentioned before. Lastly, I will spend a week to invite my potential stakeholders to experience the app to get some suggestions.

Preliminary Results

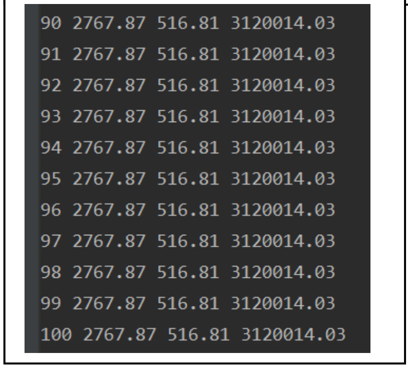
Up to now, I have several preliminary results. First, like what I mention in the Survey and Data Collection part, I gave this survey which asks what the most important things are during COVID-19 to more than 50 people. Based on the results, medical masks, alcohol (spray or pads), and medical gloves are the most popular choices. Due to the survey results, I chose to include the image of medical mask as users’ marker in the GUI. This is important because it helps me to design the game that is as similar to the current COVID-19 situation as possible. Secondly, I received responses from children I interviewed that it is better to include cartoon images in the game. Thirdly, I collected data on COVID-19 reported cases in each state in US from CDC website and got a screenshot of the top 10 states that contain the most cases. Also, I downloaded data about the reported cases in New York from March 3rd, 2020 till May 2nd, 2020 and revised them. In particular, my data has two columns (dates and reported cases), and I changed the dates into numbers for convenient calculation. March 3rd corresponds to 3, March 4th corresponds to 4, and May 2nd corresponds to 63. The reason for assigning 3 to the first date is that the first case of COVID-19 in New York was confirmed on March 1, 2020. This gave me an idea of doing a toy example on New York state's data, which shows the toy example is plausible and is a good starting point. Fourthly, I wrote Python functions to read data, compute closed form betas for the regression, and predict the number of future infections. Some output results are attached below. Interestingly, after doing normalized gradient descent and comparing the result to the closed form, I found the MSE converges. Specifically, I ran normalized gradient descent algorithm with T=100 and eta=0.1 and attached the result of last ten iterations below.



Jurisdiction	Cases	Deaths	Community Transmission
● New York	303,129	23,673	Yes, widespread
● New Jersey	121,190	7,538	Yes, widespread
● Massachusetts	64,311	3,716	Yes, widespread
● Illinois	56,055	2,457	Yes, widespread
● California	50,442	2,073	Yes, widespread
● Pennsylvania	46,971	2,354	Yes, defined area(s)
● Michigan	42,356	3,866	Yes, widespread
● Florida	34,728	1,314	Yes, widespread
● Texas	29,229	816	Yes, defined area(s)
● Connecticut	28,764	2,339	Yes, widespread



```
>>> i.compute_betas()
(1807.190163934426, 29.11147540983608, 3120014.0321418964)
```



```
90 2767.87 516.81 3120014.03
91 2767.87 516.81 3120014.03
92 2767.87 516.81 3120014.03
93 2767.87 516.81 3120014.03
94 2767.87 516.81 3120014.03
95 2767.87 516.81 3120014.03
96 2767.87 516.81 3120014.03
97 2767.87 516.81 3120014.03
98 2767.87 516.81 3120014.03
99 2767.87 516.81 3120014.03
100 2767.87 516.81 3120014.03
```

Similar Project

Since the basic idea of the project is inspired by P6 teeko_player, P6 that we have done this semester is a similar project. Even if some basic steps in P6 such as printing the board, the idea of checking wins, and the idea in the minimax algorithm are similar, I revised it a lot and tried to do more complicated and challenging steps. Different from P6 where each player can only place markers 4 times and need to move their markers around to win the game, my game allows players to continue placing their markers in each turn until someone wins the game. Furthermore, I introduced the idea of letting one player to place markers twice in certain rounds (simulating the faster spread of the virus) and added the blockers (one alcohol in each column) in the board. Also, the conditions of winning are different because players need to get six of their markers horizontally, vertically, or diagonally, which is different from four in a row in P6 and does not have the 2*2 box winning condition. Additionally, my game includes an ending condition that the player will win if there is not a win in 30 steps, and the order of dropping phases is determined before. When I searched on the Internet, I found Gomoku or Five in a Row is similar to mine. The winner of this game is the first player to form an unbroken chain of five stones horizontally, vertically, or diagonally. The conditions of winning are very similar to my game. However, I introduced the idea of placing twice in several rounds and added blockers during gameplay. More importantly, I incorporated elements of COVID-19 to my game. I have the prediction part which provides information of the predicted number of coronavirus infections in the future, so players will have a big picture of what the situation will be and start to prepare preventive measures to protect themselves. Moreover, I think the markers in my game is novel because it matches the theme of fighting against the virus. Creatively, the pop-up windows in GUI emphasize the importance of precautions such as wearing masks and disinfecting your rooms during the pandemic, giving players knowledge of precautions and helping them understand the importance of self-protection. Besides these two projects, I have not found any other projects that are similar to mine and relate to the COVID-19 as well.

References

Cases in the U.S., Centers for Disease Control and Prevention, https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/cases-in-us.html

COVID-19: Data, NYC Health Department, https://www1.nyc.gov/site/doh/covid/covid-19-data.page

COVID-19 Pandemic in New York (state), Wikipedia, https://en.wikipedia.org/wiki/COVID-19_pandemic_in_New_York_(state)

P6: Game AI (given code), CS540 Professors and TAs, https://canvas.wisc.edu/courses/176717/files/11928934