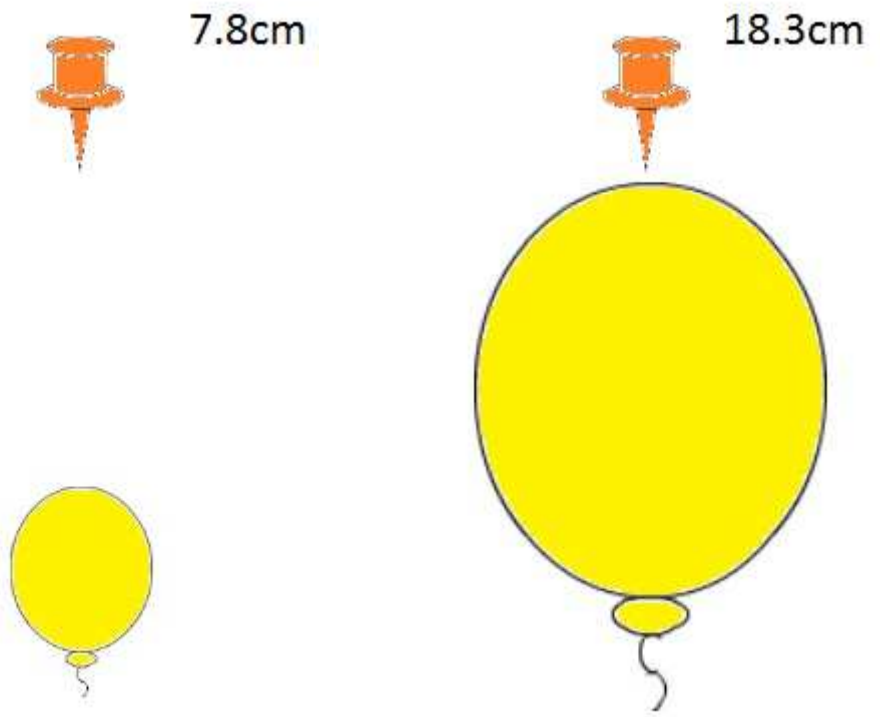
Question 5 **[12 marks]**

"Balloon Piercing" is a game in which player needs to blow the balloon as big as it can. After the game started, the balloon is blown automatically. The player can stop it by tapping on the screen. Once the balloon is stopped from blowing, the game ends and the size of the balloon will be sketched on the screen. However, there is a thumb-pin inside the game. The game ends when the balloon is pierced by the thumb-pin and "Game Over" message will be shown to the player.



(a) (b)

Figure 5. Game "Balloon Piercing".

The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Balloon Piercing" game. The design should be specific to this game.

[12]

[END OF QUIZ PAPER]

Question 5 **[12 marks]**

Handle user input
- **Detect if the player taps on the screen to stop blowing the balloon**
- **If so,**
    - **Draw the size of the balloon on the screen**
    - **Exit game loop (or End game)**

Update game state
- **Update the size of the ball**

Collision detection
- **Evaluate if the balloon collided with the thumb-pin**
- **If so,**
    - **Draw "Game Over" message on the screen**
    - **Exit game loop (or End game)**

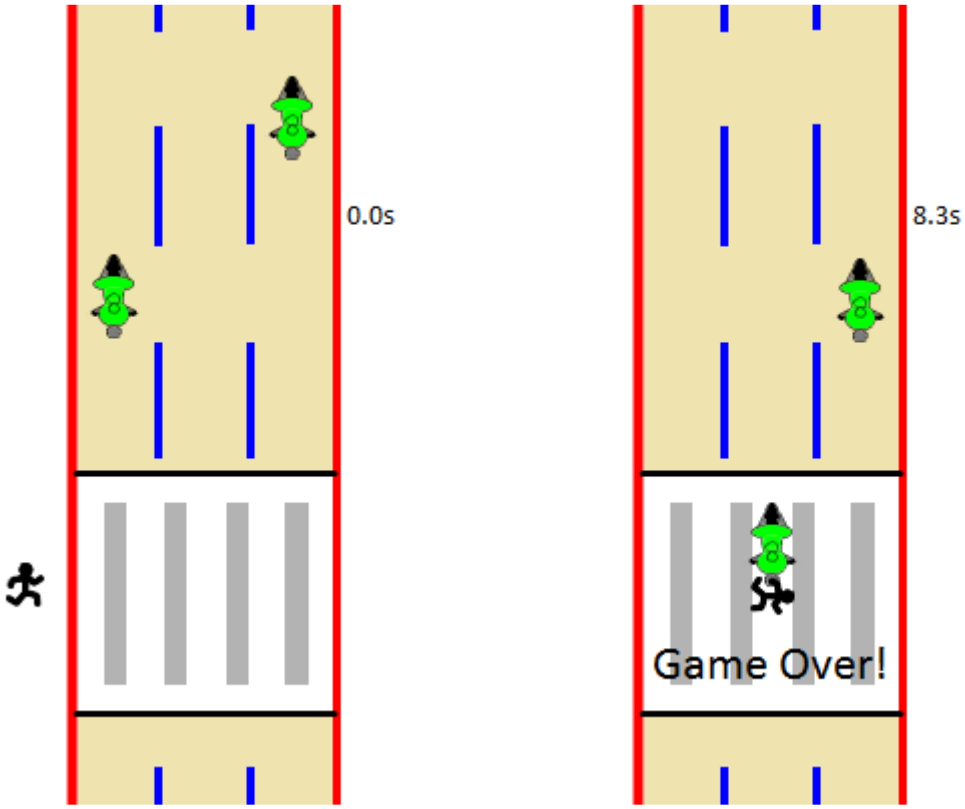
Refresh the screen (or Update display)

