The Development of a Role-Playing Game for The Android Operating System

Brandon, Skye, Chris, David

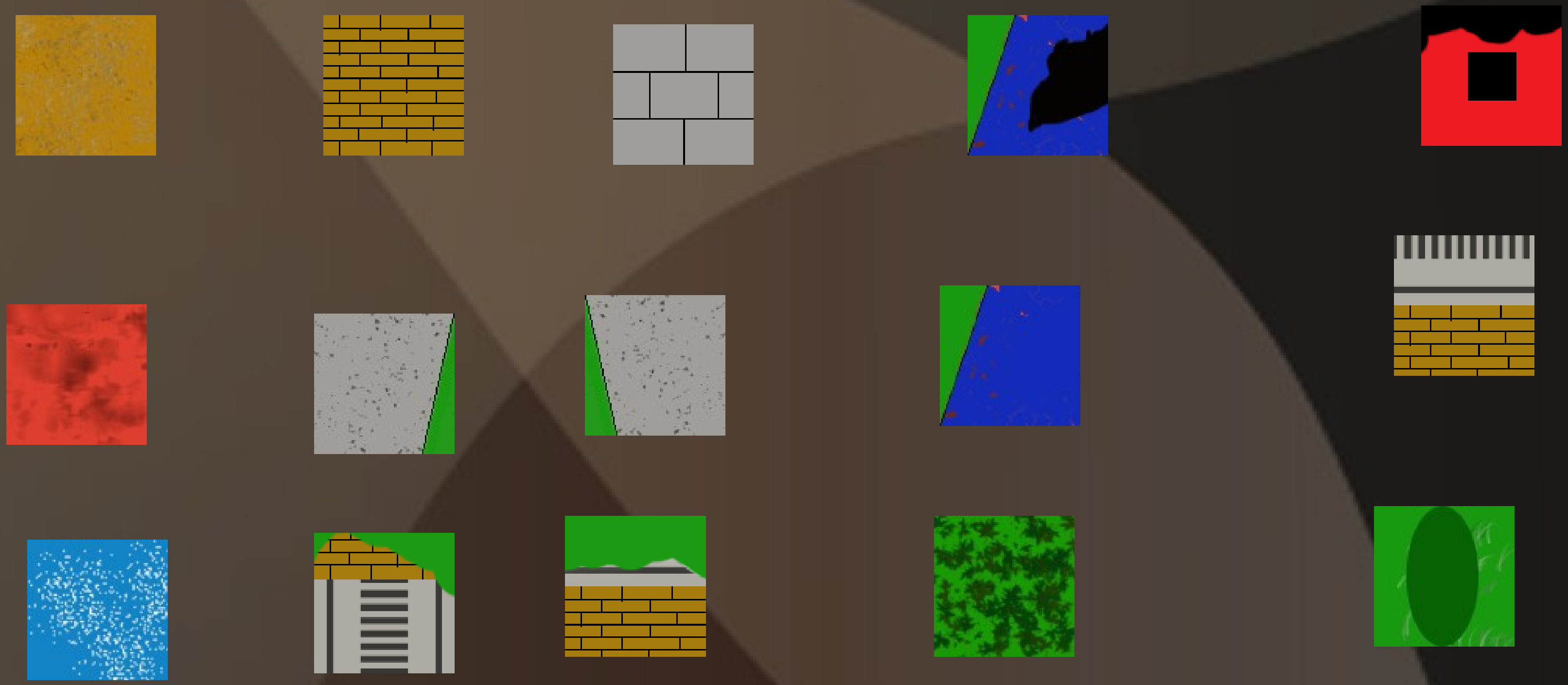
Game Overview

- Dungeon crawler
- Main and side quests
- Hero will level up, gain new abilities and get access to better weapons to defeat his enemies.
- New items will become available to the hero after completing quests.
- Main quest will be saving the town's people that were captured by the antagonist.

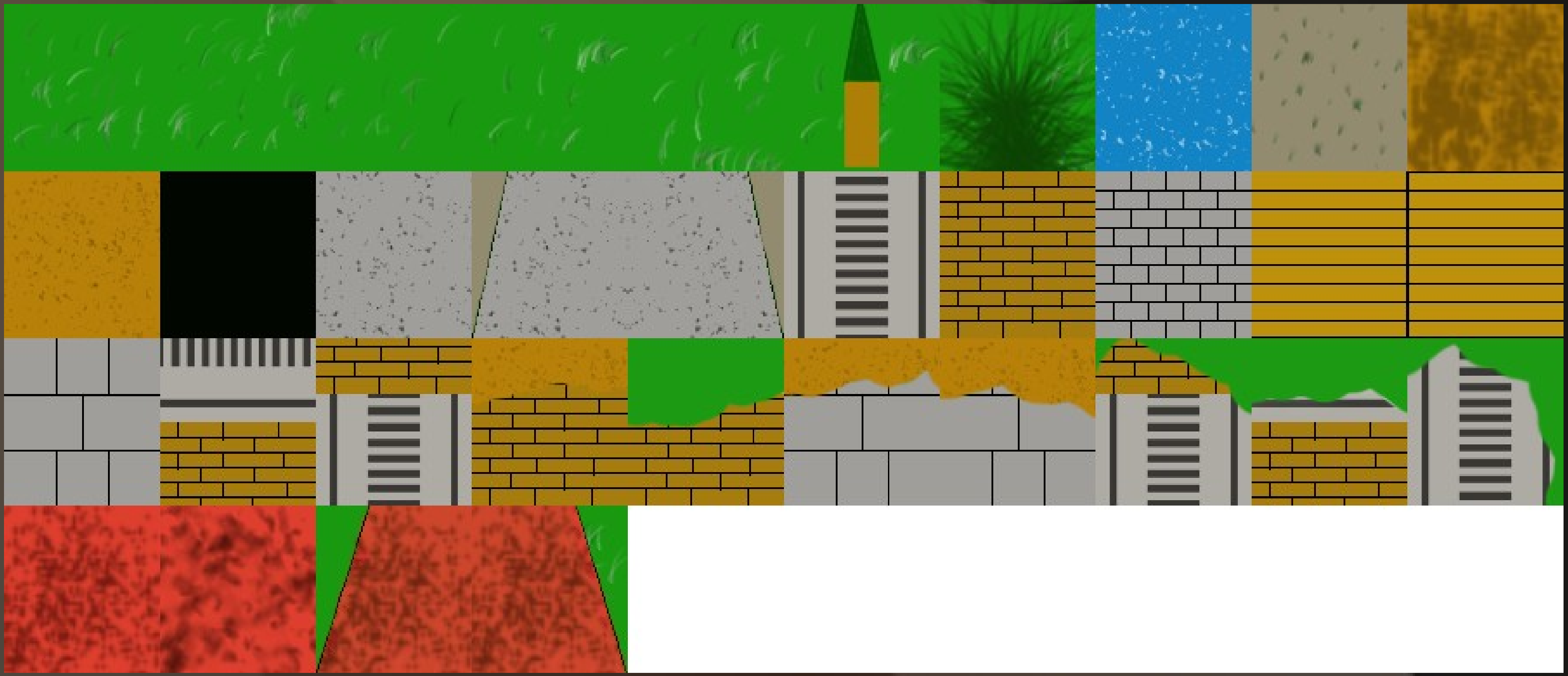
Main Map Overview

- Free roam area
- Shops and town
- Rescued towns people
- Rebuilding of city
- Entrance to dungeon

