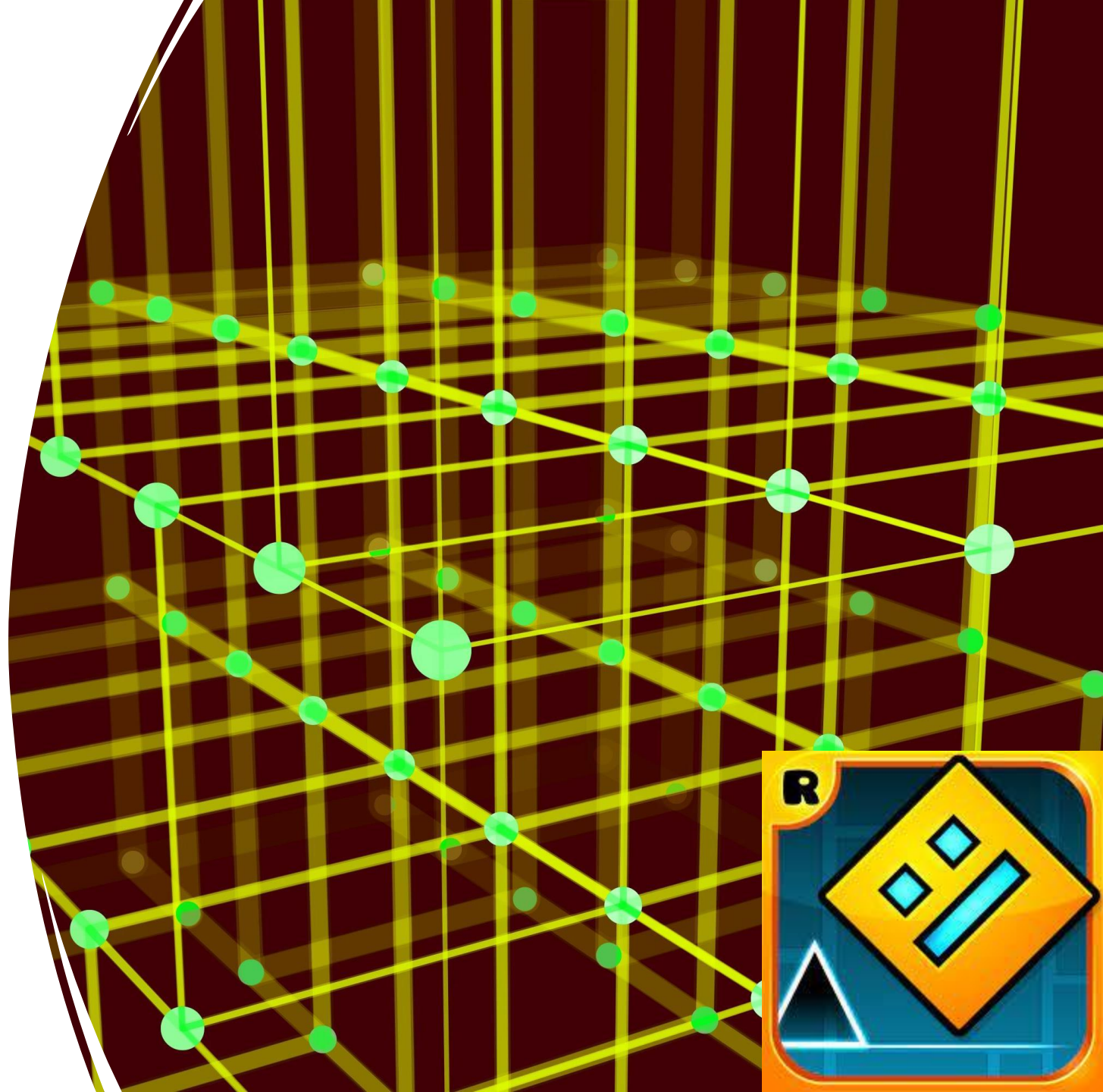


GEOMETRY DASH

By Nirmal and Dhruv



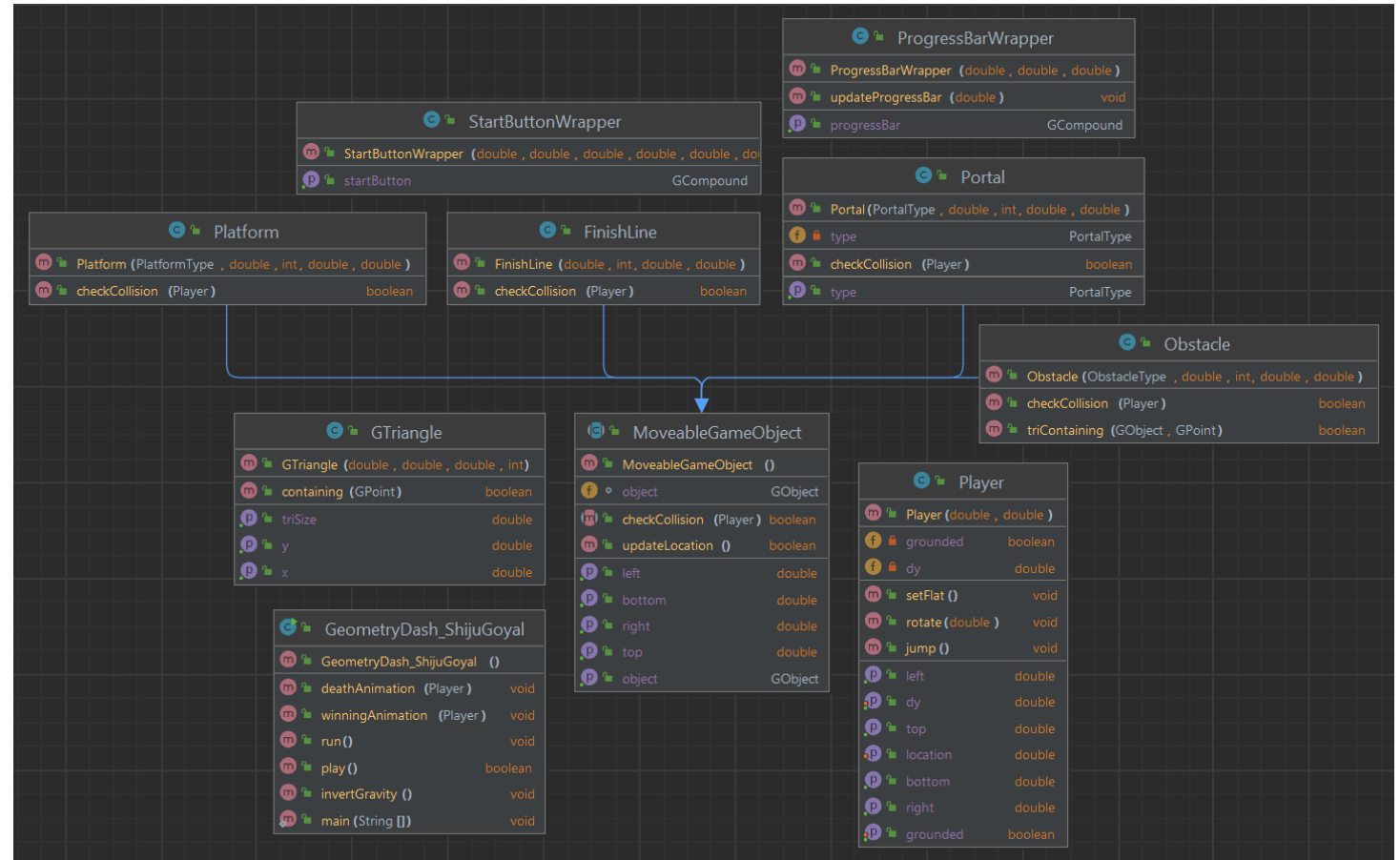
GOALS

- Create a realistic recreation of Geometry Dash
- Include accurate collisions with multiple shapes
- Replicate the smooth rotation and landings of the player
- Include portals to spice up the gameplay
- Make a game that is fun to play!



