Jasmine Guevara

12/18/2020

CST338-40

Text Game GUI with MVC Model

Understand the Application

We will be developing a GUI text adventure game using Swing, AWT, and the MVC design model. We accept user input with buttons, and we use the input to forward the game into the next scene.

The Program Specs

Phase 1: Create view, model, and controller classes

You need to set up view, model, and controller classes appropriately. Remember, the view is used to display the output after the controller manipulates the model. So, first, you should create the view, which has nothing to do with calculations. Then, create the bones of the controller. Afterwards, the model can be developed. When the model is good to go, then the controller can be used to update data onto the view after it has received data from the model class.

You need 3 classes and a main:

1. Create the gameView class
2. Create the gameModel class
3. Create the gameController class
4. Establish the main method

Make sure to follow the MVC design. When this is done, you can move on to create the view in the next phase.

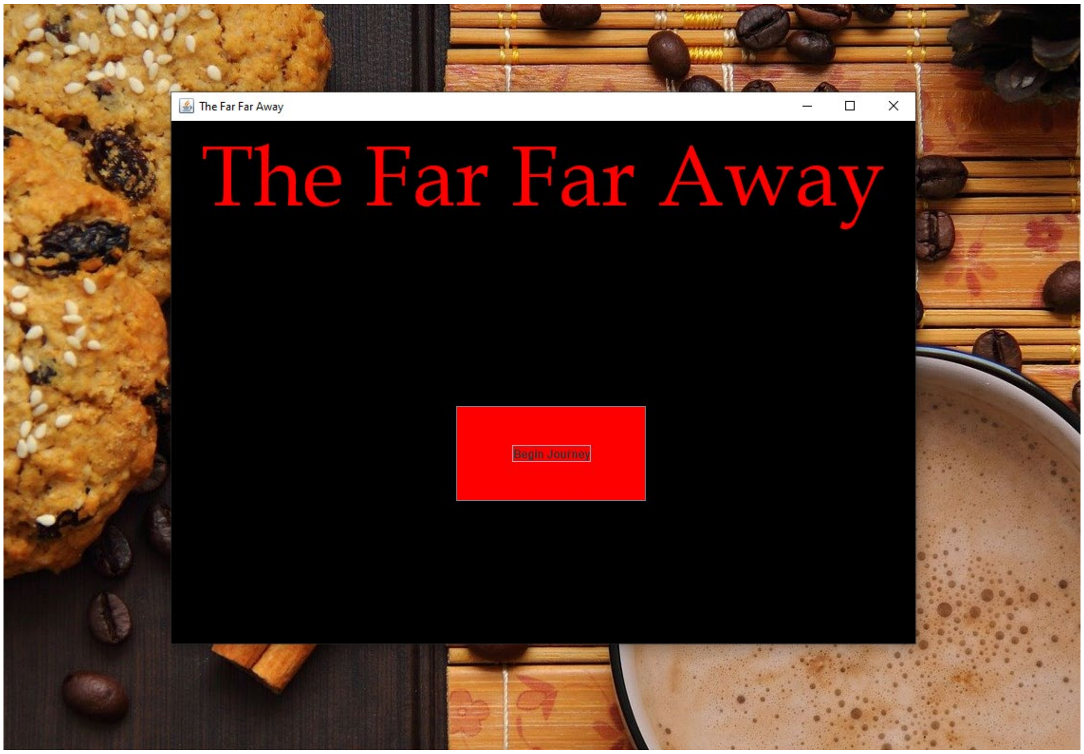
Phase 2: Create the title screen/main menu and start button

I recommend starting with gameView, so you can set up where the game actions happen. In our game, we will have a title screen/main menu and a start button to start playing the game. When using a button to perform an action, we need to implement ActionListeners.

