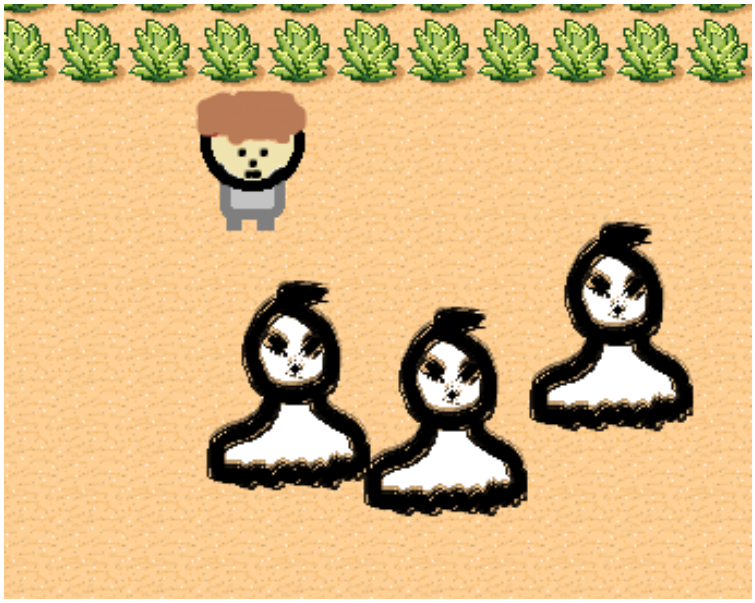
"Doodle Siege" – Simple 2D Shooter Game with Java

https://github.com/olehsau/doodleSiege



Abstract

"Doodle Siege" is a 2D shooter game developed as a university project using the libGDX framework in Java. This project showcases fundamental game development concepts and programming skills in a concise and entertaining package.

