William Le, Kamran Hussain, Kevin Choi
417 Expectation Failed
5/10/2021
Intro to Programming Java
Ed Taylor

Beta Release, 5/10/21
Version 0.0.5

Dependency on the Resources Folder

ReadMe

Program Purpose:
Bob's Kitchen is a game where players strive to make the best dish they can! The game is coded in Java with a clean and easy to use

Target User Profile:
This game is intended for high school students who are interested in cooking and food.

Feature List:

- A clean and easy to use home page with a help window.
- Confirmations before playing each level/dish
- Clearly labeled options for user choices and inputs
- User options are simple and easy to select
- Submit page provides a summary of the user selected options
- Grader can grade a grilled cheese sandwich

Class List: Main, Home, ConfirmSandwichPlay, ConfirmBurgerPlay, ConfirmPancakePlay, ConfirmRamenPlay, ConfirmWellingtonPlay, ConfirmExit, Help, SandwichStart, SandwichVeggies, SandwichMeat, Sandwich condiments, SandwichFinish

Team Responsibility:
Kevin Choi - Made all the images needed to run the program. Made a label for each image (if needed) and found images appropriate for the program to run smoothly. Research color and themes for the program. Implemented the game engine.

Kamran Hussain - Implemented features in the program, specifically the graphics in the sandwich level. Took the sketches from Will and put them into coded classes primarily graphics. Began work on developing the game engine with Kevin and Will, starting to implement the sandwich scoring system when the user inputs are completed. Changed the code so it only uses one frame.

William Le: Listed Fields, Methods, and Constructors in each class via comments. Drew out by hand and made a UML Diagram for the program including arrows that show hierarchy and class paths. I came up with the idea of a cooking game including the game engine (how the game works), the various dishes that we will soon have, and I drew out some sketches on what the game should look like. Helped with the game engine. Currently, working on the scoring engine of the game, I have not figured out how to put the scoring system in the gui, but when I do I'll put the two classes in the pre production release.

Known Bugs/Workarounds:
-User selected fields do not deactivate when the page is reloaded
-Drop menu images do not update images
-Sandwich Condiments window is not formatted correctly
-Formatting in the finish window right gridlayout jpanel is not functioning properly, formatting is all on one line, GridLayout may be implemented
-Multiple Frames are needed for stability, a single, refreshing frame is in the works
-Only Sandwich level is functioning

Key Learnings:
-Learned how to implement graphics spanning multiple types of input fields across multiple fields.

-Learned how to use Github for fast file transfers and collaboration
-Learned how to create images based off written request

Credit List:
Image sources, game engine (Kevin), GUI, Algorithm(Kamram) UML Diagram, game engine(Will)

Referenced the uml diagram from the final project, and applied what we learned about class hierarchies and "depends on" relationships between the class to draw out the class paths.

Oracle Java Documentation API