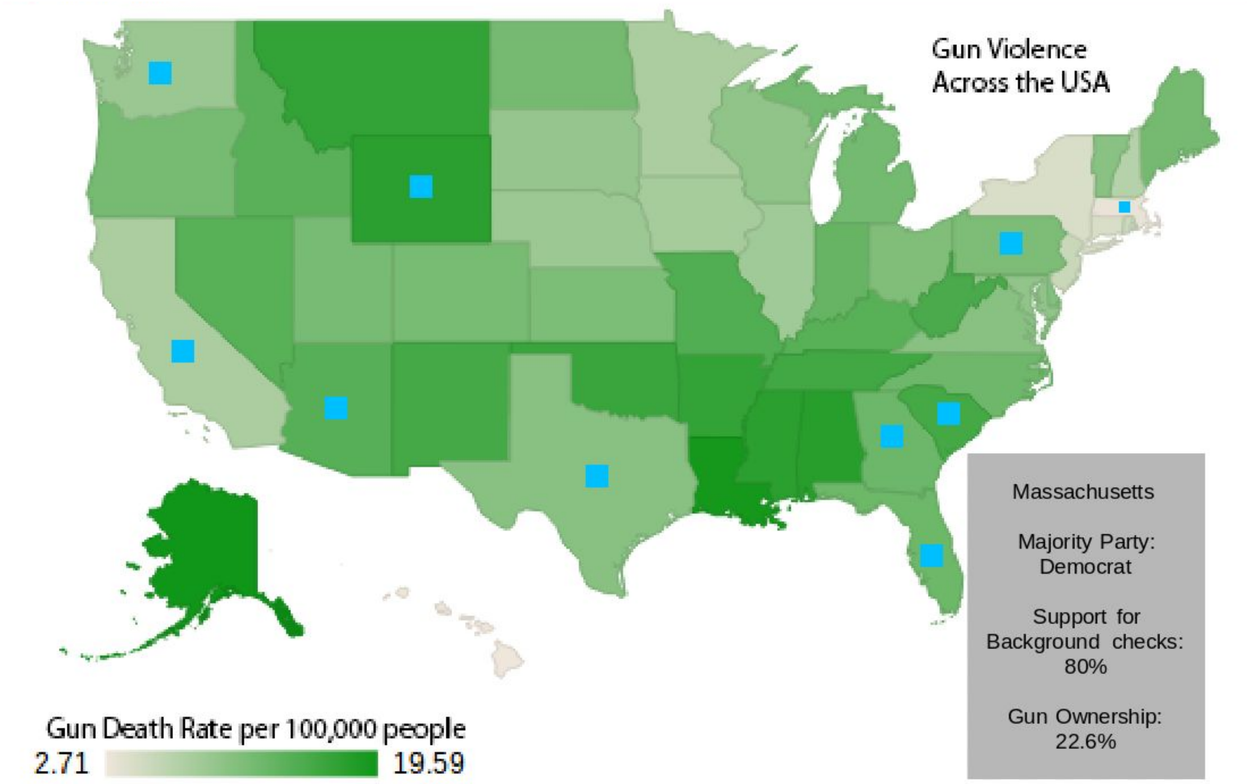
Maya Calabria & Emily Nasiff
Software Design
3/16/18

Reflection

Abstract

We decided to create an interactive data map for the US, that hides and reveals information about gun violence, ownership, etc. by state.