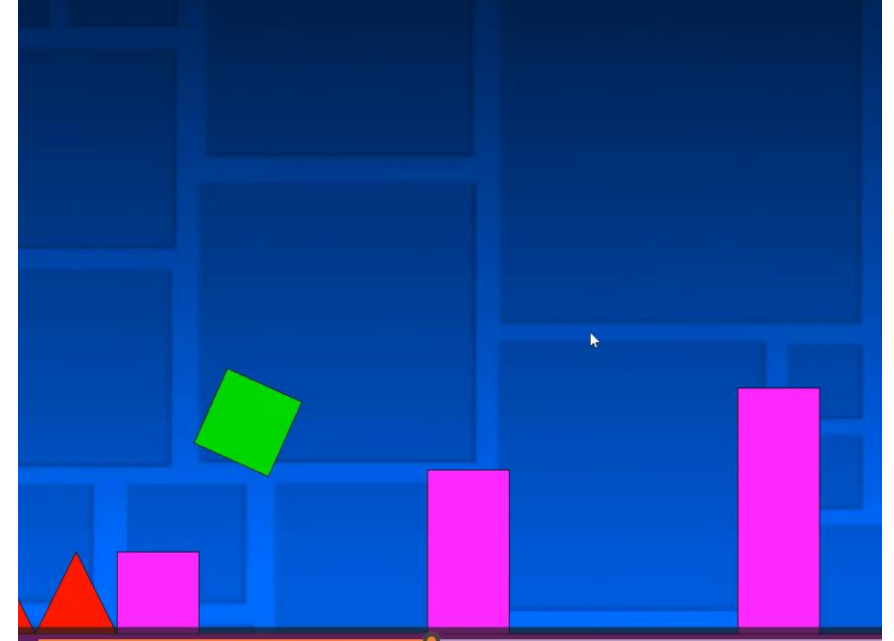
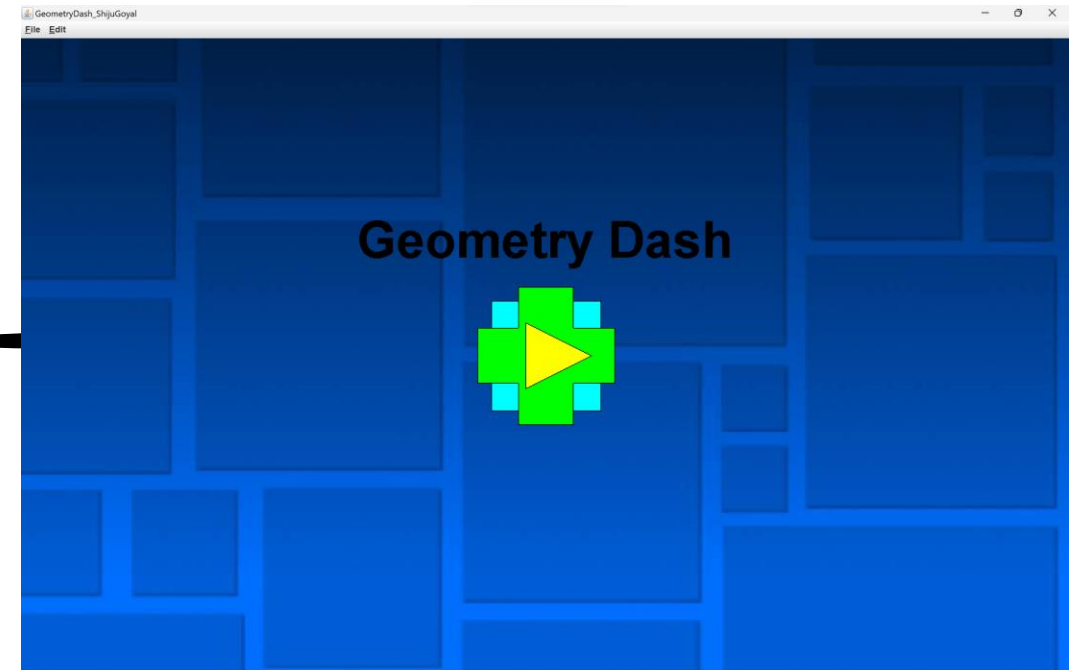
OUR DESIGN

