```java
package program;

import java.util.ArrayList;

public class Hero implements IAnimatable, IEntity{
    private Animation idleAnimation;
    private Animation runLeftAnimation;
    private Animation runRightAnimation;
    private Point position;
    private DungeonWorld level;


    public Hero() {
        ArrayList<Texture> idle = new ArrayList<>();
        idle.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_idle_1.png"));
        idle.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero.png"));
        idle.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_idle_2.png"));
        idle.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero.png"));
        idle.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_idle_1.png"));
        idleAnimation = new Animation(idle,8);

        ArrayList<Texture> runLeft = new ArrayList<>();
        runLeft.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_run_left_1.png"));
        runLeft.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_run_left_2.png"));
        runLeftAnimation = new Animation(runLeft, 8);

        ArrayList<Texture> runRight = new ArrayList<>();
        runRight.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_run_right_1.png"));
        runRight.add(new Texture("C:\\Users\\oooo\\eclipse-workspace\\PM-Dungeon\\assets\\textures\\characters\\hero_run_right_2.png"));
        runRightAnimation = new Animation(runRight, 8);
    }

    public void setLevel(DungeonWorld level) {
        this.level = level;
        findRandomPostion();
    }

    public void findRandomPostion(){
        this.position = new Point(level.getRandomPointInDungeon());
    }

    @Override
    public Point getPosition() {
        return this.position;
    }

    @Override
    public Animation getActiveAnimation() {
```

```
59          if(Gdx.input.isKeyPressed(Input.Keys.D) || Gdx.input.isKeyPressed(Input.Keys.W)){
60              return runRightAnimation;
61          }else if(Gdx.input.isKeyPressed(Input.Keys.A) || Gdx.input.isKeyPressed(Input.Keys.S)){
62              return runLeftAnimation;
63          }else {
64              return idleAnimation;
65          }
66      }
67
68      @Override
69      public boolean deleteable() {
70          // TODO Auto-generated method stub
71          return false;
72      }
73
74      @Override
75      public void update() {
76          Point newPosition = new Point(this.position);
77          float movementSpeed = 0.1f;
78          if(Gdx.input.isKeyPressed(Input.Keys.W))
79              newPosition.y += movementSpeed;
80          if(Gdx.input.isKeyPressed(Input.Keys.S))
81              newPosition.y -= movementSpeed;
82          if(Gdx.input.isKeyPressed(Input.Keys.D))
83              newPosition.x += movementSpeed;
84          if(Gdx.input.isKeyPressed(Input.Keys.A))
85              newPosition.x -= movementSpeed;
86          if(level.isTileAccessible(newPosition))
87              this.position = newPosition;
88          this.draw();
89      }
90
91  }
92
```