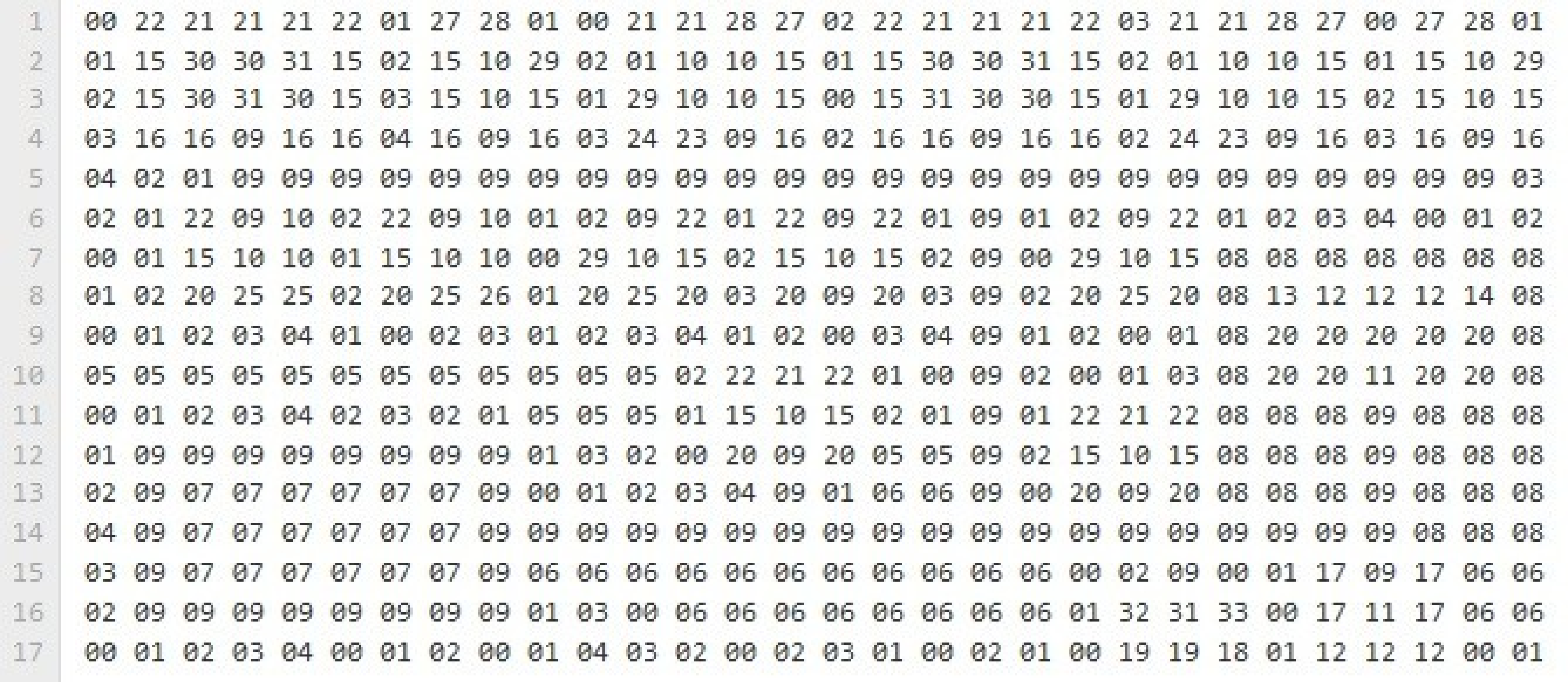
Sprites



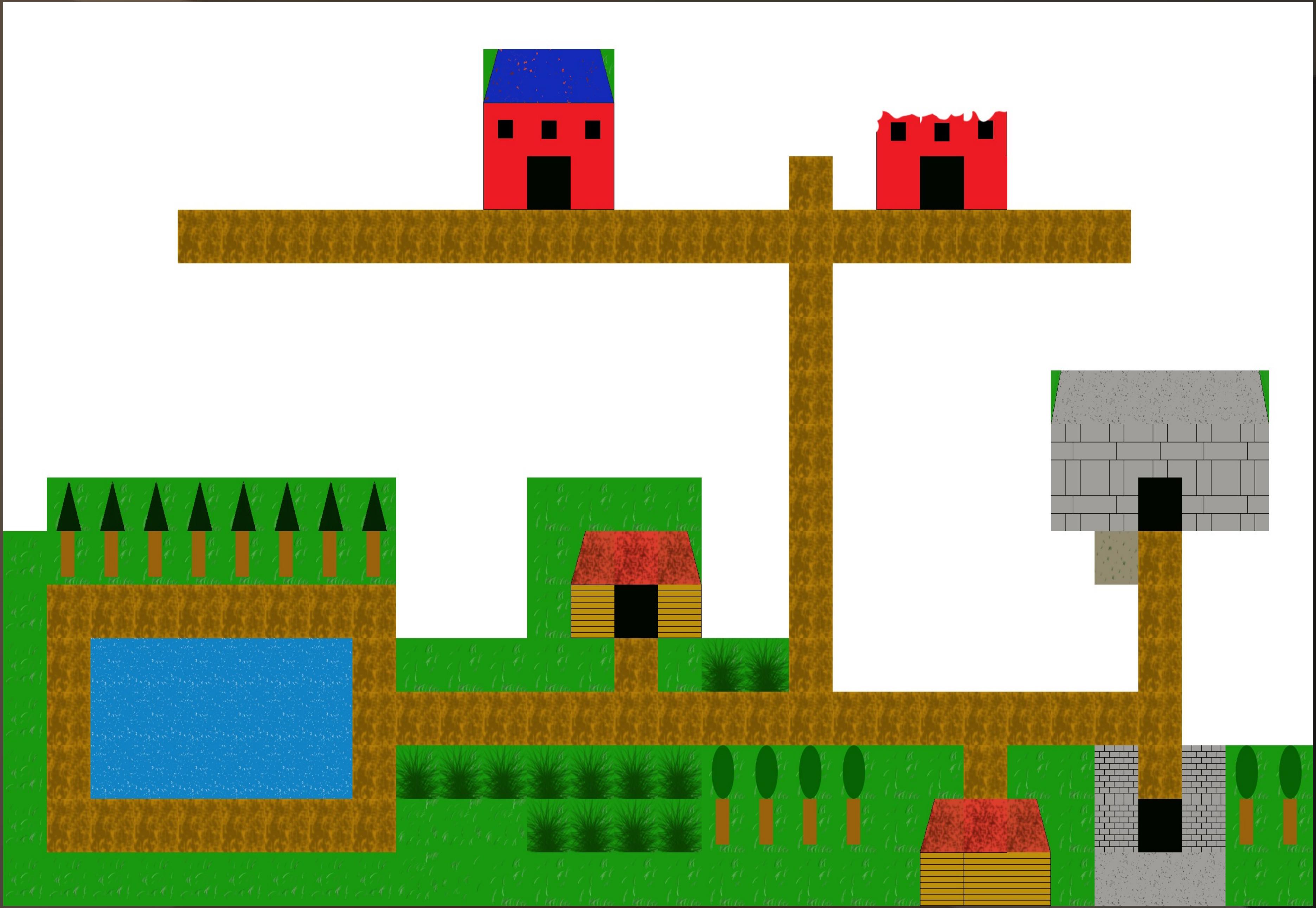
Sprite Sheet

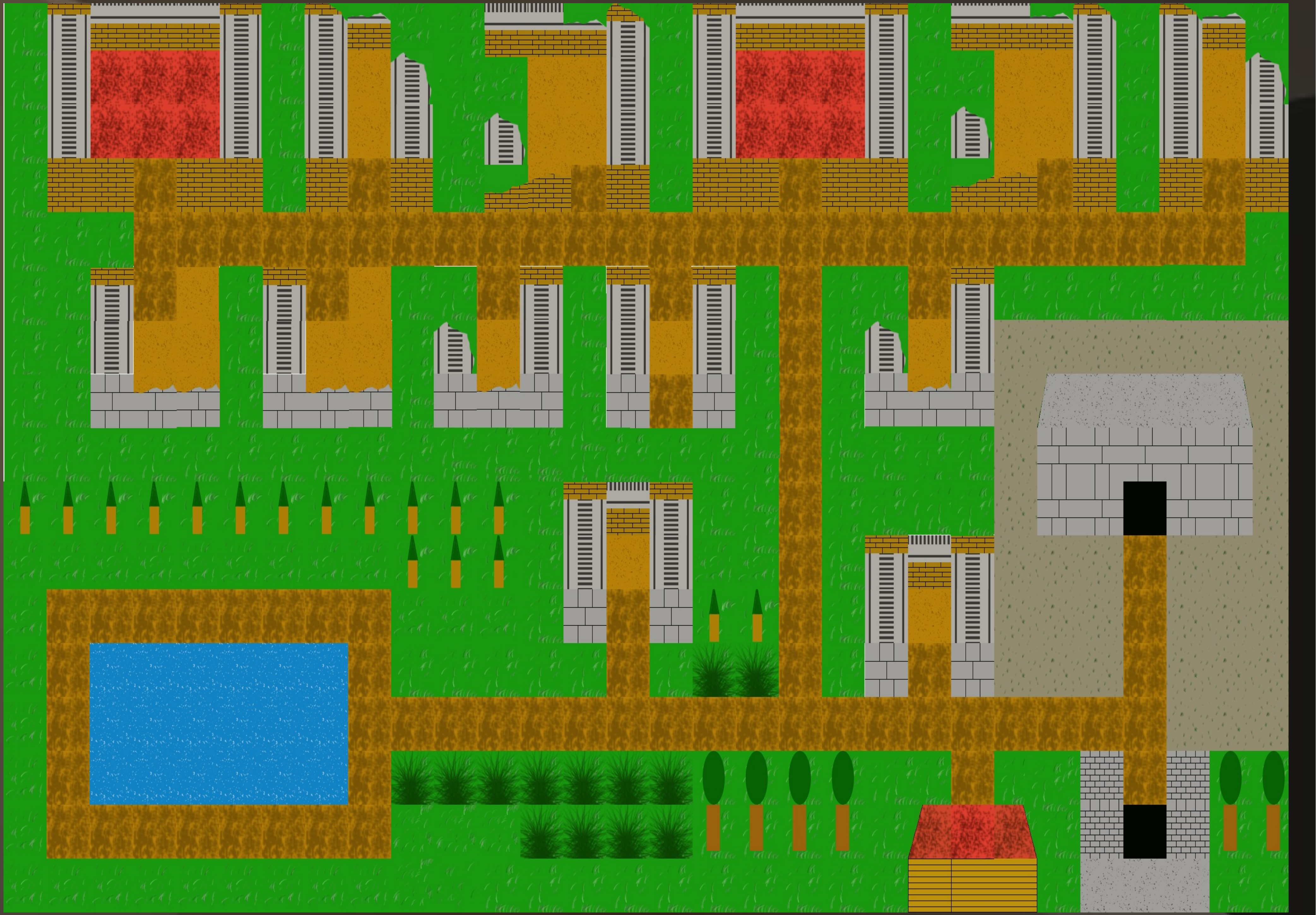


Sprite Coding



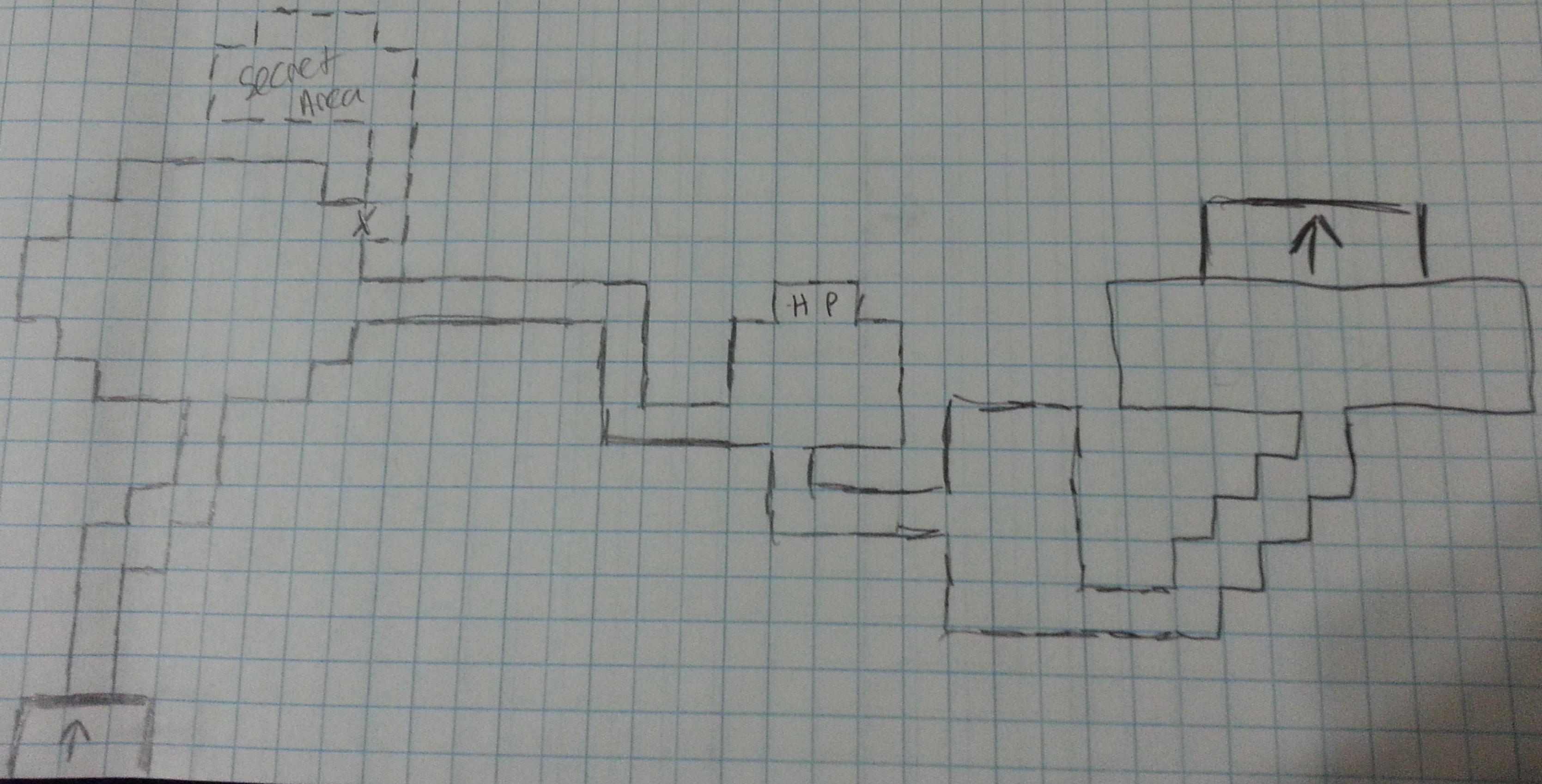
```
00 22 21 21 21 22 01 27 28 01 00 21 21 28 27 02 22 21 21 21 22 03 21 21 28 27 00 27 28 01
01 15 30 30 31 15 02 15 10 29 02 01 10 10 15 01 15 30 30 31 15 02 01 10 10 15 01 15 10 29
02 15 30 31 30 15 03 15 10 15 01 29 10 10 15 00 15 31 30 30 15 01 29 10 10 15 02 15 10 15
03 16 16 09 16 16 04 16 09 16 03 24 23 09 16 02 16 16 09 16 16 02 24 23 09 16 03 16 09 16