You need to:

- Change the name of the window with setTitle()
- Set up a main game JPanel area, title JPanel and title JLabel, and JButton for user to start the game
- Change the background color of the panels and the start button
- Adjust title and button so they look presentable

Screenshot of Phase 2 Output:



Part 3: Create the Choose-A-Character UI

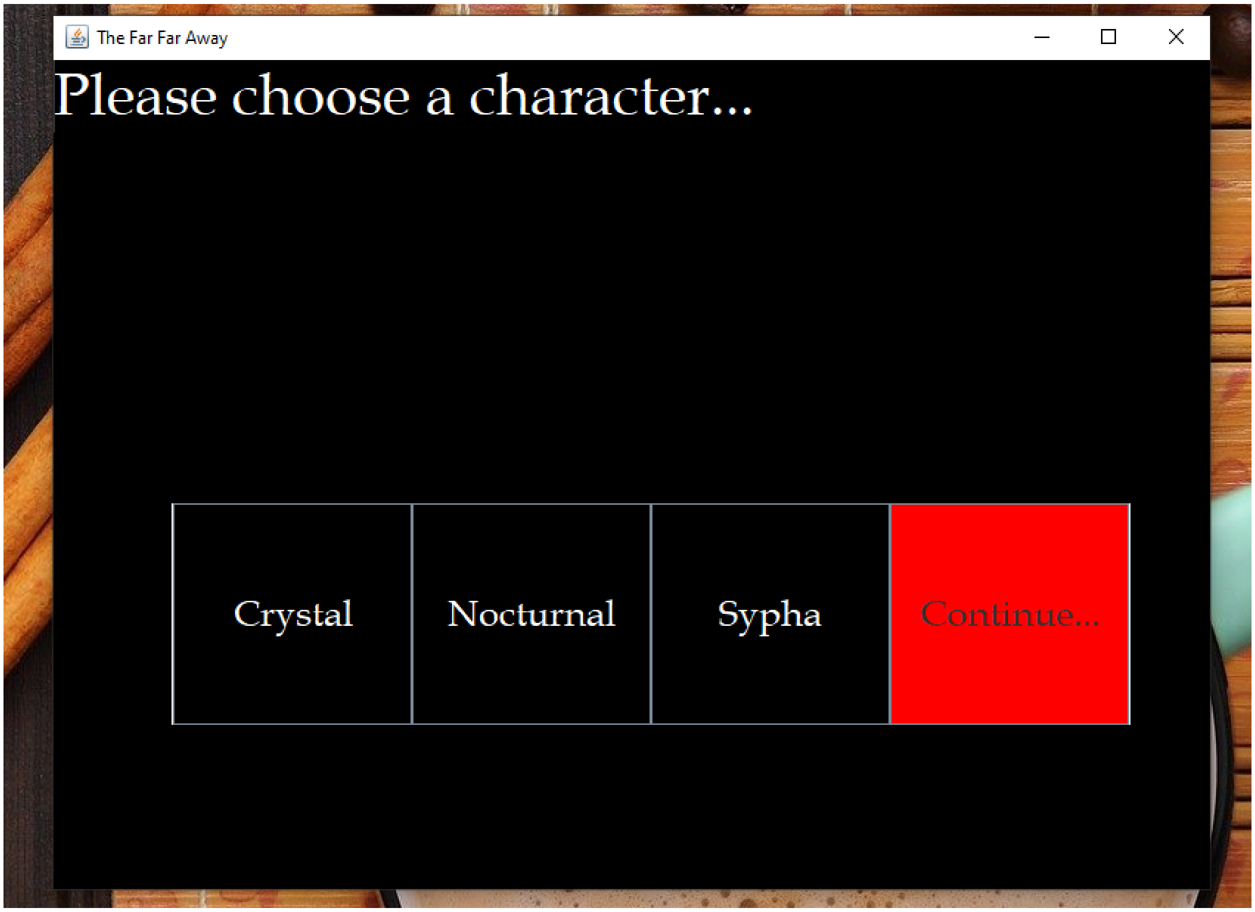
You will need some characters for the user to pick from. When the user selects a character, their character information is updated. Remember, when updating the character information, you need to use the controller to update gameModel then have gameView update the screen accordingly.

You need to:

- Set up a JPanel for the buttons. You can have as many characters as you want, but 3 would look nice, since there is still the need to have a connection between choosing a character and playing the game afterwards.
- You need a JButton to continue onto the next screen. You will also need JButtons for the characters that are available to select from.

- You will need to have JTextArea or JLabel to display which part of the game the user is at.
- You will also have to store the information of the character chosen to the gameModel.

Screenshot of Phase 3:



Part 4: Create the game UI.