- Utilized abstract class: MoveableObject
 - Platform, Obstacle, Portal, and FinishLine classes extend MoveableObject
- Utilized KeyListener interface
- Utilized static final ints, booleans, doubles, arrays
- Utilized ACM graphics
 - Player extends GPolygon
 - GTriangle extends GPolygon



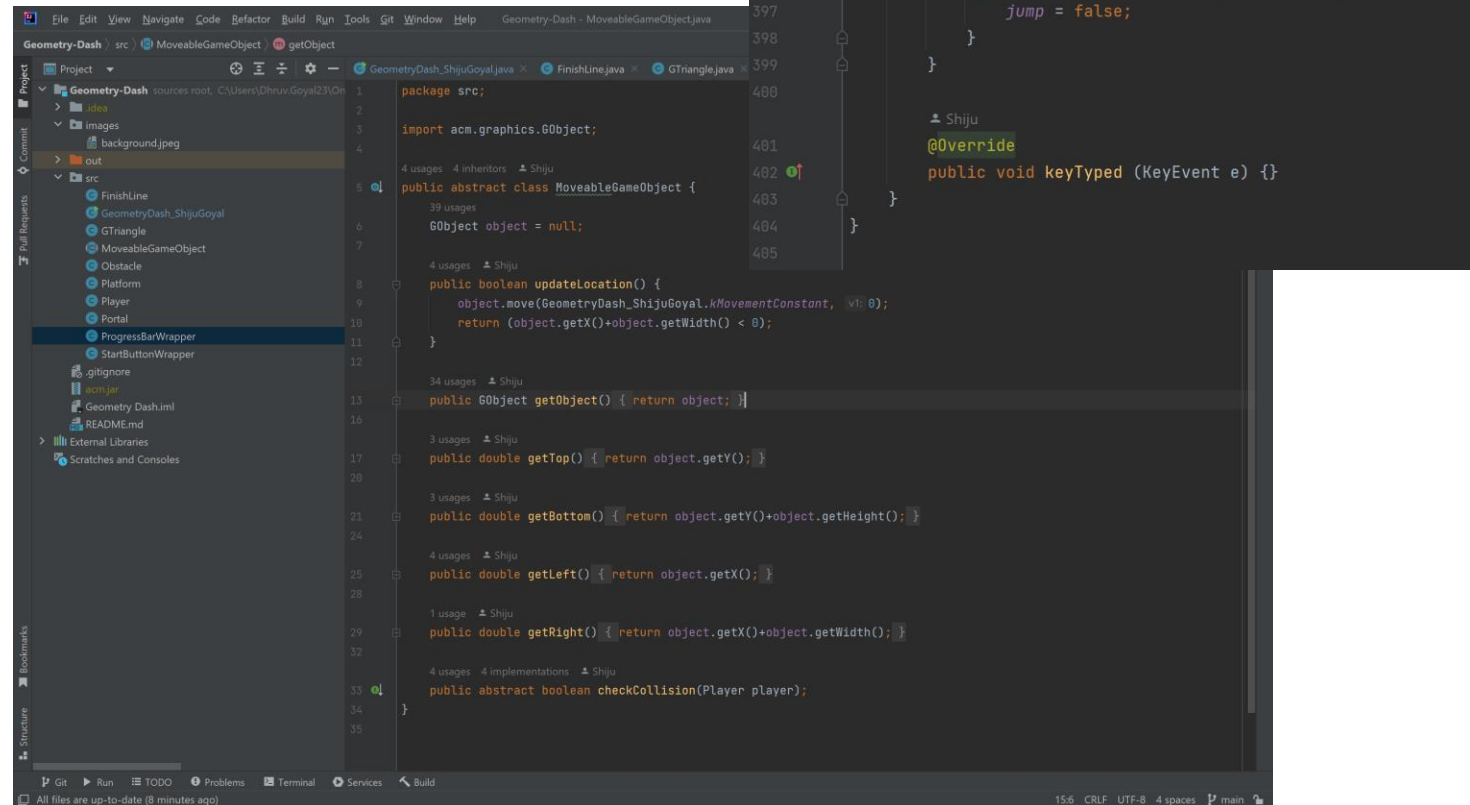
WHAT WE ACHIEVED

- Created game with same mechanics and feel as Geometry Dash
 - Start menu that mirrors real game
 - Similar background as real game
 - Player rotates perfectly while jumping
 - Have a progress bar at bottom that mirrors real game
 - Player explodes after collision with obstacles
 - Restarts level after failures
 - Includes pause button
- Working collision detection with different objects (triangles, rectangles, lines)
- Implemented an inverse gravity portal



WHAT WE LEARNED

- A practical application of an abstract class and interfaces
- The math behind collisions between various geometric shapes, such as rectangles hitting triangles
- Mastered ACM graphics
- How to create a feeling of smooth, aesthetically pleasing motion