04 02 01 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 03
02 01 22 09 10 02 22 09 10 01 02 09 22 01 22 09 22 01 09 01 02 09 22 01 02 03 04 00 01 02
00 01 15 10 10 01 15 10 10 00 29 10 15 02 15 10 15 02 09 00 29 10 15 08 08 08 08 08 08 08
01 02 20 25 25 02 20 25 26 01 20 25 20 03 20 09 20 03 09 02 20 25 20 08 13 12 12 12 14 08
00 01 02 03 04 01 00 02 03 01 02 03 04 01 02 00 03 04 09 01 02 00 01 08 20 20 20 20 20 08
05 05 05 05 05 05 05 05 05 05 05 05 02 22 21 22 01 00 09 02 00 01 03 08 20 20 11 20 20 08
00 01 02 03 04 02 03 02 01 05 05 05 01 15 10 15 02 01 09 01 22 21 22 08 08 08 09 08 08 08
01 09 09 09 09 09 09 09 09 01 03 02 00 20 09 20 05 05 09 02 15 10 15 08 08 08 09 08 08 08
02 09 07 07 07 07 07 07 09 00 01 02 03 04 09 01 06 06 09 00 20 09 20 08 08 08 09 08 08 08
04 09 07 07 07 07 07 07 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 09 08 08 08
03 09 07 07 07 07 07 07 09 06 06 06 06 06 06 06 06 06 06 06 00 02 09 00 01 17 09 17 06 06
02 09 09 09 09 09 09 09 09 01 03 00 06 06 06 06 06 06 06 06 01 32 31 33 00 17 11 17 06 06
00 01 02 03 04 00 01 02 00 01 04 03 02 00 02 03 01 00 02 01 00 19 19 18 01 12 12 12 00 01
```





