

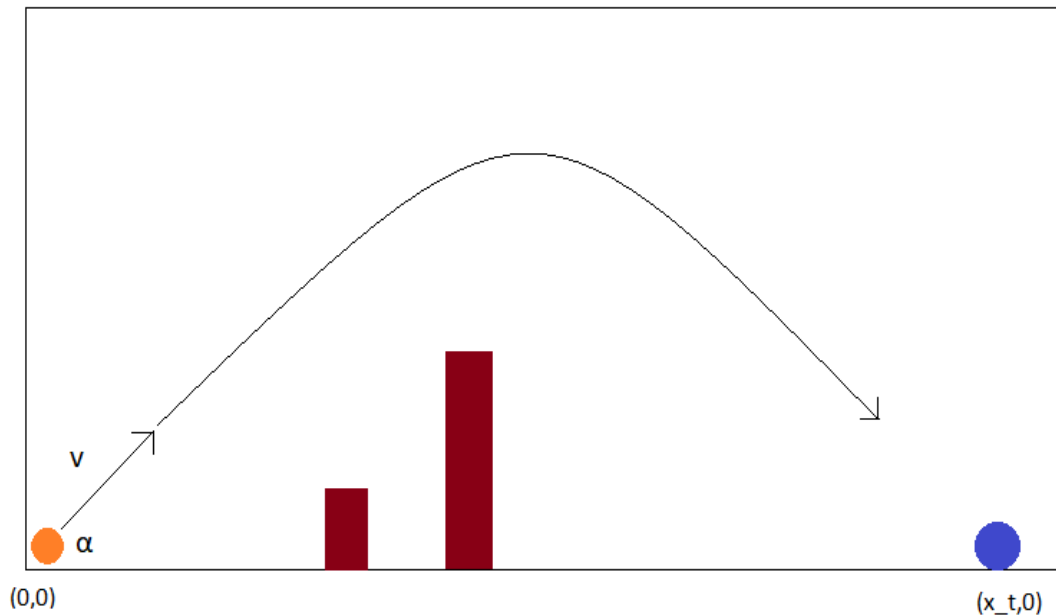
Projectile Motion (100 points)

Please read [important information](#) first.

Description

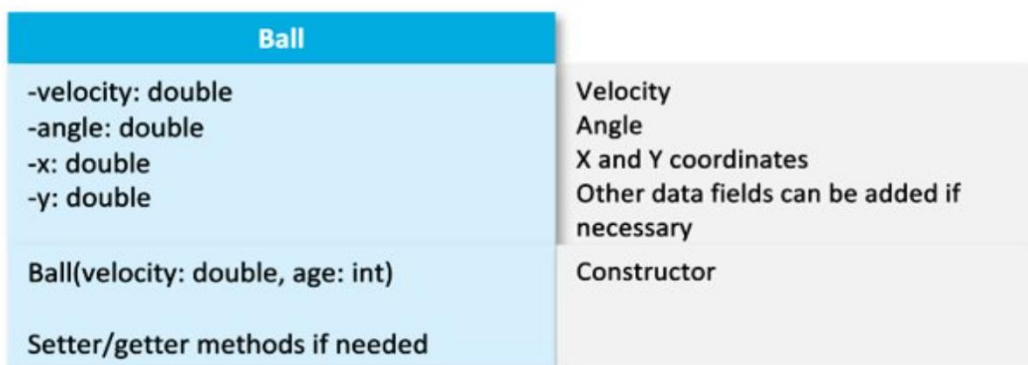
In this project you are going to simulate projectile motion.

We have a **Game Scene**, with a ball, a target and several obstacles. Our aim is to find the velocity for a given angle (shown as $\hat{\alpha}$), which we can hit the target with our ball without hitting any of the obstacles. Game scene is illustrated below:



You are required to complete the following classes:

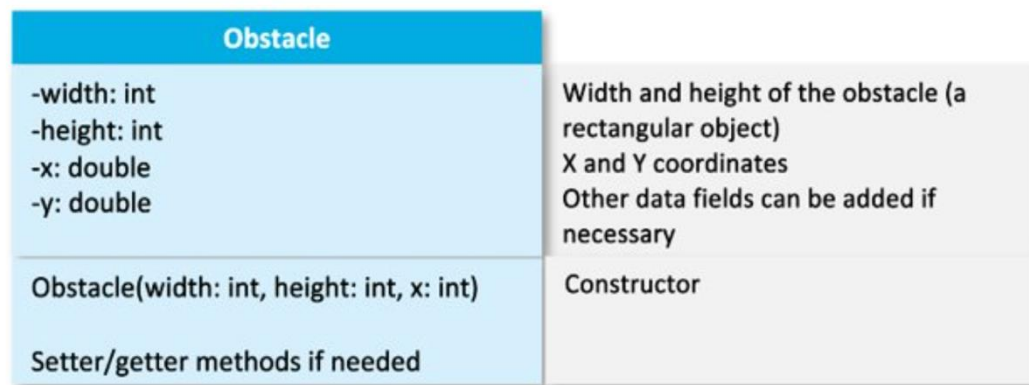
- **Ball:** Ball object will represent the orange ball in the game scene. We will calculate the appropriate velocity that will make the ball hit the target (blue ball). Complete the ball class using the following UML diagram. You can add extra datafields according to your



solution.