Sleep

[END OF QUIZ PAPER]

Question 5 **[13 marks]**

"Cross the danger" is a game in which player needs to control the pedestrian to cross the road as fast as he can. After the game started, motorcycles are driving on the road. The player can change the moving direction of the pedestrian in order to cross the road by tapping on the screen. Once the pedestrian crosses the road successfully, the game ends and the total time elapsed will be sketched on the screen. When the pedestrian is crashed by motorcycle, the game ends and "Game Over!" message will be shown to the player.



(a) (b)

Figure 5. Game "Cross the danger".

The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Cross the danger" game. The design should be specific to this game.

[13]

[END OF QUIZ PAPER]

Question 5 **[13 marks]**

Handle user input
- **Evaluate the moving direction of the pedestrian**

Update game state
- **Update the position of the pedestrian**
- **Update the position of the motorbikes**
- **Update the time elapsed**

If the pedestrian collided with any motorcycle
- **Draw "Game Over!" on the screen**
- **Exit game loop (or End game)**

If the pedestrian cross the road successfully
- **Draw the time elapsed on the screen**
- **Exit game loop (or End game)**

Generate motorcycle randomly

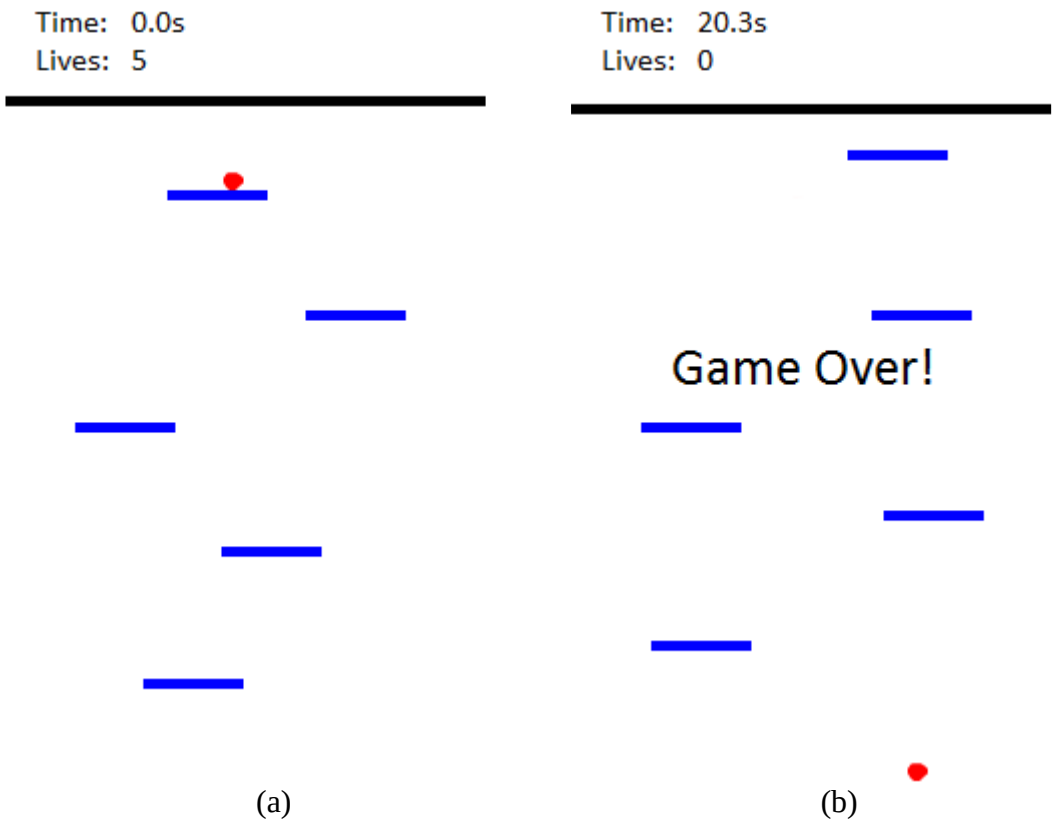
Refresh the screen (or Update display)

