**Pre-Planning - 4.2.3 Simulation Project**

**Ideas:**

I chose the **Rebellion** simulation in the NetLogo social science library as my basic template.

* Change initial-cop-density to fear percentage
* Change initial-agent-density to generosity percentage
* Show that fear will lead to less rebellions <- “to be feared is much safer than to be loved”
* Show that leader reputation matters to the public <- “because it is perfectly possible to be feared and not hated”
* Change government-legitimacy to leader-reputation
* Change vision to cop-power <- “leaders must do anything necessary to hold onto power”
* Add a taxes slider
* Add a manipulate slider
* Add an execute button

**Research & Rules:**

My article was “The Morals of the Prince”

4.2.3PD Machiavelli-The-Morals-of-the-Prince.pdf

* Increasing fear-percent results in more cops to control the population
* Increasing generous-percent causes more people to be unloyal to the government and rebel
* Increasing cop-power will give more power to the police, which causes more arrests to happen. This results in more people in jail
* Increasing taxes will anger the population and cause more rebellions
* Increasing manipulation controls the population and causes less rebellions
* Increasing leader reputation causes the people to have a more favorable opinion of their leader, which causes less rebellions
* Increasing jail time means that prisoners stay in jail longer
* Pressing execute-prisoners kills off a certain number of prisoners and causes the population to become less rebellious against the government

**Presentation Questions:**

4a. We chose the Rebellion simulation under the social science library, since this would be the best template for a leader simulation. An “ideal” leader described by Machiavelli should be able to control their subjects no matter what it takes, which is what our simulation shows.

4b. We renamed initial-cop-density to fear-percent, initial-agent-density to generous-percent, vision to cop-power, and government-legitimacy to leader reputation. The code for these functions did not change, we just renamed it to fit our theme. Next, the threshold required for citizens to rebel was changed to a random float so that each setup is unique. (Meaning citizens in every simulation will react differently based on what you’ve stimulated on the population). Then a taxes and manipulation slider was added to the simulation. If the taxes get too high, then the population will rebel. When the manipulation slider is increased, then the population begins to calm down. Finally, the execute-prisoners kills of a certain number of prisoners and then increases the threshold to simulate terrifying the population.

4c. The graphs in the simulation show who is rebelling (red/active), who is quiet, and who is in jail for each tick. There are also small counters for you to see the exact number of rebels, police officers, quiet people, jailed, and active people. The visualization shows everyone in the simulation. People with a body of yellow mean that they aren’t rebelling against the government. People with a blue body means that they are a police officer. People with a red body means that they are rebelling. People with a black body means that they are in jail. The darker the head is, the more that individual dislikes the government.

4d. We observed that when fear-percent is increased, there are more initial police officers. When generous-percent is increased, there are more people who dislike the government. Cop-power affects how far an officer can see a rebel, and arrest him. Taxes cause more people to rebel against the government when the slider is increased. The manipulation slider causes less rebellions when it is increased. A low Leader-reputation will cause more rebellions, and jail-time affects how long people are in prison.

4e. There are now many variables affecting the state of the citizens, meaning it would be hard to predict each outcome since each variable affects the simulation in different ways. Also, the threshold for citizens to rebel is a random number from 0-1.5, which means that each simulation will always be different. There are many other variables in the simulation that are also based on random floats, meaning that it would be impossible to predict the outcome each time. If each variable was made a constant va

**Create Prompt:**