Dungeon

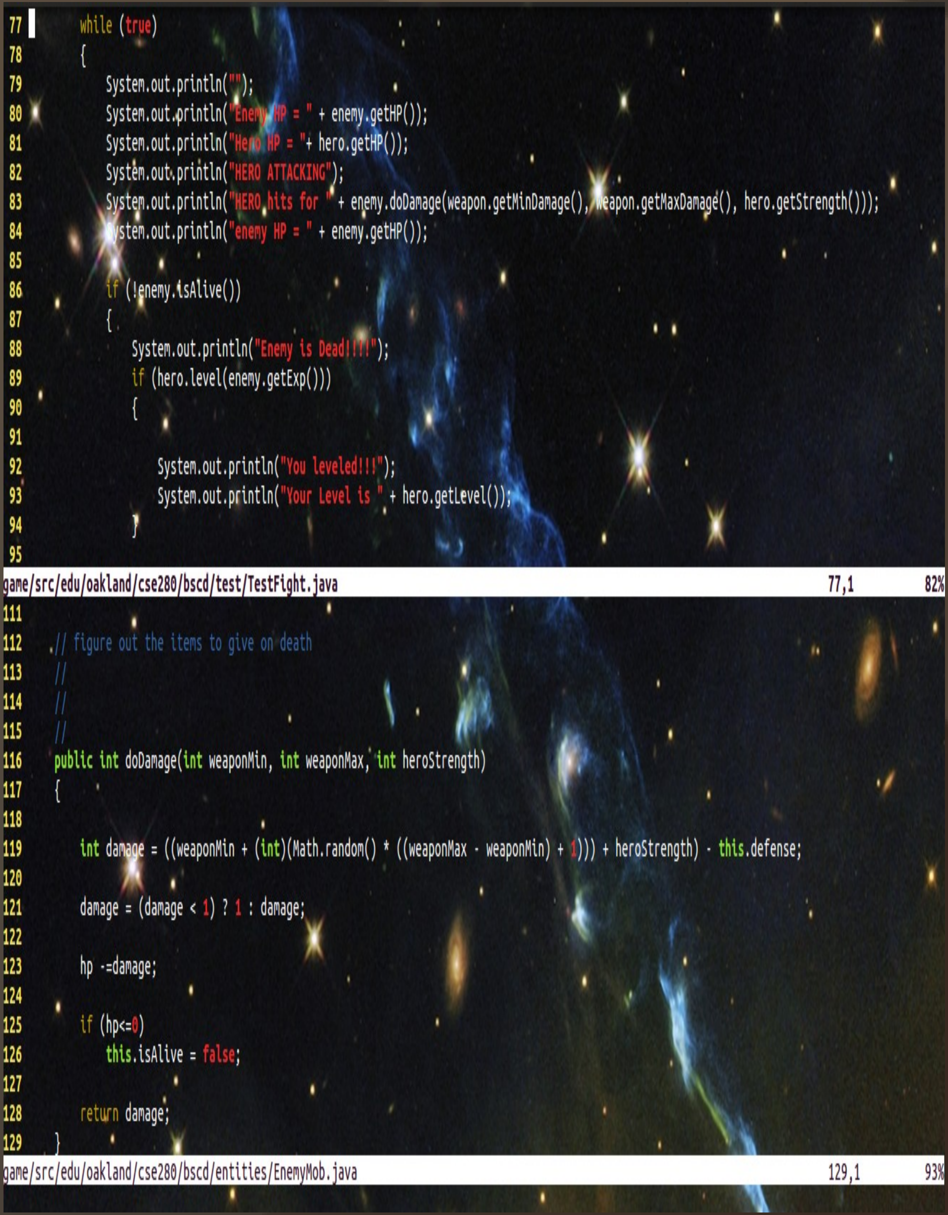
- Level-based
- Enemies (random encounter)
- Items in chests
- Bosses and mini-bosses
- Towns-person rescue