Sleep

[END OF QUIZ PAPER]

Question 5 **[15 marks]**

"Falling Ball" is a fantasy game in which player needs to control the ball movement with finger tap for avoiding collision with arena boundary with the help of paddle. When the game started, the ball will fall from the top to the bottom while the paddles will move upwards for preventing the ball from reaching the bottom boundary. The ball will move to the right when the player tapping on the right hand side of the ball and move to the left when the player tapping on the other side. The player should keep the ball surviving within the arena by using the paddles as long as possible. When the ball collided with the boundary of the arena, life of the player is lost. When all the preset lives of the player are lost, the game ends and "Game Over!" message will be shown to the player.



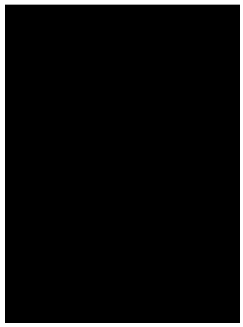
(a) (b)

Figure 5. Game "Falling Ball".

The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Falling Ball" game. The design should be specific to this game.

[15]

[END OF QUIZ PAPER]



Question 5 **[15 marks]**

Handle user input
- **Evaluate the moving direction of the ball (left/right)**

Update game state
- **Update the position of the ball**
- **Update the position of the paddle**
- **Update the time elapsed**

If the ball collided with the paddle OR
If move out of the paddle
- **Change the moving direction of the ball in y-axis**

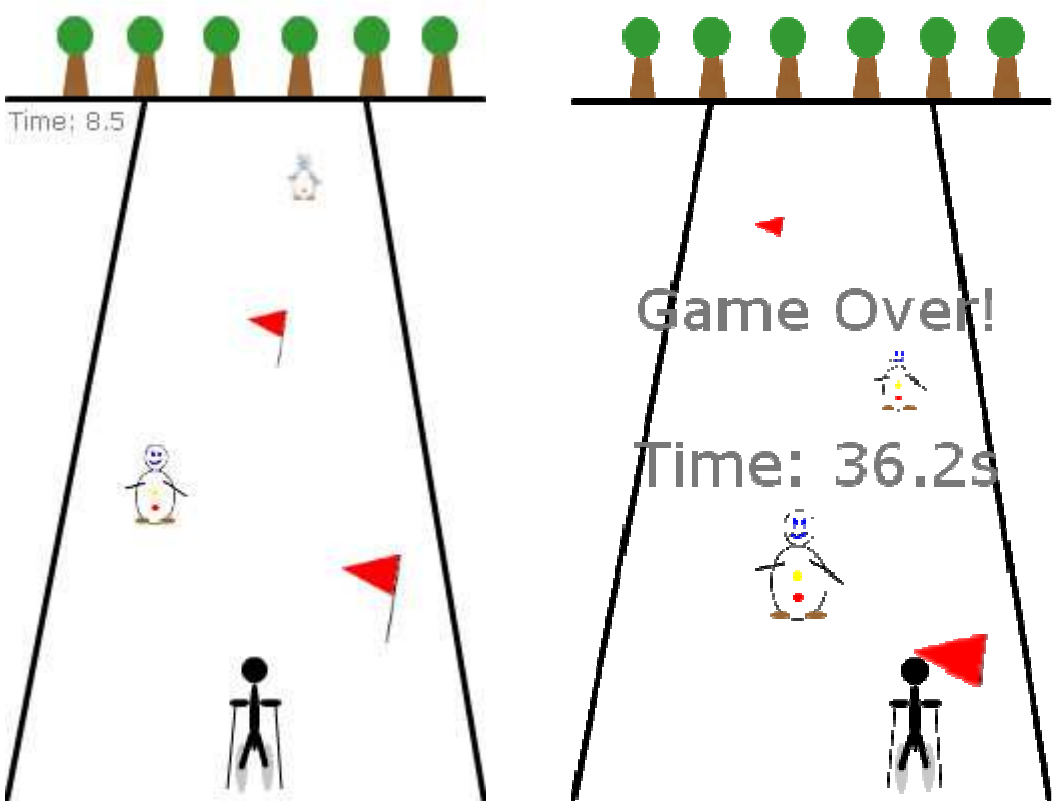
If the ball collided with boundary of the screen
- **Draw "Game Over!" on the screen**
- **Exit game loop (or End game)**

Generate paddle randomly
Refresh the screen (or Update display)
Sleep

[15 marks, 1 mark for each operation]

Question 5 **[13 marks]**

"Skiing" is a classic sport game in which skier is gliding down the steep mountain quickly. The player needs to tap on the screen to controls the moving direction of the skier to avoid crashing on the snowman and flag. In order to increase the playability, the speed of the skier will be increased regularly.



