Pong GUI Assignment

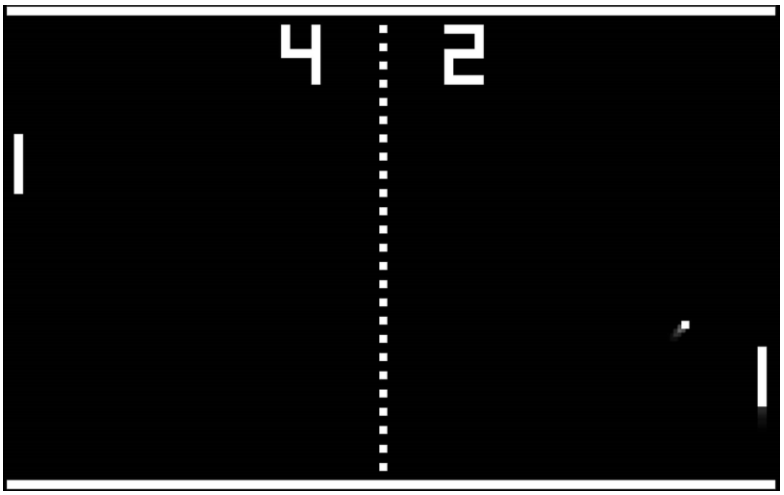
Task

Your task is to create the widely known video game Pong. You are to use the code given to you in the 5.3 lesson as the basis for your program.

Example Output

In a traditional pong game, the background is black, with white graphics representing the player paddles, ball, and score.

You do not need to follow this colour scheme.



Requirements

The program must include all of the following:

- Must be a form of the pong game
- Ball must realistically bounce off paddles and screen edges
- Score must calculate properly
- One paddle must be controlled by the player
- The second paddle can be either controlled by the computer, or player controlled
- Proper header comments
- Proper body comments
- Proper variable names
- Appropriate spacing and indentation
- Other professional formatting (no unused variables, etc.)
- User-friendly (instructions and controls are clear, etc.)
- An end condition (the program must at some point conclude with a winner or loser)
- Has no syntax errors (it runs successfully)
- Must be a robust program (no noticeable runtime or logic errors)
- Must divide program up into methods and classes (use the 4.3 lesson as a template)
- Must be coded in Java

Completing one of the following challenge features can earn a grade up to 100%. Completing none caps the grade at 80%:

- Play music or other sound effects in the program (not possible in repl.it with java)
- Utilize a functional timer (this could be used to award points based on timing, or to control an end condition for the game)
- Save data to a file for some useful purpose (saving high scores, returning to a game in progress, etc.) If saving high scores, they must somehow be displayed / accessible in the game
- A power-up system (reaching a certain score unlocks benefits, or can spend points on a power up, or if losing by x points get power ups, or start game with limited number of power ups, etc.)
- Replace the basic shape graphics with 2D images. For example, the ball could instead be an image of a cannon ball, and the paddles could be little pirate ships. Note that these images do not need to be animated (sprites)

Submission Details

Due date: Monday May 22nd, 2023 at the start of class.

Submission format: Submissions will be made through Brightspace Assignment tab. Files must be placed in a single folder, with your name, and compressed (zip).

Submission must include a batch file (.bat) which runs the program automatically. This will be explained in class.

Additional Details: unlike previous assignments, there will be no opportunity to re-submit work. Only one submission will be accepted per student.

This assignment will have only half the weighting of previous assignments on your grade.

	Level 4	Level 3	Level 2	Level 1
Code Format	The code is properly formatted. There is a header, thorough use of body comments, proper variable names, good use of white space, and generally good code structure.	The code is mostly properly formatted. However, the comments may not always be helpful or descriptive, or there be poor variable names, or slightly disorganized code structure.	The code is not formatted well. Variable names, comments, white space, and/or general code structure have not been organized well.	The code is very poorly formatted. No attention has been paid to formatting techniques such as comments or variable names.
Required Features	The program includes all required features. The features have been executed effectively and thoroughly.	The program includes most or all features. However, either execution is sloppy or one feature is missing.	The program only includes some required features. It is significantly limited in what it can do.	The program includes few if any functioning features. Substantial work is needed.
Additional Features	The program includes one additional feature. Well done!	The program does not include any additional features. You should aim higher!	The program does not include any additional features. The program appears to represent an absolute minimum of work.	The program includes an attempt at additional features that causes major errors and ruins execution.
Creative Solutions	The implementation is effective, efficient, and unique. The writer of this code is working beyond what is expected of them!	The implementation is effective and efficient. The writer of this code is working at the level expected of them.	The implementation is effective. The writer of this code should think about how to make more efficient and direct solutions in the future.	The implementation is ineffective. The solution is incomplete and/or extensively error-prone.

Mark Breakdown

Knowledge	• Communication	Application	Thinking
/10	**/10**	**/10**	**/10**