

Assignment 4: Online Hangman Game (25% of course total) Marking Guide

This assignment must be done individually. This guide is for reference only and might be changed.

Total = [30] marks. Use the formula $\text{Quiz_completed?}(\text{Quiz_mark} + \text{Program_mark}) : (0.5 * \text{Program_mark})$.

Quiz_mark

[5] Quiz: Online quiz.

Program_mark has three parts [15]+[5]+[5]

0 on Program_mark if cannot compile (communicate this to the student on comments), or uses Javascript in any of the .html files.

[15] Gameplay

- [2] welcome: display the “welcome.html” template, with the following components (0.5 each)
 - A welcome message
 - Information of the creator (student)
 - Rules of the game
 - A button to start the game
- [3] game: display the “game.html” template, with the following components (0.5 each)
 - A message indicating the ID of the current game
 - The status of the game (Active)
 - The progress of the game
 - The information of the game (i.e., some form of representation of the “hangman”, number of guesses made, number of incorrect guesses)
 - A way for the player to guess a number (e.g., a textbox + a button, buttons for all 26 characters)
 - A link taking the player back to the /welcome URL
- [3] game/id: display the game.html template with the game indicated by id
 - [1] If the game with the id exists, display the “game.html” template populated with the correct information (must remember the progress and information of the game)
 - [1] If the game with the id exists and is either won/lost, display the “gameover.html” template with appropriate message, the answer, and a link taking the player back to the /welcome URL
 - [1] If the game with the id doesn't exist, show the “gamenotfound.html” template
 - [0.5] Provide a message indicating the page is not found
 - [0.5] Provide a link taking the player back to the /welcome URL
- [2] Everytime the player starts a game:
 - [1] A new ID is generated
 - [1] A new word is randomly selected
- [2.5] The player loses a game
 - [1] By making 8 incorrect guesses
 - [0.5] Duplicate incorrect guesses are counted separately

- [0.5] Display the “gameover.html” template with appropriate message, the answer, and a link taking the player back to the /welcome URL
- [2.5] The player wins a game
 - [1] By making guessing the word correctly with at most 7 incorrect guesses
 - [0.5] Duplicate correct guesses are counted separately
 - [0.5] Empty guesses do not crash the game, and are not counted towards any guesses
 - [0.5] Display the “gameover.html” template with appropriate a message, the answer, and a link taking the player back to the /welcome URL

*no need to check the URL after each step. Some students might choose to take the player to the URL “game/id” for a new game. This is a valid approach.

[5] File Structures & General aesthetics

- [1] Correct package hierarchy
 - Package ca.cmpt213.a4.onlinehangman contains Application.java
 - Package ca.cmpt213.a4.onlinehangman.controllers contains HangmanController.java, GameNotFoundException.java, and any related controller/exception classes (if any)
 - Package ca.cmpt213.a4.onlinehangman.model contains Game.java, and any related model classes (if any)
 - Folders resources/* contain corresponding view components
- [4] Look & feel of the game interface (Grade gets to decide on 0.5 increments)
 - [1] font is used in a consistent way across all pages
 - [1] welcome page shown in a clear and pleasant way
 - [1] progress and information of the game shown in a clear and pleasant way
 - [1] gameover page shown in a clear and pleasant way

[5] Code Quality and Style Guide

- Reasonable object-oriented structure
 - must separate packages for UI and game logic.
- Very minor violations of style guide have no penalty
 - (e.g., having "int myCount=0;" (spacing wrong) once)
- Lose a few marks for consistent problems
 - (like always getting the spacing wrong).
- Larger penalties possible for horrific code (0 on Program_mark)

Some specifics to check with suggested deductions (to Program_mark)

- -3 Poor/missing JavaDoc comment on a class (not needed on methods and fields)
- -5 Incorrect indentation, brackets, or spacing (use IDE's reformat if needed).
- -5 Poor intention revealing class, method/variable names.
 - (e.g., making everything public or static, undecipherable names)
- -3 Bad display of information

Forward to instructor if...

- Material is suspiciously similar to another submission or code posted online.