(a) (b)

Figure 2. Game "Skiing".

The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Skiing" game. The design should be specific to this game.

[13]

Question 5 **[13 marks]**

Handle user input
- **Calculate the direction of the skier going to move**

Update game state
- **Update the position of the skier**
- **Update the position of the obstacles (i.e., the flag and snowman)**
- **Scale the obstacles (as the size of the obstacle should be larger if it moves closer to the skier)**
- **Update total time elapsed**

If the skier collided with either the flag or the snowman
- **Exit game loop (or End game) (or Check and handle loss conditions)**

Otherwise
- **Generate snowman and flag randomly**
- **Increase the moving speed periodically**

Refresh the screen (or Update display)

Sleep

Question 5 **[15 marks]**

"Car Racing" is a sport game in which player partakes in a racing competition by controlling the car movement with finger tap. The following screen shows a circular car racing track with a start/finish line. The aim of the game is to control a racing car to dash around the track for a number of times and to finish within a time limit. To finish the race quicker, the racing car must be kept on the track. While on the track, the racing car will speed up until the maximum speed. If the racing car has hit the dirt (outside the track), the racing car will slow down drastically.

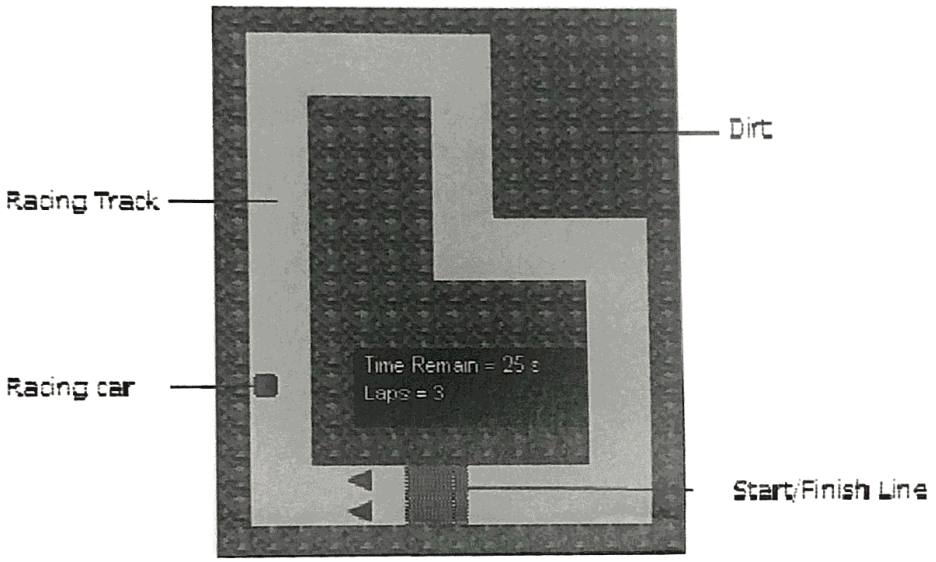


Figure 5. Game "Car Racing".

The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Car Racing" game. The design should be specific to this game.

[15]

[END OF QUIZ PAPER]

Question 7 **[15 marks]**

"Breakout" is a game in which the player controls the movement of a paddle horizontally to bounce a ball and hit some bricks. The player wins when all bricks have been hit and gone, and loses when the ball reaches the bottom of the play area.

(a) (b) (c)

Figure 7. Game "Breakout".

(a) The game will provide two screens:
(1) The menu screen for starting the game and quitting the game.
(2) The play screen for actual game playing.
How many Activities should there be in the game?

[1]

(b) Consider the game-play for this game. **Discuss** each of the following game-play elements: (1) victory condition(s) for human player, and (2) the main game-play challenge.

[4]

(c) The main game loop is the place where the game logic and operation are integrated together. **Design** the main game loop in pseudo code form for this "Breakout" game. The design should be specific to this game.

[10]

[End of Question 7]

Question 7 **[15 marks]**

(a) **There should be at least 2 Activities, one for each screen.**

(b)
- **The player wins when all bricks have been hit and gone.**
- **Main game challenges include physical coordination of the paddle to keep the ball from going beyond the paddle.**

(c)

Save the position of the ball
Handle user input
Update game state
- **Move the ball**

Collision detection
- **Handle if collision detected**
  - **Restore saved position**
  - **If the ball reach the bottom of the play area (game over)**
    - **Exit game loop / End game**
    - **(Check and handle loss conditions)**
  - **If all bricks have been hit and gone (game win)**
    - **Proceed to next level**
    - **(Check and handle victory conditions)**
  - **Otherwise**
    - **Bounce the ball to opposite position**

Refresh the screen (or Update display)
Sleep

[End of Question 7]