Secret
Area
X
HP

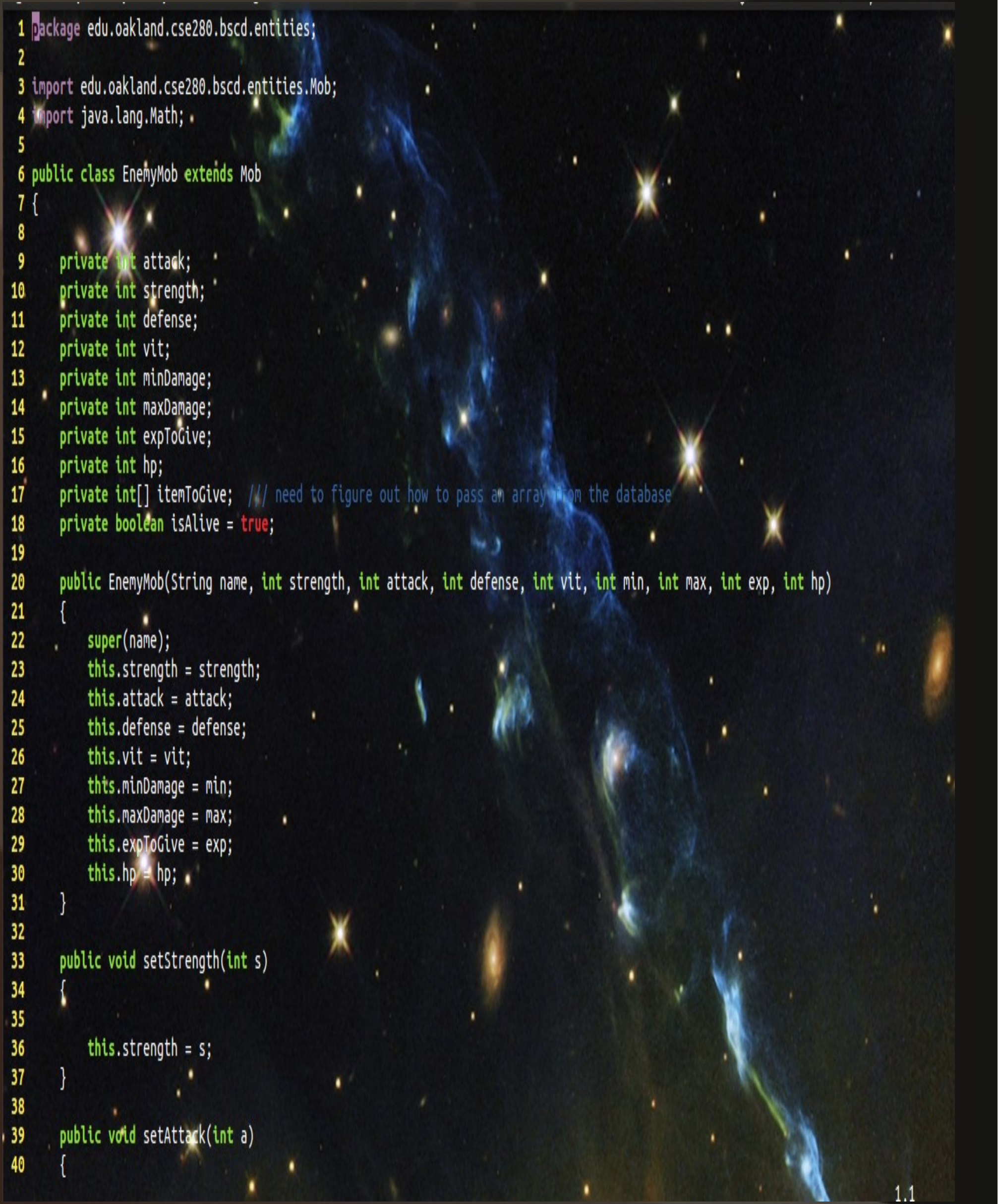
Systems/Engines

Combat System

(Damage Control / Enemy Mob

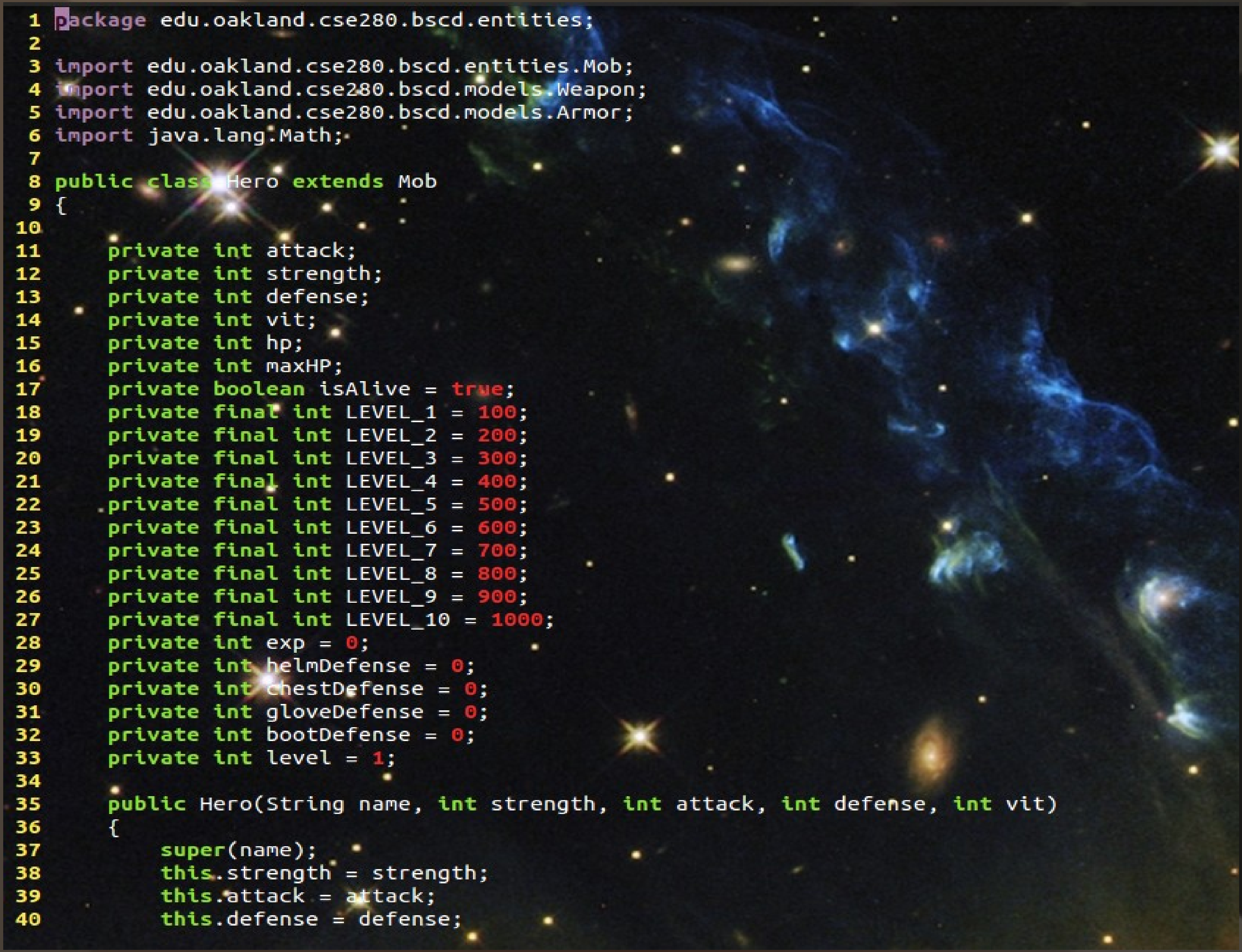


```
    while (true)
    {
        System.out.println("");
        System.out.println("Enemy HP = " + enemy.getHP());
        System.out.println("Hero HP = "+ hero.getHP());
        System.out.println("HERO ATTACKING");
        System.out.println("HERO hits for " + enemy.doDamage(weapon.getMinDamage(), weapon.getMaxDamage(), hero.getStrength()));
        System.out.println("enemy HP = " + enemy.getHP());

        if (!enemy.isAlive())
        {
            System.out.println("Enemy is Dead!!!!");
            if (hero.level(enemy.getExp()))
            {

                System.out.println("You leveled!!!");
                System.out.println("Your Level is " + hero.getLevel());
            }

game/src/edu/oakland/cse280/bscd/test/TestFight.java                    77,1        82%

  // figure out the items to give on death
  //
  //
  //
  public int doDamage(int weaponMin, int weaponMax, int heroStrength)
  {

      int damage = ((weaponMin + (int)(Math.random() * ((weaponMax - weaponMin) + 1))) + heroStrength) - this.defense;

      damage = (damage < 1) ? 1 : damage;

      hp -=damage;

      if (hp<=0)
          this.isAlive = false;

      return damage;
  }
game/src/edu/oakland/cse280/bscd/entities/EnemyMob.java                 129,1       93%
```