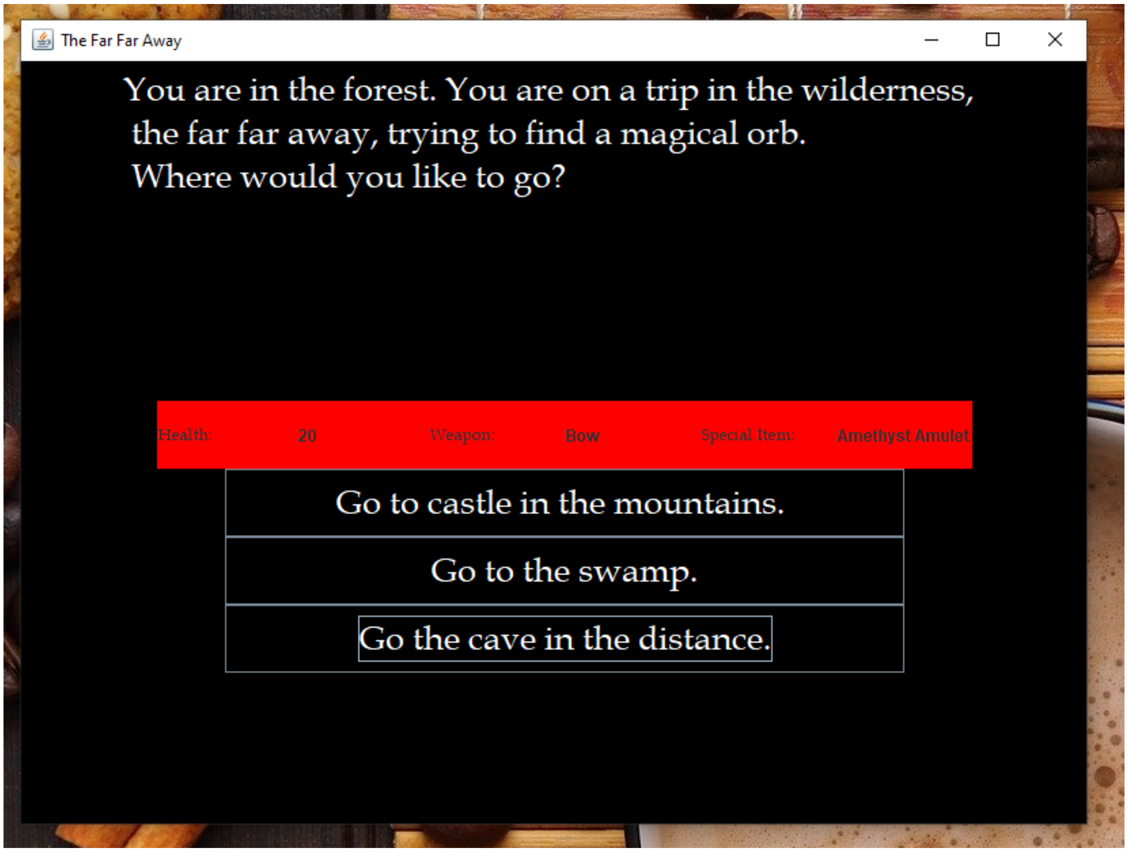
This is where the game scenes will be played. There are options available to the user to progress forward in the game. In my example, I used 3 buttons for options. There should be a JTextArea that displays the game text content on the screen.

There is also an area where the character's status bar should display health points, weapon being used, and a special item.

You need to:

- You will need to create a JTextArea to display the game text content
- You will need to create JButtons and implement listeners
- You will need to create JPanel for the character's status bar/ information

Screenshot of Phase 4 Output:



You are in the forest. You are on a trip in the wilderness,
the far far away, trying to find a magical orb.
Where would you like to go?

Health: 20 Weapon: Bow Special Item: Amethyst Amulet

Part 5: Create the game storyline.

This is your story, so get creative! In my story, I have three magical women who are searching for a rare orb. The journey leads them to a far and mysterious place.

Test run:

Make sure to progress all the way to the end of your story! It will need to have a final ending to ensure the user understands it is the end.

Game over:

When the user's health reaches to zero, the game will display a Game Over title screen.