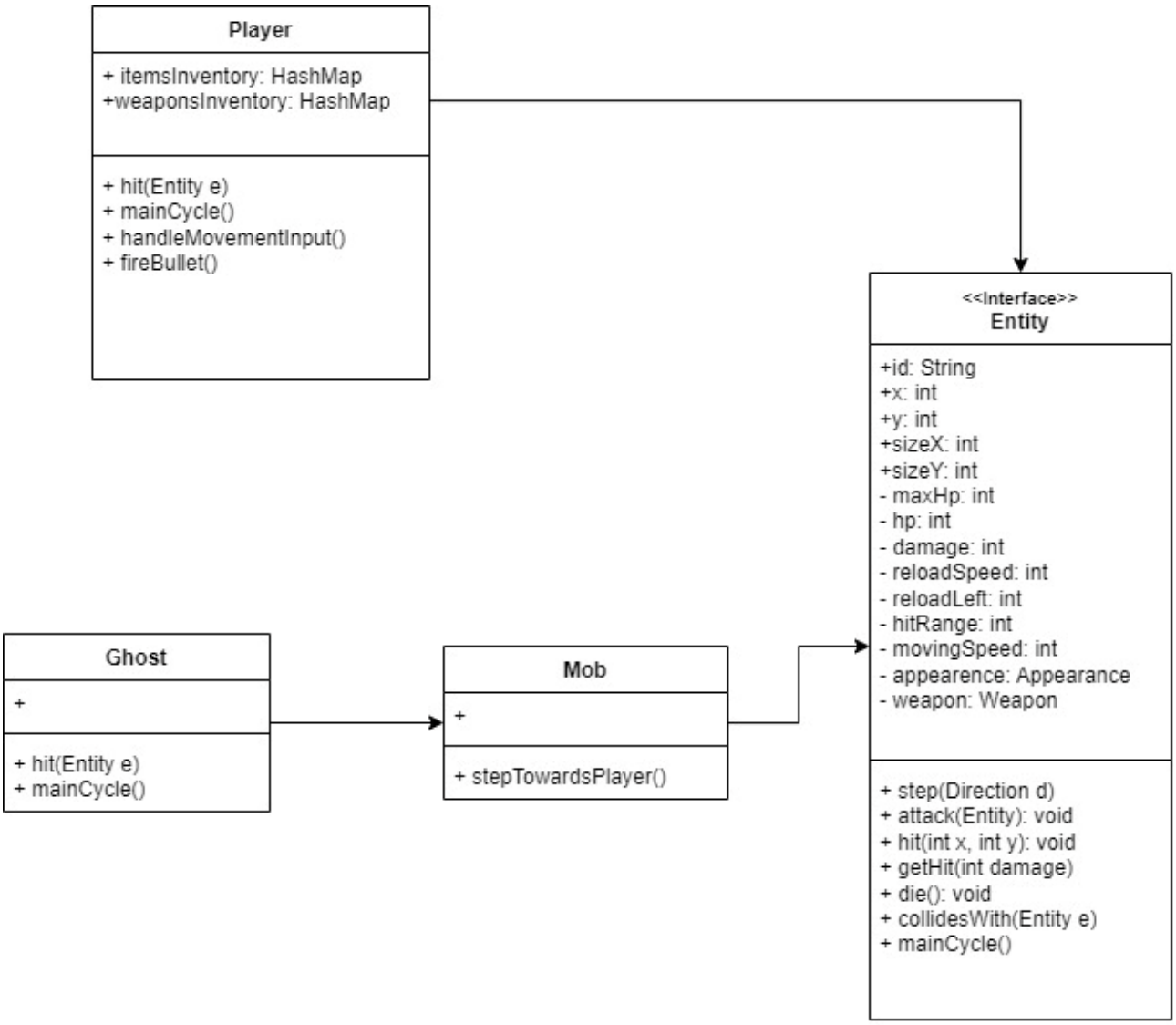
Key Features

1. Intuitive Controls
2. Healthbar
3. Kill Counter
4. Displaying AI generated game title screen at the start

Technical Details

Entity interface and its implementations

The most extensive class system – implements Entity interface. It contains classes Mob, Ghost and Player, which contain lots of entities methods and values. You can see the UML scheme below (not everything is actually implemented yet)



DoodleSiege class

The main class, `DoodleSiege`, which extends the ApplicationAdapter (defined by libGDX), is the main part of the project. This class overrides *create(), render(), and dispose()* methods, which are executed once when the application is created, every time rendering should be performed, and when the application is destroyed accordingly.

```
package com.mygdx.doodlesiege;
//
// *all imports*
//

public class DoodleSiege extends ApplicationAdapter {
    MapManager mapManager;
    ShapeRenderer shapeRenderer;
    Texture blank;
    MobsManager mobsManager;
    MyInputProcessor inputProcessor;
    BulletsManager bulletsManager;
    BitmapFont font;
    BitmapFont fontTitle;
    String AITitle;
    float timePlayed;

    @Override
    public void create () {
        batch = new SpriteBatch();
        batchFixed = new SpriteBatch();
        mapManager = MapManager.getInstance();
        cameraManager = CameraManager.getInstance();
        mobsManager = MobsManager.getInstance(mapManager);
        player = new Player("player",0,0,32,34, 100, 100, 10,0.5f, 40, 5, "player.png",
null);
        shapeRenderer = new ShapeRenderer();
        blank = new Texture("blank.png");
        inputProcessor = new MyInputProcessor();
        Gdx.input.setInputProcessor(inputProcessor);
        bulletsManager = new BulletsManager(mobsManager, mapManager);
        FreeTypeFontGenerator generator = new
FreeTypeFontGenerator(Gdx.files.internal("GAME OVER.TTF"));
        FreeTypeFontGenerator.FreeTypeFontParameter parameter = new
FreeTypeFontGenerator.FreeTypeFontParameter();
        parameter.size = 36;
        font = generator.generateFont(parameter);
        generator = new FreeTypeFontGenerator(Gdx.files.internal("Roboto-Regular.ttf"));
        parameter = new FreeTypeFontGenerator.FreeTypeFontParameter();
        parameter.size = 56;
        fontTitle = generator.generateFont(parameter);
        generator.dispose();
        AITitle = prompt("story of the hero fighting with ghosts");
        timePlayed = 0;
    }

    @Override
    public void render () {
        ScreenUtils.clear(0.5f, 0.5f, 0.5f, 1);
        batch.begin();
        mapManager.render(cameraManager.getCamera());
        player.mainCycle();
        cameraManager.update(player);
        mobsManager.mainCycle();
        batch.end();

        batchFixed.begin();
        batchFixed.setColor(Color.GRAY);
```