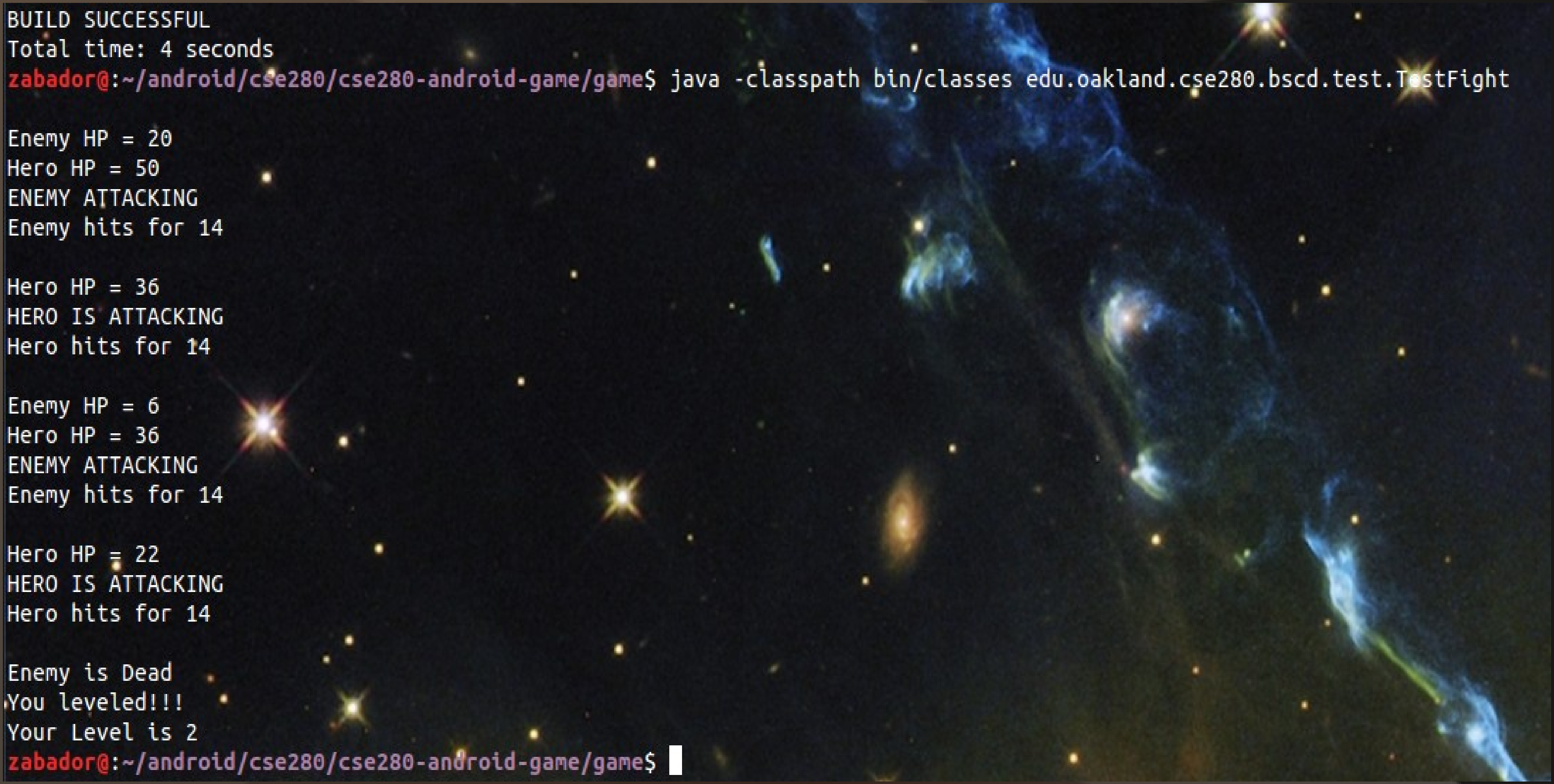
```
package edu.oakland.cse280.bscd.entities;

import edu.oakland.cse280.bscd.entities.Mob;
import java.lang.Math;

public class EnemyMob extends Mob
{

    private int attack;
    private int strength;
    private int defense;
    private int vit;
    private int minDamage;
    private int maxDamage;
    private int expToGive;
    private int hp;
    private int[] itemToGive;  // need to figure out how to pass an array from the database
    private boolean isAlive = true;

    public EnemyMob(String name, int strength, int attack, int defense, int vit, int min, int max, int exp, int hp)
    {
        super(name);
        this.strength = strength;
        this.attack = attack;
        this.defense = defense;
        this.vit = vit;
        this.minDamage = min;
        this.maxDamage = max;
        this.expToGive = exp;
        this.hp = hp;
    }

    public void setStrength(int s)
    {

        this.strength = s;
    }

    public void setAttack(int a)
    {
                                                                          1,1
```

Hero System



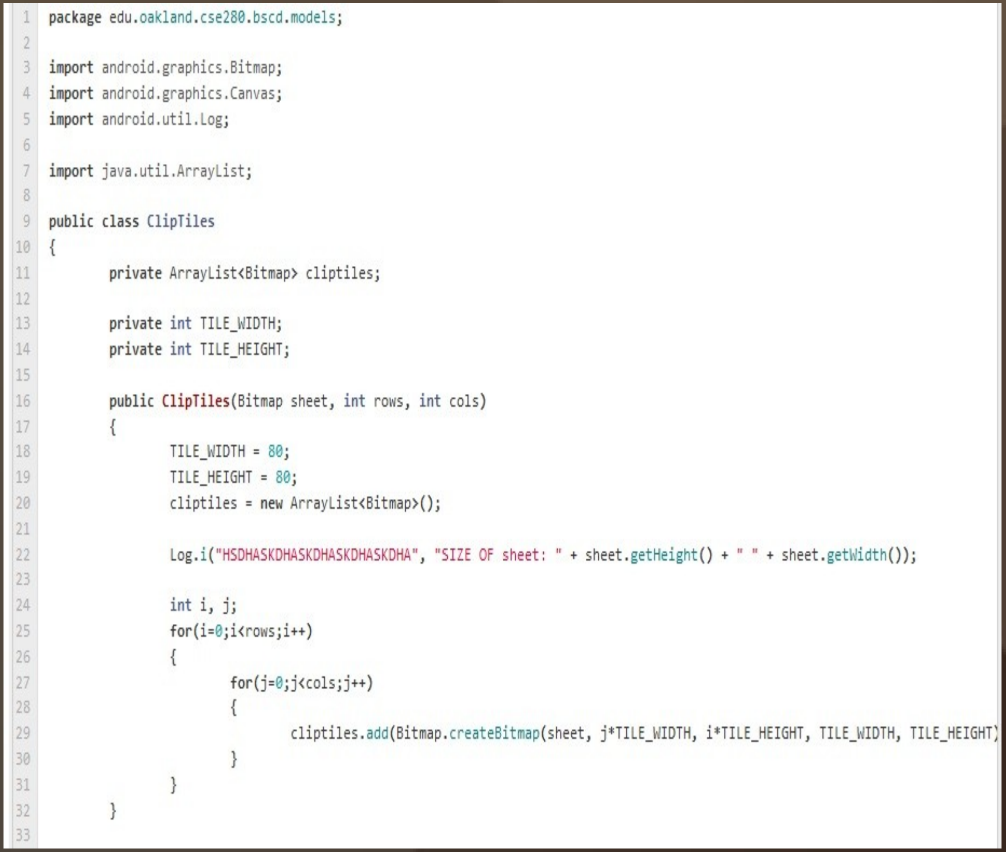
```java
package edu.oakland.cse280.bscd.entities;

import edu.oakland.cse280.bscd.entities.Mob;
import edu.oakland.cse280.bscd.models.Weapon;
import edu.oakland.cse280.bscd.models.Armor;
import java.lang.Math;

public class Hero extends Mob
{

    private int attack;
    private int strength;
    private int defense;
    private int vit;
    private int hp;
    private int maxHP;
    private boolean isAlive = true;
    private final int LEVEL_1 = 100;
    private final int LEVEL_2 = 200;
    private final int LEVEL_3 = 300;
    private final int LEVEL_4 = 400;
    private final int LEVEL_5 = 500;
    private final int LEVEL_6 = 600;
    private final int LEVEL_7 = 700;
    private final int LEVEL_8 = 800;
    private final int LEVEL_9 = 900;
    private final int LEVEL_10 = 1000;
    private int exp = 0;
    private int helmDefense = 0;
    private int chestDefense = 0;
    private int gloveDefense = 0;
    private int bootDefense = 0;
    private int level = 1;

    public Hero(String name, int strength, int attack, int defense, int vit)
    {
        super(name);
        this.strength = strength;
        this.attack = attack;
        this.defense = defense;
```

