Nora T. Everly

Project Proposal

Product Description
Election Simulator is a game in which you become a candidate who is running for president. You get to decide which party to represent, choose your views, and then try to beat your opponent to win the presidency. By polling, fundraising, running media campaigns, and making speeches, try to win over states so that you can gain their electoral votes in the final election.

Competitive Analysis
There are not many election simulation games out there but I did find some at the start of my project that gave me some inspiration as well as ideas for what not to do. The first result on google is on icivics.org. It is probably the most similar to my project but with several major differences. First off, for every action, you had to answer multiple choice questions that aligned with the view you selected. However, these were extremely simple and unnecessary so my game will not have anything similar. Second, I am weighting my game very differently; they had a very linear structure where 1$ = 1 momentum (similar to influence in my game). Finally, you could only do each move once during a round which I thought took away a large element of strategy (you ended up just doing each move and it got repetitive). The way that I am making my game, there may be times where it will be beneficial to save money and buy more expensive moves, and other times where you will want to do several cheap moves in a row.

Structural Plan

- Main file
  - Some graphics, app code, events code
  - Puts all the backend code together
- State Class file
  - State Class
- Candidate Class file
  - Candidate Class
- Constants file
  - Store constants so as to only have to change them in one place (lists of issues, # of turns, # of rounds)
- End of Round file
  - Functions that are called at end of round (updateMap(), endRound())
  - End of game functions (count electoral votes, declare winner)
- State List file
  - Alphabetical list of states with population and electoral votes
- Moves file

    - Backend for core moves (fundraise, poll, run ads, make speech)
- Computer Player file
    - CPU class (subclass of candidate)
    - Functionality of CPU
- Screens file
    - Contains graphic helper functions for different screens
- CPU Logic file
    - Contains CPU class and logic
- Map Drawing file
    - All draw functions for map

Algorithmic Plan

Interactive Map:

- Created state class
    - Stores information about state (polling data, geometric data, hot topics, which candidate the state is swaying towards, etc)
    - Has different functions that will help later with updating individual states
- Storing every state in a dictionary (key is state, value is state class of state)
- Call createStateDict function in appStarted
- 2 for loops
    - Iterate through list of states, adding to dict, adding pop data, and # of electoral votes, call methods such as generateIssues() and generateWealth()
    - Iterate through rows in geodata, add polygon information
- Whenever a move changes a state, change that state's attributes

Computer Player:

- Instance of candidate class but with more methods
    - Call these methods in move functions
- Reactive strategy to player (plays defense)
- Will play defense so as to not let opposing candidate flip states
- Will optimize fundraising
- Targets close states with more votes

Timeline Plan

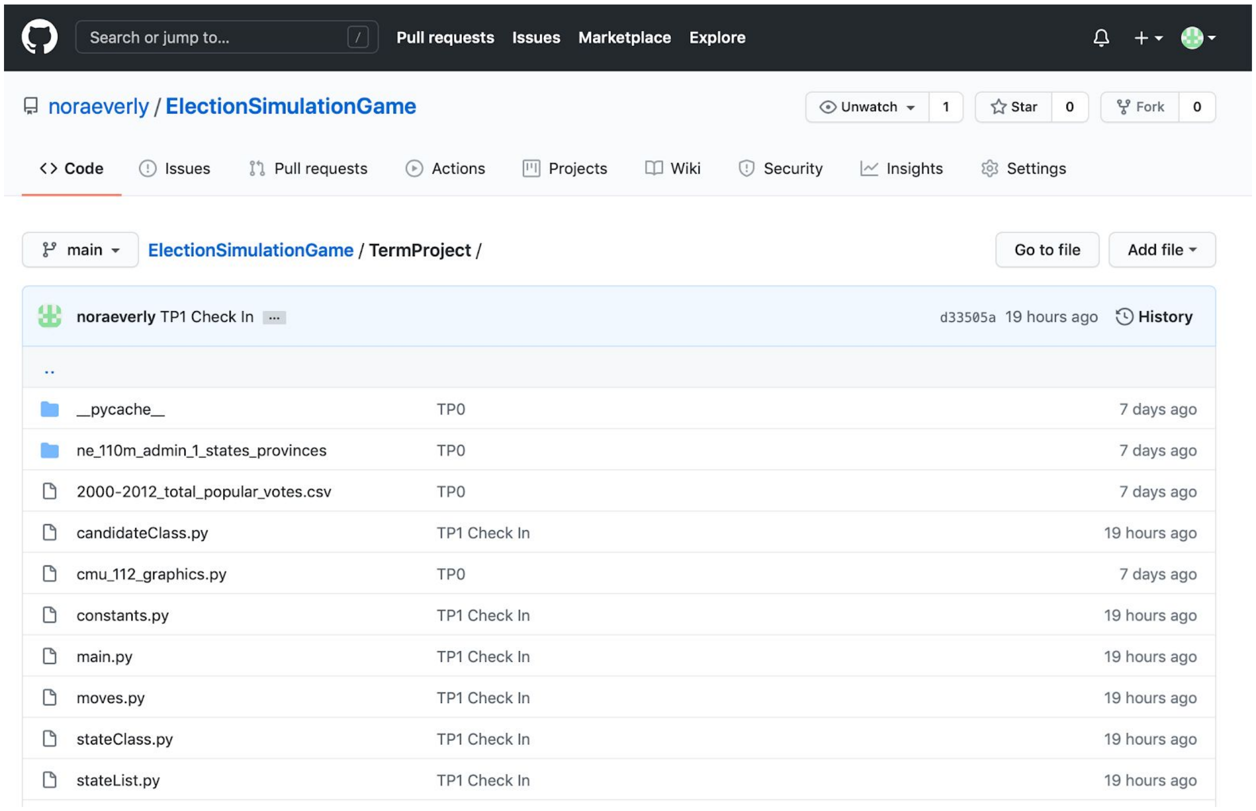
TP0: Interactive map

TP1: Working backend / playable game with rules coded

TP2: Update UI by Thursday (make it more intuitive/functional), finish CPU code by Saturday, add ability to choose issue when campaigning by Wednesday, make game as balanced as possible, title screen, replay button

TP3: Make starting map more realistic, user can customize candidate (issues, maybe how they looks), test UI on friends (get their input), splash pages with information

Version Control Plan
I am using GitHub to back up my code. I upload all of my files about every other time I work on the project.



ElectionSimulationGame / TermProject /

noraeverly TP1 Check In … d33505a 19 hours ago

..		
__pycache__	TP0	7 days ago
ne_110m_admin_1_states_provinces	TP0	7 days ago
2000-2012_total_popular_votes.csv	TP0	7 days ago
candidateClass.py	TP1 Check In	19 hours ago
cmu_112_graphics.py	TP0	7 days ago
constants.py	TP1 Check In	19 hours ago
main.py	TP1 Check In	19 hours ago
moves.py	TP1 Check In	19 hours ago
stateClass.py	TP1 Check In	19 hours ago
stateList.py	TP1 Check In	19 hours ago

Module List
- Geopandas (extension for pandas)

TP2 Update
- Refactored Code (Slightly different structural plan - code organized into more separate files)
- No other major design updates