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

- The first lines of all source files must be comments containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members

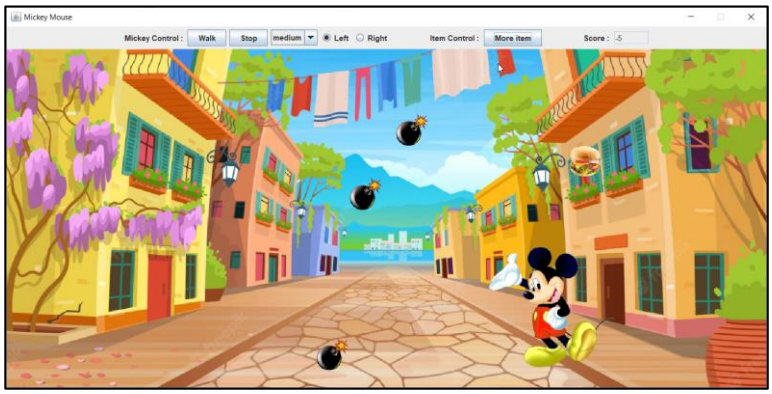
- Put all files (source, input, readme.txt) in folder Ex9_xxx where xxx = ID of the group representative, i.e. your source files must be in package Ex9_xxx (assumedly in Maven's src/main/java). Input files must be read from this path

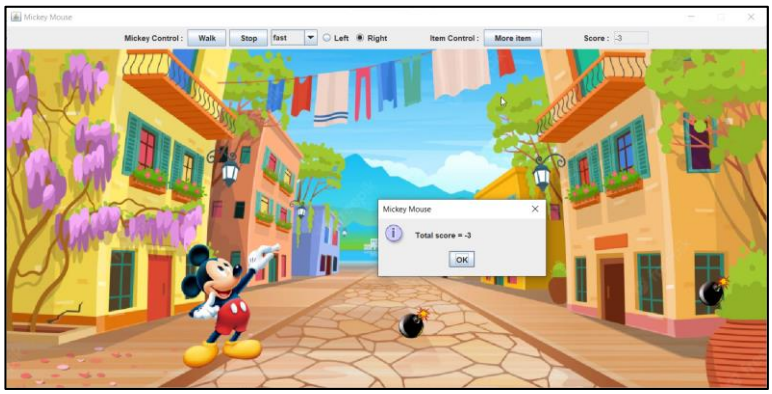
- The group representative zips Ex9_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted

==========

Use the given image/sound files and source file (MickeyFrame.java). Unzip resources.zip and put this folder in your project folder (Ex9_xxx). Complete the source file to make program work as follows:

Mickey and falling items (burgers and bombs) are controlled by separate threads





1. Mickey control
   1.1 Walk & Stop buttons to walk & stop
   1.2 Combo box to set Mickey's speed
   1.3 Radio buttons to turn left & walk to the left, or turn right & walk to the right. When reaching one side of the frame, he'll appear on the other side
2. Item control
   2.1 More button to random 1 falling item (burger or bomb)
   2.2 Update score when each item hits Mickey
3. Report total score when closing frame

4. All listener classes must be anonymous classes. Add listeners as follows
   4.1 Add ActionListener to Walk & Stop buttons, to make Mickey walk or stop
   - Walk → create and start mickeyThread
   - Stop → stop mickeyThread

   4.2 Add ItemListener to combo box, to set Mickey's speed
   - Fast = short sleeping time for mickeyThread
   - Slow = long sleeping time for mickeyThread

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

   4.4 Add ActionListener to More button, to add a falling item. 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 show the final score when closing it

5. Use mickeyThread & itemThread to make all labels move automatically. Anonymous class can also be applied. Complete method setItemThread and class ItemLabel, using example from setMickeyThread and MickeyLabel

6. Complete method updateScore to increase/decrease score when an item hits Mickey. This method requires proper synchronization because it can be called by multiple itemThreads

All given code can be modified as needed