Running Fight Test

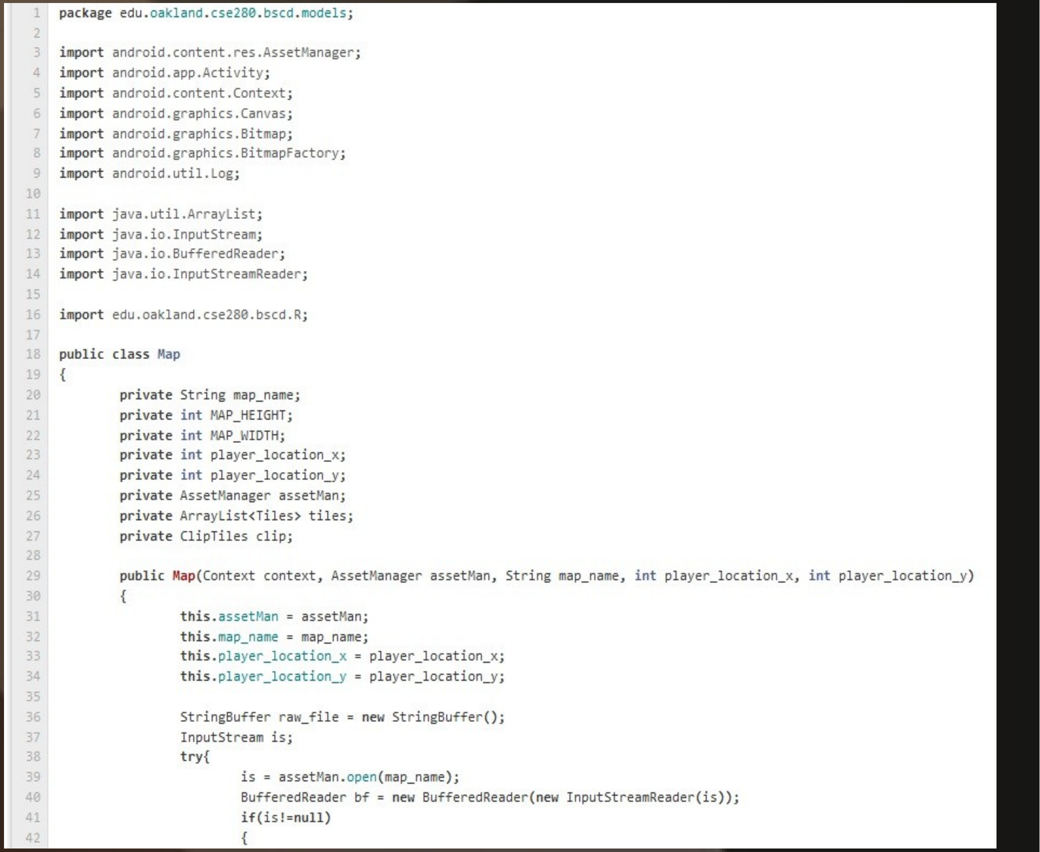


```
BUILD SUCCESSFUL
Total time: 4 seconds
zabador@:~/android/cse280/cse280-android-game/game$ java -classpath bin/classes edu.oakland.cse280.bscd.test.TestFight

Enemy HP = 20
Hero HP = 50
ENEMY ATTACKING
Enemy hits for 14

Hero HP = 36
HERO IS ATTACKING
Hero hits for 14

Enemy HP = 6
Hero HP = 36
ENEMY ATTACKING
Enemy hits for 14

Hero HP = 22
HERO IS ATTACKING
Hero hits for 14

Enemy is Dead
You leveled!!!
Your Level is 2
zabador@:~/android/cse280/cse280-android-game/game$
```

Mapping Engine



```java
package edu.oakland.cse280.bscd.models;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.util.Log;

import java.util.ArrayList;

public class ClipTiles
{
        private ArrayList<Bitmap> cliptiles;

        private int TILE_WIDTH;
        private int TILE_HEIGHT;

        public ClipTiles(Bitmap sheet, int rows, int cols)
        {
                TILE_WIDTH = 80;
                TILE_HEIGHT = 80;
                cliptiles = new ArrayList<Bitmap>();

                Log.i("HSDHASKDHASKDHASKDHASKDHA", "SIZE OF sheet: " + sheet.getHeight() + " " + sheet.getWidth());

                int i, j;
                for(i=0;i<rows;i++)
                {
                        for(j=0;j<cols;j++)
                        {
                                cliptiles.add(Bitmap.createBitmap(sheet, j*TILE_WIDTH, i*TILE_HEIGHT, TILE_WIDTH, TILE_HEIGHT)
                        }
                }
        }

```



```java
package edu.oakland.cse280.bscd.models;

import android.content.res.AssetManager;
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;

import java.util.ArrayList;
import java.io.InputStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;

import edu.oakland.cse280.bscd.R;

public class Map
{
        private String map_name;
        private int MAP_HEIGHT;
        private int MAP_WIDTH;
        private int player_location_x;
        private int player_location_y;
        private AssetManager assetMan;
        private ArrayList<Tiles> tiles;
        private ClipTiles clip;

        public Map(Context context, AssetManager assetMan, String map_name, int player_location_x, int player_location_y)
        {
                this.assetMan = assetMan;
                this.map_name = map_name;
                this.player_location_x = player_location_x;
                this.player_location_y = player_location_y;

                StringBuffer raw_file = new StringBuffer();
                InputStream is;
                try{
                        is = assetMan.open(map_name);
                        BufferedReader bf = new BufferedReader(new InputStreamReader(is));
                        if(is!=null)
                        {
```

Project Time line

- By October 2nd:

  ~~Main map designed~~

  ~~UI prototype~~

  ~~Quest & stories designed~~

- By October 16th:

  ~~Map drawn out~~

  ~~UI finalized~~

  ~~List of items~~

  ~~Graphics engine prototype~~

- By October 30th:

  ~~Loadable game~~

  UI implemented

  Coding

- By November 13th:

  Main quest finalized and runnable

  Characters fully developed

  Items implemented

  Game runnable

- By November 27th:

  Game fully functional

  Side quests added

  Performance tweak

  Finished