Exercise 9 (10 points) – can be done individually or in pair

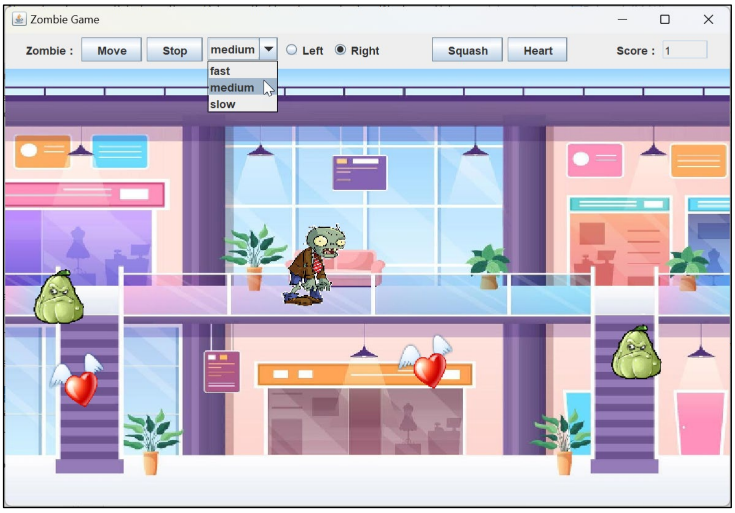
- The first lines of all source files must be comment containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members.
- Put all files (source, input, output) in folder Ex9_xxx where xxx = your full ID. That is, your source files must be in package Ex9_xxx and input/output files (if there is any) must be read from/write to this folder. From now on, you’ll get point deduction for wrong package & folder structure.
- The group representative zips Ex9_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted.
- The exercise is graded only once, and after graded, members can’t be added.

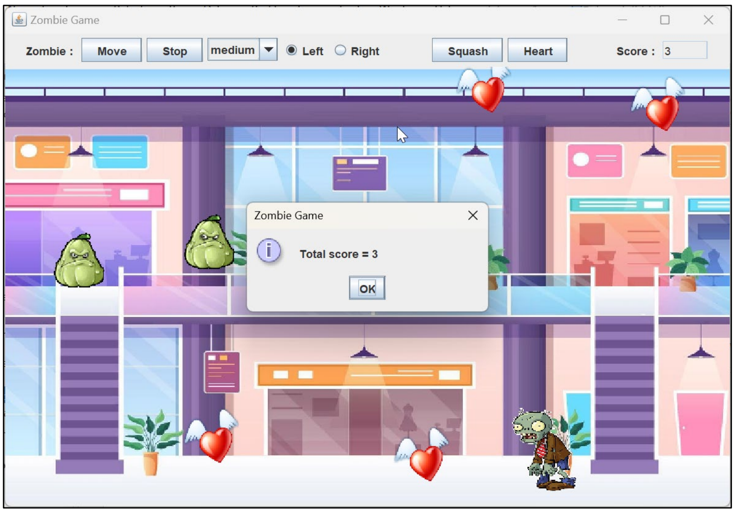
==

Use the given image/sound files and source file (MainApplication.java). Unzip resources.zip and put this folder in your project folder (Ex9_xxx)

Complete the source file to make program work as follows:

Zombie and items (Squashes and Hearts) are controlled by separate threads.





Total score = 3

1. Zombie
   1.1 Move & Stop buttons to move & stop.

   1.2 Combo box to set Zombie's speed.

   1.3 Radio buttons to turn left & move to the left, or turn right & move to the right. When reaching one side of the frame, it'll appear on the other side and switch between upper & lower floors.

2. Item
   2.1 Squash button to add a Squash at (random X, top Y) location and let it fall down. When the Squash hits Zombie, decrease the score.

   2.2 Heart button to add a Heart at (random X, bottom Y) location and let it float up. When the Heart hits Zombie, increase the score.

   2.3 When a Squash hits Zombie that is standing or walking on the upper floor, also push Zombie down to the lower floor.

4. All listener classes must be anonymous classes. Add listeners as follows

4.1 Add ActionListener to Move & Stop buttons, to make Zombie move or stop

- Move → create and start zombieThread
- Stop → stop zombieThread

4.2 Add ItemListener to combo box, to set Zombie's speed

- Fast = short sleeping time for zombieThread
- Slow = long sleeping time for zombieThread

4.3 Add ItemListener to each radio button, to set Zombie's direction

4.4 Add ActionListener to Squash & Heart button, to add a Squash & Heart. It can be done by creating & starting a new itemThread (each item is controlled by each thread)

4.5 Add WindowListener to the frame, to stop all threads and show the final score when closing it

5. Use zombieThread & itemThreads to make all labels move automatically. Anonymous class can also be applied. Complete method setItemThread and class ItemLabel, using example from setZombieThread and ZombieLabel

6. Complete method updateScore to increase/decrease score when Squash/Heart hits Zombie. This method requires proper synchronization because it can be called by multiple itemThreads at the same time (if >1 items overlap with Zombie)

- Making individual itemThreads check collision with Zombie is more efficient than making only 1 zombieThread check collisions with all items

All given code can be modified as needed