

CIS2168 006 Assignment 2

Simple Secret Face Guessing Game

1. Objectives

- Learn how to program using inheritance, static fields/methods, method overriding
- Learn how to write java application that contains more than one method
- Have some fun with basic 2D GUI game programming

2. Overview

You write a java game with a graphical user interface (GUI), not command line console. The game asks the user to guess the shape and facial expression of 4 random secret faces. The player gets feedback in terms of correctness and sees the secret face at each guess. At the end, the player's score, as the percentage of correct guesses, is displayed.

- 1) Display the game instruction
- 2) Ask user to guess the shape of the secret face: only 2 choices: Box or Ellipse
- 3) Ask user to guess the facial expression of the secret face: only 3 choices: Smiling, Frowning, Serious
- 4) Generate the secret face randomly from 6 choices:
 - 3 ellipse faces: smiling, serious, frowning
 - 3 box faces: smiling, serious, frowning.
- 5) Display feedback to user (Correct or Wrong guess). And display the secret face.
- 6) Let the player guess 4 secret faces totally (repeat step 2) to step 5) 4 times totally
- 7) Display the player's game score (the percentage of correct guess)

3. Implementation Requirements

- You must use the classes AbstractFace, BoxFace, EllipseFace in the project Faces2 and package faces2. You cannot change the definitions of them. You must use them as is.
 - These classes will make your programming much easier. The default basic facial expression is already done in AbstractFace. The shapes are already done in BoxFace and EllipseFace.
- You must implement classes SmilingBoxFace, FrowningBoxFace, SeriousBoxFace, SmilingEllipseFace, FrowningEllipseFace, SeriousEllipseFace.
 - The first three classes SmilingBoxFace, FrowningBoxFace, SeriousBoxFace must inherit and extend class BoxFace.
 - The last three classes SmilingEllipseFace, FrowningEllipseFace, SeriousEllipseFace must inherit and extend class EllipseFace.
 - **Note:** the shapes Box or Ellipse are already implemented in BoxFace or EllipseFace. Your 6 subclasses need to override the method drawFacialExpression(), which is initially implemented in the base class AbstractFace.

- This means you must also use classes in SimpleGUI project.
- Your code must generate each secret face **randomly** and only from 6 possibilities: 2 shapes (Box, Ellipse) by 3 facial expressions (Smiling, Frowning, Serious).
- Your code must display everything in one single GUI window. The window content can be refreshed multiple times with different contents displayed each time. Do not use command line user interface. Game instructions in texts must be displayed in the GUI window too.
- Your code must let user know and enter **b** for Box shape, **e** for Ellipse shape.
- Your code must let user know and enter **m** for Smiling, **f** for Frowning, and **r** for Serious facial expressions.
- Your code must let user enter any key to start the game after the game instruction is shown.
- Your code must pause for 2 seconds after each secret face is shown.
- Your code must display the player's score as something like 75% or 75, which 3 correct guesses out of 4.
- Your Java application (the class with main() method) must contain at least two methods: main() and another method for the game play. You can add more methods if you want to.

Bonus points:

You will get bonus if you draw more beautiful faces and make the user interface better looking. But you will NOT get bonus if your programs do not complete all minimum required functionalities and requirements first.

4. Submission Requirements & Grading

This assignment is **due by 11:50PM, Thursday, September 10, 2015**.

Please see the file CIS2168 006 Assign2 Submission Requirements.pdf for more details.

5. Major Steps

- a. Add 6 subclasses: SmilingBoxFace, FrowningBoxFace,..., SmilingEllipseFace, etc . And test each class use a driver class (a class with main() method and creates some objects of the class implemented)
- b. Add the Java application with main method.
- c. Add the other method in the Java application that contains the game play code. You can name this method mainGame(..) or just playGame(..).

6. Detailed Hints

Implement 6 subclasses

- Can simply use mouth shapes to represent facial expressions. Don't have to change nose and eyes.
- First implement SeriousEllipseFace, then SmilingEllipseFace, then FrowningEllipseFace. Then implement the other 3 classes. Remember to add constructors too.

Workflow in the framing program main()

- 1) Show game instructions
- 2) Wait for user input to start the game
- 3) Initialize the game
- 4) Call playGame() to play the game.
- 5) Show the game score

Workflow in the main game playGame()

- 1) Initialization
- 2) Ask user to guess the face shape and get user input
- 3) Ask user to guess the face expression and get user input
- 4) Create and display a random secret face and display user feedback
- 5) If not 4 faces yet, go back to 2)
- 6) Compute and return the score

Technical Details

- Display texts in fonts and colors in the window. See the demo program in SimpleGUI project: Demo03b_DifferentFonts.java
 - Methods in class SimpleGUI: setFont(...), drawText(...)
- Clear the screen
 - Method in class SimpleGUI: eraseAllDrawables()
- Pause the program for some time in milliseconds
 - Method in class SimpleGUI: pauseProgram(time in milliseconds);
- Keyboard input
 - Method in class SimpleGUI: keyReadChar()
- Random number generation
 - Math.random() creates a random double between 0 and 1 (1 excluded).
- Draw items in one single window
 - Need to create a SimpleGUI object and assign the reference to this object (say sgInMain) to the static data field sg in class AbstractFace