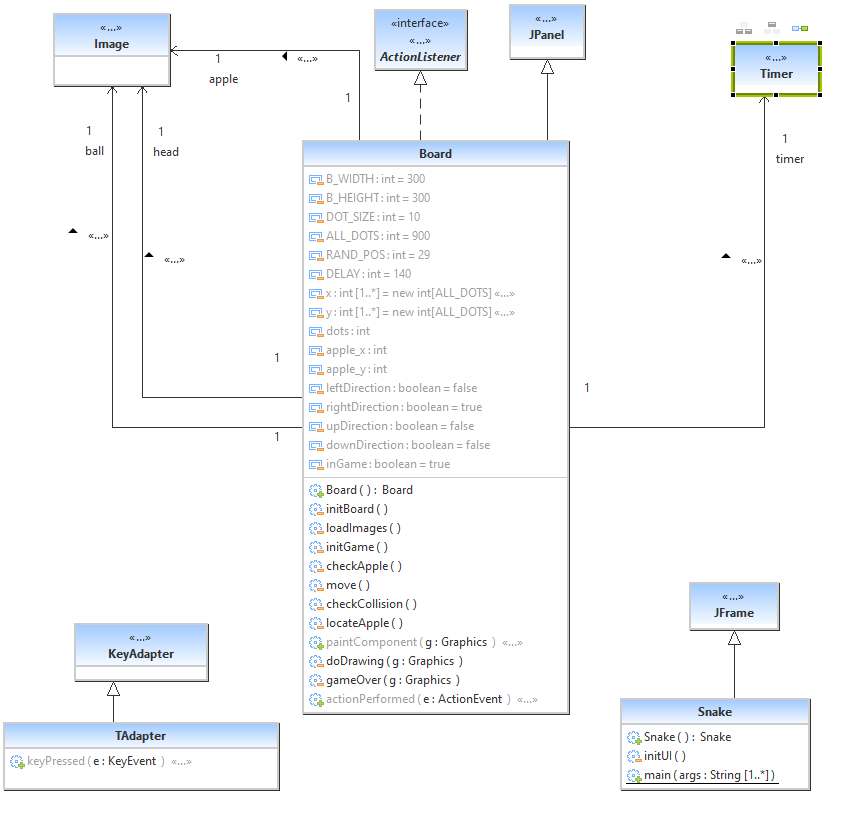
## I. Class Diagram



## III. Pseudo Code

a. Board Subsystem:

class Board extends JPanel implements ActionListener {

//Define constants used in snake game

//The B\_WIDTH and B\_HEIGHT constants determine the size of the board.

private final int B\_WIDTH = 300;

private final int B\_HEIGHT = 300;

//DOT\_SIZE is the size of the apple and the dot of the snake.

private final int DOT\_SIZE = 10;

//ALL\_DOTS constant defines maximum number of possible dots on board (900 = (300\*300)/(10\*10)).

private final int ALL\_DOTS = 900;

//RAND\_POS constant is used to calculate a random position for an apple.

private final int RAND\_POS = 29;

//DELAY constant determines the speed of the game.

private final int DELAY = 140;

// Both arrays stores x and y coordinates of all joints of snake.

private final int x[] = new int[ALL\_DOTS];

private final int y[] = new int[ALL\_DOTS];

}

private void loadImages() {

//The ImageIcon class is used for displaying PNG images.

ImageIcon iid = new ImageIcon("src/resources/dot.png");

ball = iid.getImage();

ImageIcon iia = new ImageIcon("src/resources/apple.png");

apple = iia.getImage();

ImageIcon iih = new ImageIcon("src/resources/head.png");

head = iih.getImage();

}

private void initGame() {

//The initGame() method create the snake, randomly locate an apple on the board, and start the timer.

dots = 3;

for (int z = 0; z < dots; z++) {

x[z] = 50 - z \* 10;

y[z] = 50;

}

locateApple();

timer = new Timer(DELAY, this);

timer.start();

}

private void checkApple() {

//If the apple collides with the head, we increase the number of joints of the snake.

if ((x[0] == apple\_x) && (y[0] == apple\_y)) {

dots++;

locateApple();

}

}

//Moving snake joints up the chain.

for (int z = dots; z > 0; z--) {

x[z] = x[(z - 1)];

y[z] = y[(z - 1)];

}

//Moving snake head to the left.

if (leftDirection) {

x[0] -= DOT\_SIZE;

}

//If snake hits one of its joints with head the game is over.

for (int z = dots; z > 0; z--) {

if ((z > 4) && (x[0] == x[z]) && (y[0] == y[z])) {

inGame = false;

}

}

//Game is over if snake hits bottom of the board.

if (y[0] >= B\_HEIGHT) {

inGame = false;

}

b. Snake Subsystem:

//This is the main class along with the setResizable() method that affects the insets of JFrame

setResizable(false);

pack();