MORE CODE SNIPPETS

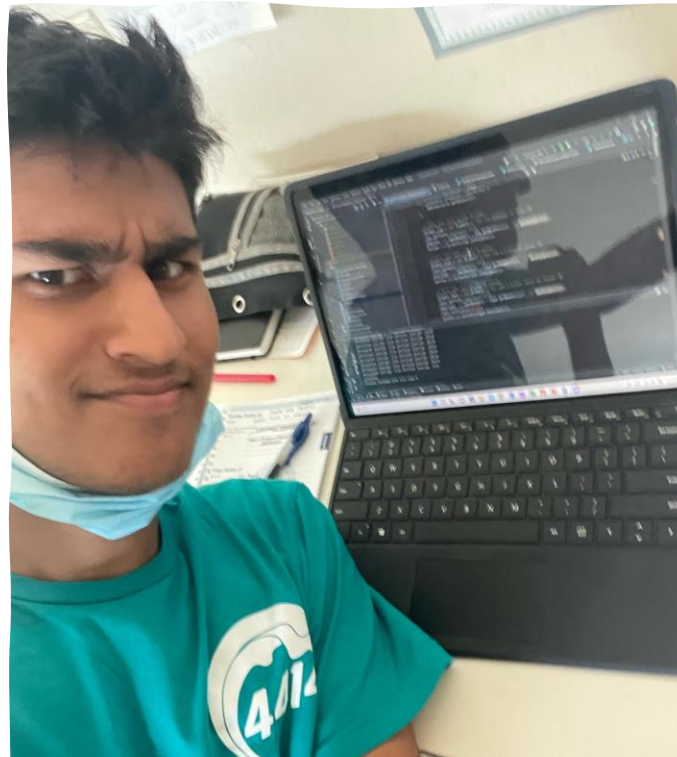
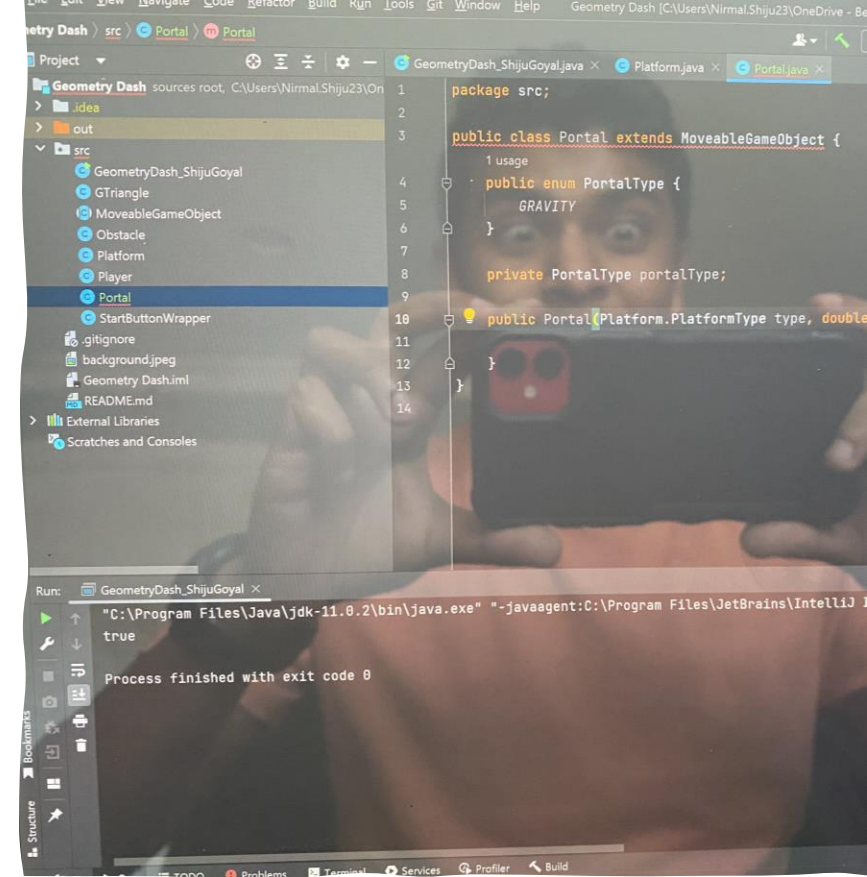
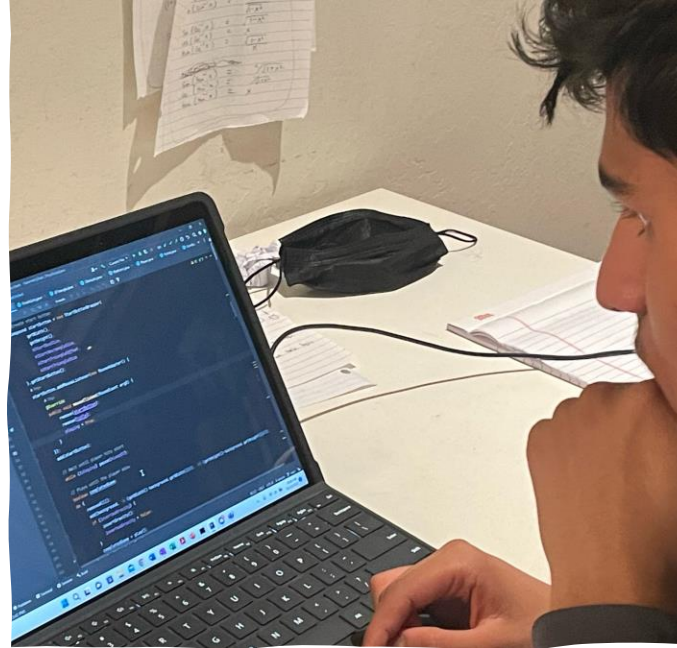
```
1 package src;
2
3 import ...
4
5 51 usages dhruvgoyaljc +1
6
7 public class Obstacle extends MoveableGameObject {
8     25 usages Shiju
9     public enum ObstacleType {
10         3 usages
11         RECTANGLE,
12         23 usages
13         TRIANGLE,
14         3 usages
15         LINE
16     }
17
18     3 usages
19     private ObstacleType type;
20
21     23 usages Shiju +1
22     public Obstacle(Obstacle.ObstacleType type, double xMultiplier, int yIndex, double param1, double param2) {
23         //for all types, xMultiplier specifies x-coord, & yIndex specifies y-coord using heights array
24
25         //assigning params to instance vars so they can be accessed by other methods in class
26         this.type = type;
27
28         //actual creation of obstacle object
29         switch (this.type) {
30             case RECTANGLE:
31                 // For rectangles, param1 is the horizontal length of the rectangle
32                 // and param 2 is the vertical length of the rectangle
33                 object = new GRect(
34                     v1: GeometryDash_ShijuGoyal.kLocationConstant*xMultiplier,
35                     GeometryDash_ShijuGoyal.kHeightLevels[yIndex],
36                     v2: GeometryDash_ShijuGoyal.kObjectSize*param1,
