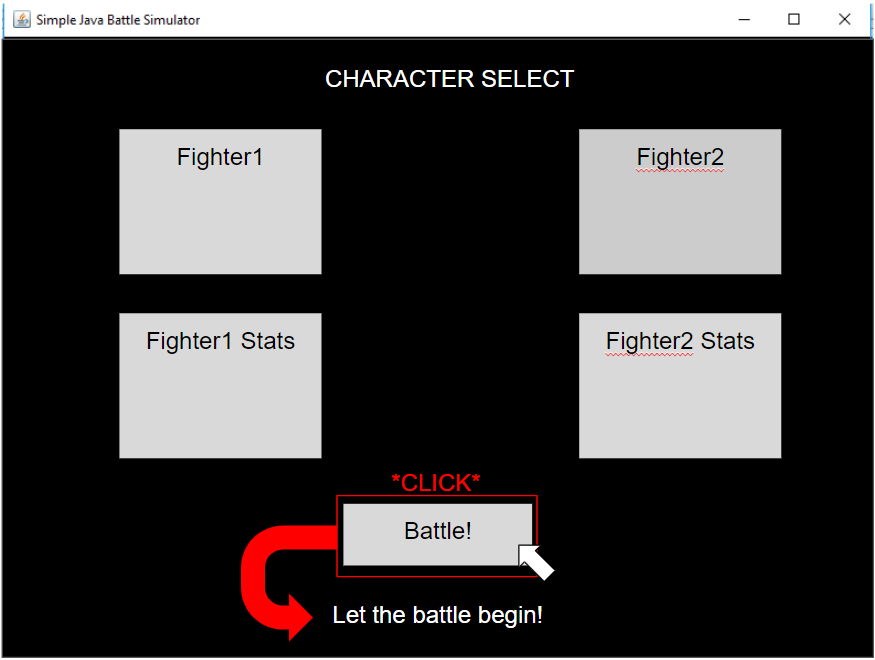
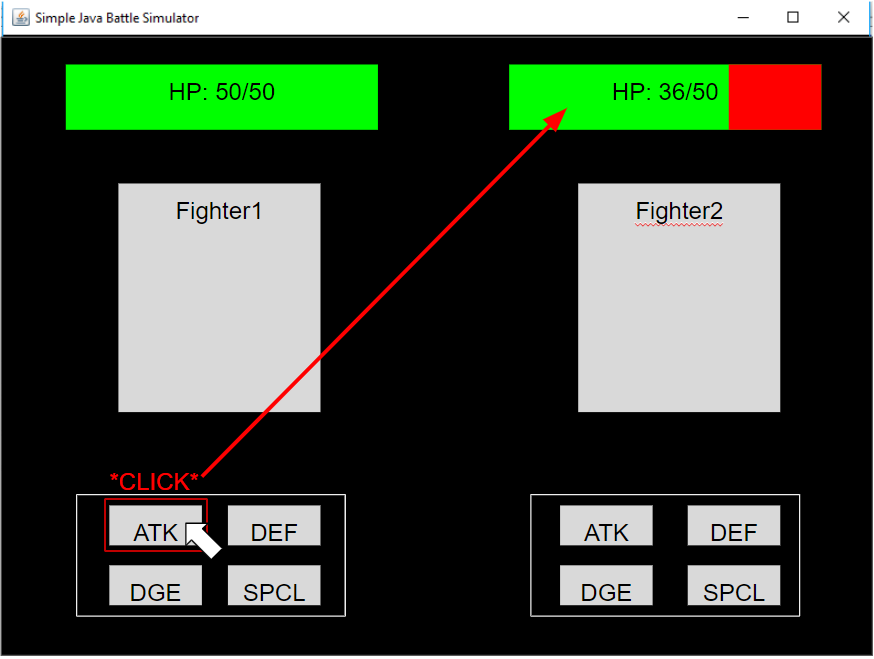
* Describe in detail all new and changed functionality your team intends to implement for project deliverable 2.
  1. **NEW:** Implement the first iteration of a turn-based game-loop
     1. I will be creating a new class called “Game” or “Battle” that will take in two Fighter objects, a ‘Game State’ variable (boolean or int) and create a method that takes the ‘Game State’ variable to indicate a player’s turn (ex. boolean player1Turn = true/false or int playerTurn = 1 or 2). For example if player1Turn = true, then it’s player1’s time to choose a move, and after choosing a move it will set that variable to false to move onto Player 2’s turn.
  2. **NEW:** Implementing the ‘Attack’ functionality (Fighter1 attacks Fighter2 means Fighter2’s HP is decreased by Fighter1’s ATK stat) in either one of two ways:
     1. Create a new method in the new ‘Game’ or ‘Battle’ class that will execute the action.
     2. Create a new method in the Fighter class that will execute the action.
  3. **NEW:** Creation of a ‘Battle’ view/screen that will show the two selected Fighters, their HP, and their moves (for an example look at Figure 2).
  4. **CHANGED/NEW:** Adding more Fighters to the ‘Character Select’ screen and add a ‘Battle’ button, which upon being clicked’ will show a message (for an example look at Figure 1) and then transition the screen to the ‘Battle’ view/screen (Figure 2).
* Describe how you will implement this functionality. This can be done through design documents (such as those submitted in preparation for project deliverable 1), or through detailed descriptions of changes to or new classes, methods, and data. Be especially careful to document information flow between MVC components and all changes required in all of these components.
  1. Creating a new class that will be controlled by the ‘Controller’ class.
     1. The new class will be instantiated by the Controller once it has received the request to from the View (when the ‘Battle’ button on the ‘Character Select’ screen is pressed and the View has moved to the ‘Battle’ screen).
  2. Calling the ‘Attack’ method on another Fighter object will start from clicking the ‘ATK’ button on the ‘Battle’ screen, which will send a request from the View to the Controller, and then from the Controller to the Model. The Model will recognize the request and then update the Fighter object’s HP stat that is being attacked. The new HP stat will be then sent back down through the Controller and to the View to display for the players to see.
  3. Handled by the View, which will be appended with new code that will instantiate the ‘Battle’ screen and transition the same way that my ‘Starting’ screen goes to my ‘Character Select’ screen.
  4. Handled by the View which will be added on with new code that will display 1 or 2 more fighters on the ‘Character Select’ screen in the same manner as the first fighter (button image of fighter and then their description box). Will also feature a new button that displays a message after being clicked on. This button will instantiate the ‘Battle’ class and change the screen to the ‘Battle’ screen.
* Provide descriptions (or drawings of) any new user interface elements.
  + Figure 1: New/Appended ‘Character Select’ screen.
  + Figure 2: ‘Battle’ screen showing off attacking the opposing fighter.
* Describe how you will test this functionality.
  + Clicking the ‘Battle’ button on the ‘Character Select’ screen will display a message and then promptly switch screens, showing that the button worked.
  + Clicking the ‘ATK’ button on the ‘Battle’ screen should decrease the HP number of the opposing fighter and display a red area over the green health bar to indicate missing health.
  + After one player attacks, that player’s buttons should not be able to be clicked on since their turn is considered “over” and must wait for the other player to choose the ‘ATK’ option, vice versa. This signifies the work of the ‘Battle’/’Game’ class and it’s turn-taking method.
* Describe the responsibilities of project team members in implementing this new functionality.
  + Since I am now working by myself, I will be responsible for implementing all of these new changes/additions in the game. Adding the new class, methods, fighters, buttons, messages, screen; everything will be done by me.