Exercise 7 (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 Ex7_xxx where xxx = ID of the group representative. That is, your source files must be in package Ex7_xxx and input files must be read from this path

- The group representative zips Ex7_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted

==========

Use the given image files and source file (MainApplication.java). Unzip resources.zip and put this folder in your project folder (Ex7_xxx)

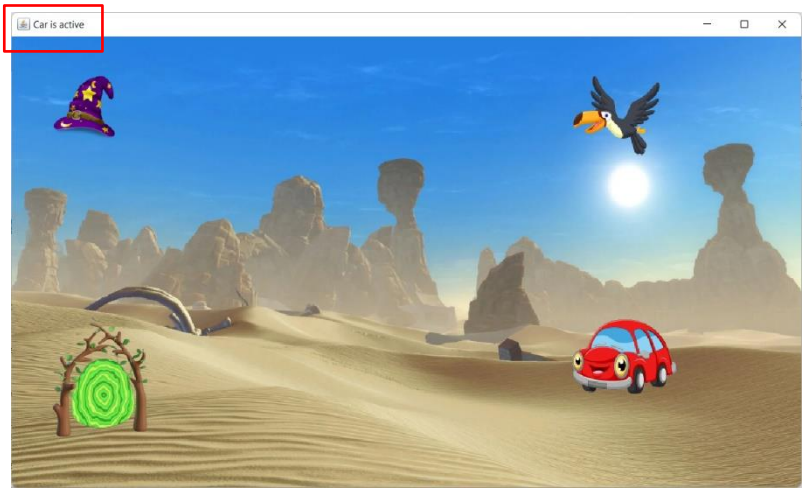
Complete the source file to make the program work as follows:

There are 2 different types of labels

- CharacterLabels (carLabel, birdLabel) can be moved by arrow keys
- ItemLabels (hatLable, portalLabel) can be move by mouse drag

1. Either carLabel or birdLabel can be moved by arrow keys at a time (the moving CharacterLabel = activeLabel)

2. Use alphabet key C to switch to Car and key B to switch to Bird. Active character must be shown on title bar



3. When Car (i.e. carLabel) is active
   - Can move left/right by arrow keys LEFT/RIGHT. Once it reaches one side of the frame, it'll appear on the opposite side

4. When Bird (i.e. birdLabel) is active
   - Can move left/right by arrow keys LEFT/RIGHT. Once it reaches one side of the frame, it'll appear on the opposite side
   - Can also move up/down by arrow keys UP/DOWN, but only within the frame

5. Portal (i.e. portalLabel)
   - Can be dragged within the frame at any time by using mouse
   - When it is dragged on top of other labels, there won't be any effect

6. Hat (i.e. hatLabel)
   - Can be dragged within the frame at any time by using mouse
   - When it is dragged on top of the Car being activeLabel, the Car will disappear. Once it is dragged out, the Car will reappear at its original location
   - When it is dragged on top of the Bird being activeLabel, the Bird will change its location to appear at the Portal
   - But when it is dragged on top of the other labels not being activeLabel, there won't be any effect

7. Complete `class MainApplication extends JFrame implements KeyListener`

   JLabel cannot hear KeyEvent. We have to make JFrame hear & handle KeyEvent on its behalf. And because JFrame can handle one JLabel at a time, we will make it handle activeLabel which can be either carLabel or birdLabel

   7.1 Set activeLabel to carLabel or birdLabel when alphabet key B/C is pressed
   7.2 Make activeLabel move (e.g. call moveUp/moveDown/moveLeft/moveRight) when arrow key is pressed
   7.3 Add variables/methods or make further modifications as needed

8. Complete `class CharacterLabel extends BaseLabel`. We will create carLabel and birdLabel from this class
   8.1 Add methods to update its location according to horizontalMove & verticalMove
   8.2 Add method to make it disappear (portalMove is false for Car) or change location to Portal's location (portalMove is true for Bird)

   Note: use `setIcon(null)` to make the label disappear
   use `setIcon(icon)` to make it reappear

   8.2 Add variables/methods or make further modifications as needed

9. Complete `class ItemLabel extends BaseLabel implements MouseMotionListener`. We will create portalLabel and hatLabel from this class
   9.1 Add method to update its location upon mouse drag. If hitEffect is true (for Hat), also check whether it overlaps with activeLabel in order to make the activeLabel disappear or change location

   Note: to check whether 2 labels overlap

```
if ( this.getBounds().intersects(activeLabel.getBounds()) )
   activeLabel.doSomething();
```

   9. Add variables/methods or make further modifications as needed