37                     v3: GeometryDash_ShijuGoyal.kObjectSize*param2
38                 );
39             case TRIANGLE:
40                 // For triangles, param1 is the horizontal length of the triangle
41                 // and param 2 is the vertical length of the triangle
42                 object = new GTriangle(
43                     v1: GeometryDash_ShijuGoyal.kLocationConstant*xMultiplier,
44                     GeometryDash_ShijuGoyal.kHeightLevels[yIndex],
45                     v2: GeometryDash_ShijuGoyal.kObjectSize*param1,
46                     v3: GeometryDash_ShijuGoyal.kObjectSize*param2
47                 );
48             case LINE:
49                 // For lines, param1 is the x-coord of the start point
50                 // and param 2 is the y-coord of the start point
51                 object = new GLine(
52                     v1: GeometryDash_ShijuGoyal.kLocationConstant*xMultiplier,
53                     GeometryDash_ShijuGoyal.kHeightLevels[yIndex],
54                     v2: GeometryDash_ShijuGoyal.kObjectSize*param1,
55                     v3: GeometryDash_ShijuGoyal.kObjectSize*param2
56                 );
57         }
58     }
59 }
```

```
Geometry-Dash - GTriangle.java
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
GeometryDash_ShijuGoyal.java x FinishLine.java x GTriangle.java x MoveableGameObject.java x Obstacle.java x ProgressBarWrapper.java x