- **Obstacle:** Obstacles are rectangular obstacles that were represented with red boxes in the game scene. While we are calculating the appropriate velocity for a given $\hat{\alpha}$, we need to make sure our ball do not hit these obstacles. Obstacle objects will be represented in the game scene in an array list, and will be given in different locations on

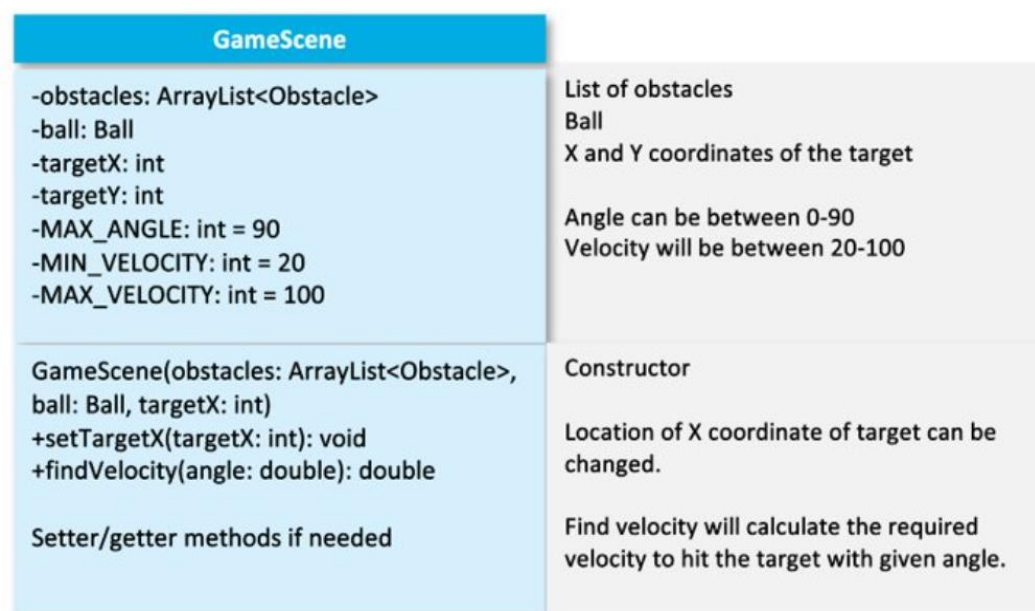
different test cases. Complete the obstacle class using the following UML diagram. You can add extra data fields



according to your solution.

• **GameScene:** Game scene will represent the whole scene, with the ball, the obstacles and location of the target. Complete this class according to the given UML diagram. **findVelocity** method will be used in test cases:

- \hat{I}_{\pm} will be given to this method as a parameter.
- You will calculate the appropriate velocity for the given \hat{I}_{\pm} , where the ball can hit the target. Velocity cannot be greater than the max velocity given in the class.
- Ball always will be thrown from (0,0)
- Obstacles and target will be on the ground, there will not be any flying obstacles or target.
- Obstacle and target information will be available on game scene (As a datafield, see UML diagram)
- findVelocity method should return "-1.0" if there is no solution. This can happen with impossible settings of obstacles.



When we throw the ball, the movement will be a projectile motion, therefore, you can do your calculations according to the given formulas:

Coordinates of the ball at time t :

$$x_t = x_0 + v_x t$$
$$y_t = y_0 + v_y t - \frac{1}{2} g t^2$$

t : time

x_0, y_0 : initial x and y coordinates

v : velocity

v_x, v_y : x and y components of v

α : angle

g : gravity of Earth (9.81)

Grading

Your code will be tested in the following manner:

- At first, a ball and several obstacles will be created. (Follow the UML diagrams for constructors.)
- A game scene will be created
- findVelocity method will be tested with several different \hat{I}_{\pm} values and different locations for the target.
- Make sure you complete all classes according to the UML diagrams. You can add extra methods and data fields.

Package classification

This is the package classification described in [important information](#).

- **Changeable**
 - question