Nick Aksamit (6177000)

Using MVC:

Model (model > Model.java) 🡪 Holds all information about the state of things

* Holds village information
* Holds GUI state information, like if in building mode, training mode, upgrade mode, or attack mode
* Holds additional GUI information, like the mouse position, and whether a building was selected for construction (selectedNewConstruction), building selected for upgrade (selectedForUpgrade), or if a combatant is selected for placement in attack mode (selectedForAttackPlacement)
* In particular, the model also holds information for Attacks, like the attacking and defending village
* There is only one thing that is constantly updated in the model as it needs to be, and that is the mouse position

Controller (controller > Controller.java) 🡪 Controls what happens based on the model and view

* What happens on mouse clicks, depending where the mouse is positioned
* What events can occur (like clicking or dragging) on the toolbar at the bottom or on the main screen depending on the mode (such as attack, build, training, upgrade)
* Uses the view to get collision detection of mouse and a symbol (which selects a new contruction or new combatee)

View (view > View.java) 🡪 Controls what is viewed on the screen depending on the model

* Mostly depends on the mode, such as what is shown depending on the model’s mode (like attack mode, building mode, training mode, upgrading mode)

I didn’t get to finish the generation aspect of the game, but I was going to use a genetic algorithm that I learned from 3P71 in order to generate a village, and the creation of the buildings in this algorithm used the Abstract Factory design, found in utility > Factory.java

Saving and loading of the main village can be found in the utility > GameState.java class. I preferred to use JSON for this, and used this: <https://mvnrepository.com/artifact/org.json/json> as a library to help me create the JSON object within Java.