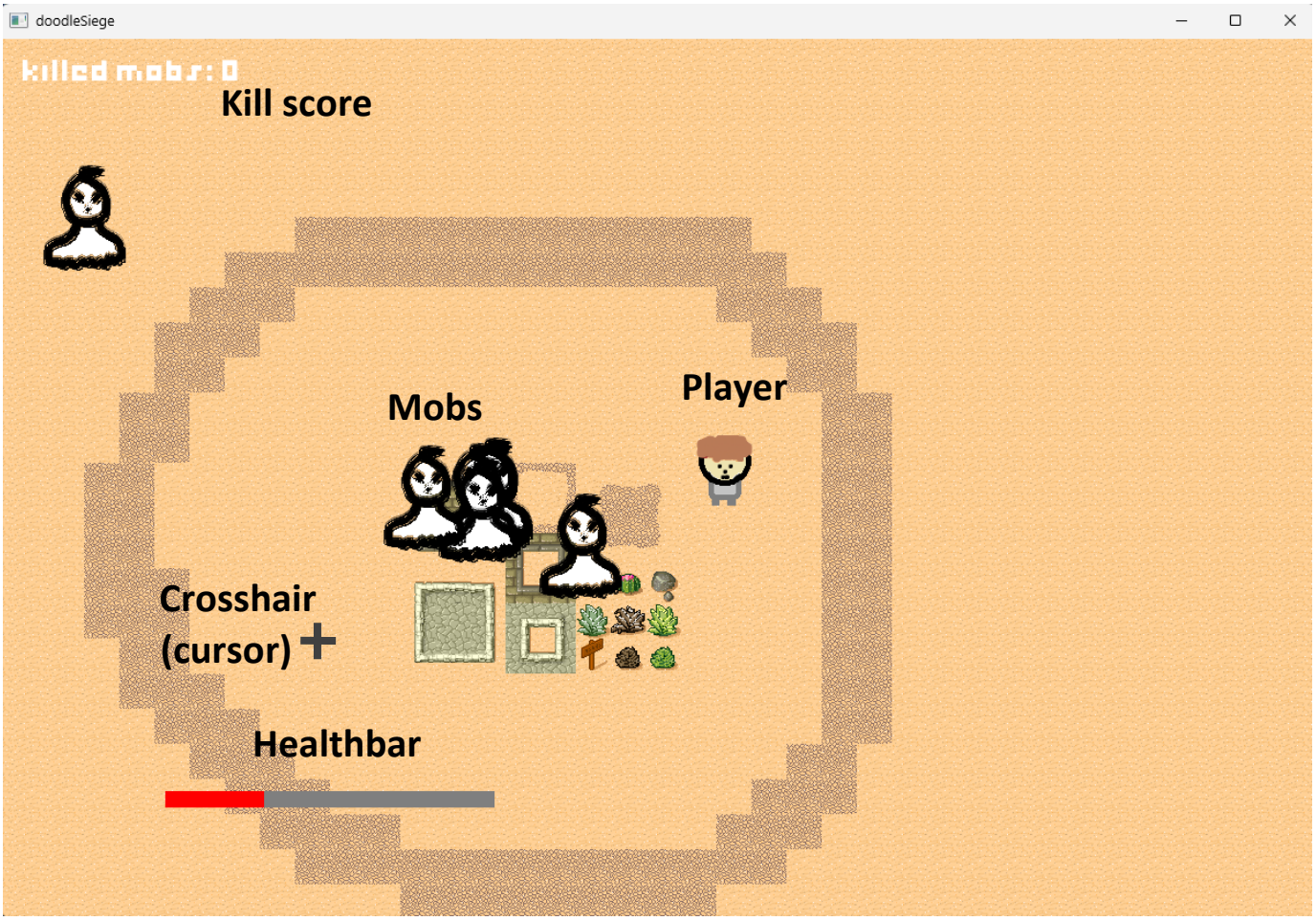
```
        batchFixed.draw(blank,150,100,300,15); // health bar
        batchFixed.setColor(Color.RED);

batchFixed.draw(blank,150,100,Math.round(((float)player.hp/(float)player.maxHp)*300),15);
        batchFixed.setColor(Color.WHITE);
        font.draw(batchFixed, "killed mobs: " + player.kills, 20, Gdx.graphics.getHeight()
- 20);
        if(timePlayed <= 6.0) {fontTitle.draw(batchFixed, AITitle, 50,
Gdx.graphics.getHeight() - 200);}
        batchFixed.end();

        shapeRenderer.setProjectionMatrix(cameraManager.getCamera().combined);
        shapeRenderer.begin(ShapeRenderer.ShapeType.Filled);
        shapeRenderer.setColor(Color.BLACK);
        bulletsManager.mainCycle();
        for(Bullet bullet : player.firedBullets){
           shapeRenderer.circle(bullet.x, bullet.y, 5.0f, 32);
        }
        shapeRenderer.end();
        timePlayed += Gdx.graphics.getDeltaTime();
    }

    @Override
    public void dispose () {
        batch.dispose();
        mobsManager.dispose();
        mapManager.dispose();
        mobsManager.dispose();
    }
}
```

AITextGenerator class

A useless but cool feature is displaying AI generated title screen text at the start. It is realized with method String prompt(String input), which essentially does HTTP request to ContentAI.net at rapidapi.com, which asks to create title on topic “story of the hero fighting with ghosts”:

```
HttpRequest request = HttpRequest.newBuilder()
        .uri(URI.create("https://contentai-net-text-
generation.p.rapidapi.com/v1/text/blog-articles?category="+input))
        .header("X-RapidAPI-Key", "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
        .header("X-RapidAPI-Host", "contentai-net-text-generation.p.rapidapi.com")
        .method("GET", HttpRequest.BodyPublishers.noBody())
        .build();
HttpResponse<String> response = HttpClient.newHttpClient().send(request,
HttpResponse.BodyHandlers.ofString());
```

Gameplay



The gameplay involves moving on static map, using WASD on desktop (other platforms don't support moving), and shoot evil mobs (currently only ghosts), which are following player, and damage him on impact. The rate of mobs spawning increases continuously.