Results



Massachusetts

Majority Party:
Democrat

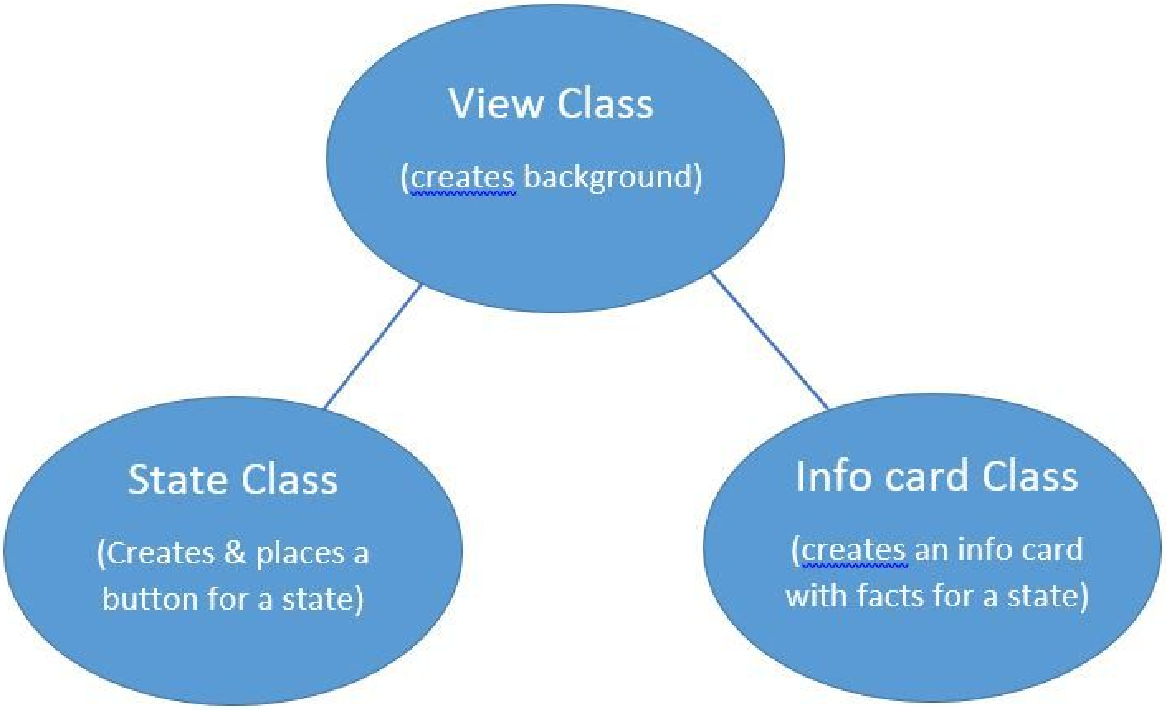
Support for
Background checks:
80%

Gun Ownership:
22.6%

We were able to create a heat map of the US that shows the greenest states as being those most affected by gun violence. For example Alaska has 19.59 gun deaths per every 100,000 citizens each year. The paler the state, the fewer gun deaths per sample size. By clicking on any of the red buttons on this map, users can see more information such as: what percentage of the population owns a gun, how much support there is for stricter background checks, and major political party.

We've included states that fall all across the spectrum to get a good range of information about gun deaths across America. We tend to see that states with higher gun ownership, have larger rates of gun related death, which makes sense and confirms our original hypothesis. In general we also found with the included states, that those that are a majority Republican tend to have higher percentages of gun ownership. Across all states selected, more than half the population supports stricter background checks.

Implementation


View Class
(creates background)
State Class
(Creates & places a button for a state)
Info card Class
(creates an info card with facts for a state)

Our first class within the code is our View class. View creates our background and since this project is a data visualization the background is the image of the map we want the user to interact with. All of the following classes work to make the map interactive. The State class creates and places a blue square onto the designated State. This square acts as a button that triggers an info card to pop up if a click is within the range of a certain states' square. The info card class creates adds an image of a states' info card onto the screen. Our find state function is what tells us which state a click was in. We print the state and use that to create the corresponding info card.

Reflection

From a process point of view, we did a good job breaking down the code into manageable chunks. We wrote a lot of long circuitous code at first that got the model working, and then worked backwards to simplify, shorten, and increase readability. This ended up working really well for us; we didn't worry about making it perfect on the first try.

We wish we knew ahead of time that Pygame does not support multiline text. This ended up changing how we could implement the Infocard class quite a bit; however, we learned a lot about the built in 'blit' function and image layering as a result. This was a great intro to data visualization for both of us, and it seemed appropriately scoped. Unfortunately we spent the majority of our time learning the ropes and working out the kinks and didn’t end up with our ideal stretch goal for final project. Although our end product is simple, it is very adaptable and with more time we could incorporate those extra features.

From a teaming point of view, we did a good job communicating and keeping our code merge conflict free. We planned on pair-programming from the start of our project. We ended up doing the majority of our work sitting together, but working on seperate parts of the code. This was a great model for us, because we were supporting each other the whole time when it came to debugging and staying on task. We could communicate in real time about how one persons changes might affect the others and the program as a whole. Plus, we were both just more motivated while together. It had a lot of the benefits of pair-programming, but it was more efficient because we could both write code at the same time.

Overall we are happy with the final outcome, but wish we had a little more time to include more data sets, and more ways to interact with the map.