This is a program that replicates the game Bubble Trouble. There are 6 classes in the code. Namely these are Main, Environment, Ball, Arrow, Player and Bar classes.

The Bar class has a single purpose, to render the time bar. To do that it calculates the time that has passed from the beginning of the game, calculates the bar length accordingly and draws it at the bottom of the window. A bar object is created in the beginning of the game and methods are called for each frame.

The Player class has methods the move and render the player image on the window. The image just moves left and right. A player object is created in the beginning of the game and its methods are used at each frame.

The Arrow class has a method that checks if the arrow object has its tip touching to the ceiling. If it is, it deletes the arrow. If not, the arrow moves one step higher and rendered. The arrow object is created at the same x-coordinate with the player.

The Ball class has methods that moves the ball object, checks if it collides with the player, arrow, floor, or the walls. It also has a method that renders the ball. Moving and rendering methods are straightforward but the collision checking method is a bit complicated. Side wall and floor collisions are checked through position and radius combination easily. Arrow collision is checked by modeling the arrow as a line without width and the balls as spheres then Euclidean distance is used to check the collision. The player collision is checked through 4 different cases, the ball hitting the player hitbox from the top, from two sides separately and from the corners. Euclidean distance is also used here and the ball is modeled as a sphere in each case.

The Environment class has all the constants of the game inside. Everything happens on this class's constructor method. So, the game runs when an instance of this class is created. All the instances of above explained classes are created in the constructor method of this class and the methods of these classes are used together in a way that makes up the game. Ending conditions of the game are also checked here and when the end conditions are matched, the player is asked to quit the game or to play again. If the player chooses to quit the game, the game ends itself and the Java code stops working. Else, another Environment object is created recursively and inside the constructor method of that new object, the game starts to run again.

The Main class is where the initial instance of the Environment class is created. Its only purpose is this. The game actually runs when the instance of the Environment class is created in the Main class. The creation process of the object is the game itself.

Link to gameplay:
https://drive.google.com/file/d/174lhyueMm-jRSZhnlnu06xdfjxHsNqIf/view?usp=sharing