The goal of the game is too kill as many mobs, using your weapon, as possible, before dying.

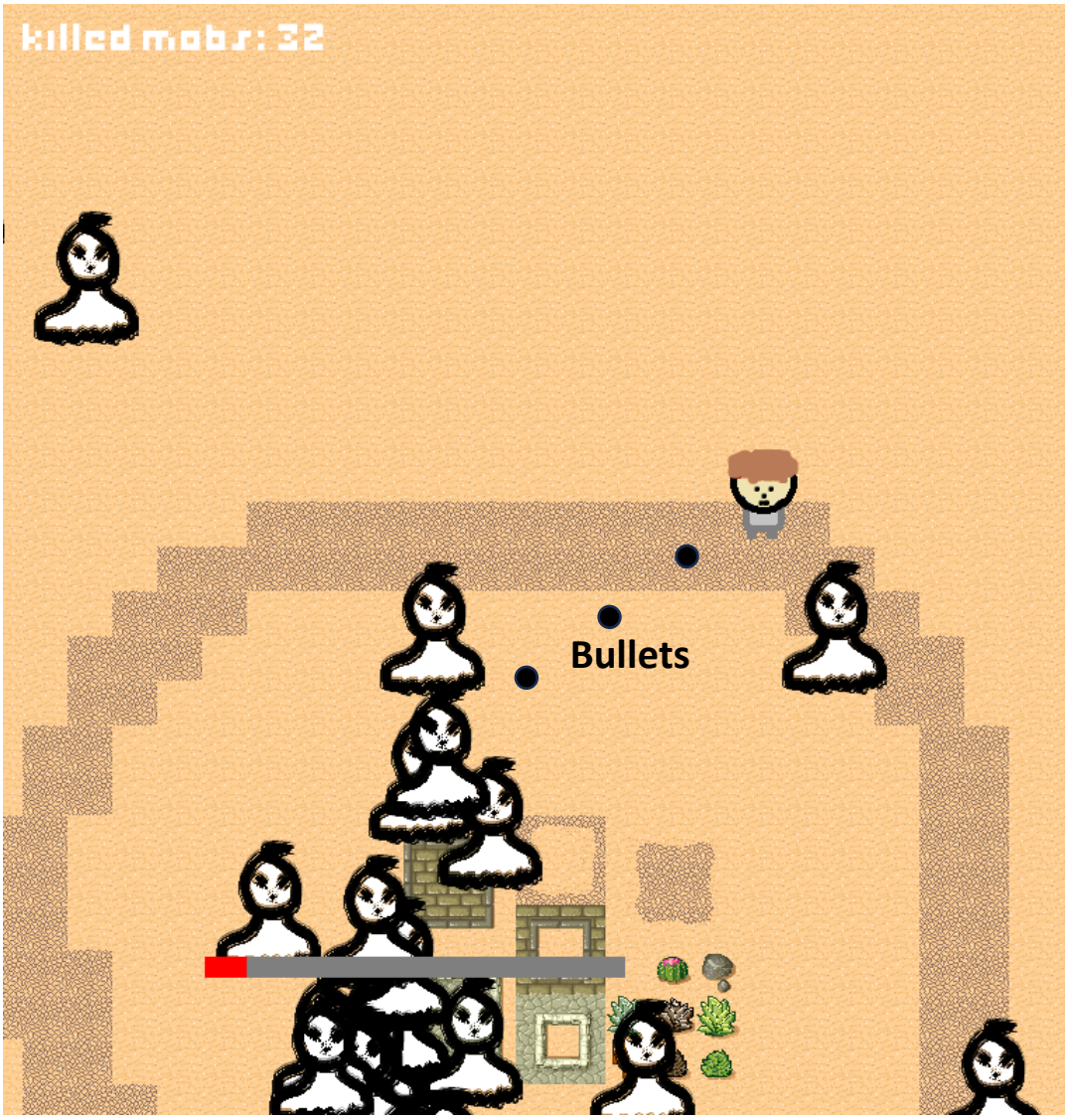


Figure 6. Death