Project
  Geometry-Dash
    src
      FinishLine
      GeometryDash_ShijuGoyal
      GTriangle
      MoveableGameObject
      Obstacle
      Platform
      Player
      Portal
      ProgressBarWrapper
      StartButtonWrapper
    .gitignore
    acm.jar
    GeometryDash.html
    README.md
  External Libraries
  Scratches and Consoles
  Pull Requests
  Commit
  Structure
  Bookmarks
  Notifications
  7 1
  20
  21
  22 @Override
  23 public double getX() { return super.getX(); }
  24
  25
  26
  27 @Override
  28 public double getY() { return super.getY(); }
  29
  30
  31 @Override
  32 public boolean containing(GPoint point) {
  33     double x0 = getX();
  34     double y0 = getY();
  35     if (point.getX() > y0 || point.getY() < (y0 + size)) {
  36         return false;
  37     }
  38     //If code reaches here, then GPoint y-coord is within triangle
  39     //vertical bounding box
  40     double distance = ((y0 + size) - point.getY())/2;
  41     if (point.getX() >= x0 - distance &&
  42         point.getX() <= x0 + distance) {
  43         return true;
  44     }
  45     return false;
  46 }
  47
  48 //Override
  49 dhruvgoyaljc +1
  50 public double getTriSize() { return size; }
  51
  52
  40/45 CRLF UTF-8 4 spaces P main
```

PROGRAM IN ACTION

<https://youtu.be/wQwpGNxKHP4>

PICTURES OF US STUDIOUSLY CODING



THANK YOU!

