Duck, Ants!

Game Design Document

Nathan Pinkelton

Overview:

This RPG game will be a very basic 2D game using simpleGE and pygame

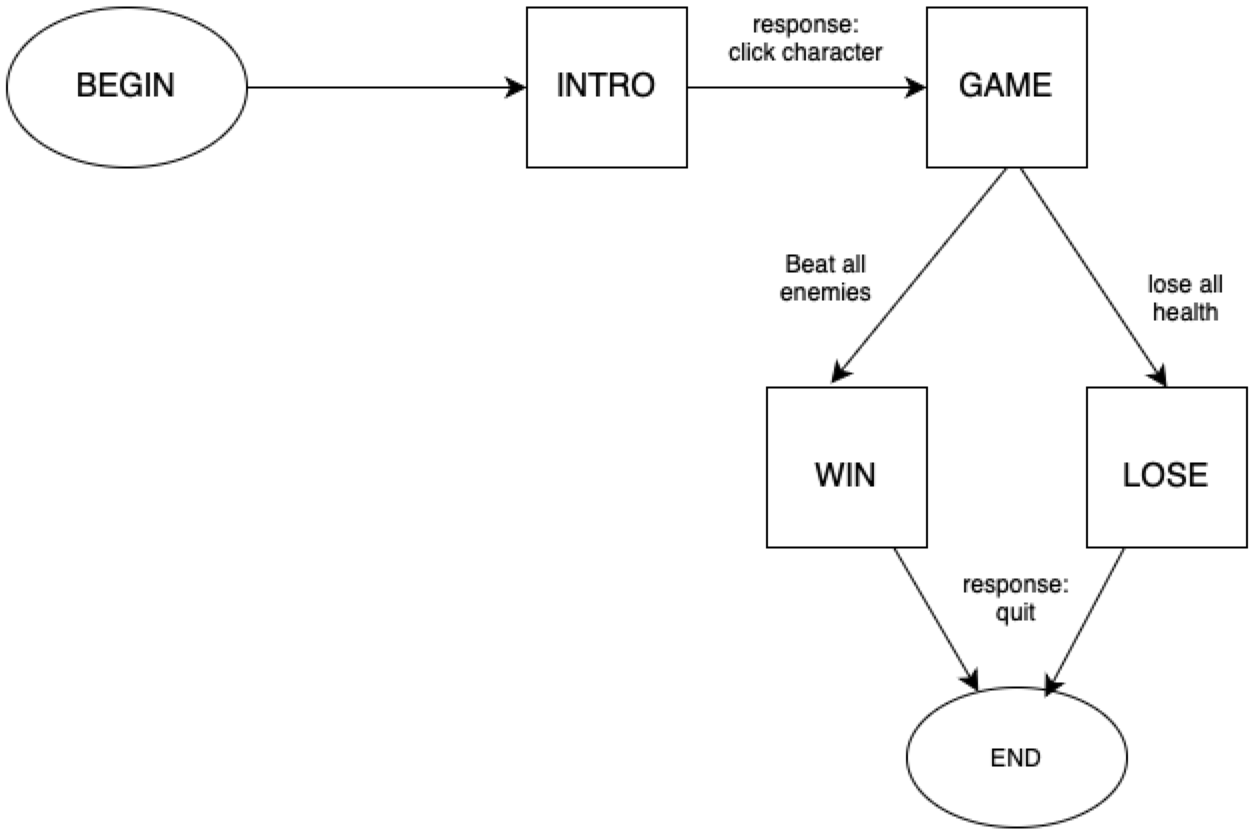
The idea is that you control a character (your choice between 3?) and will be able to move your character around a screen (up, down, left, right). The starting screen will give you a choice of characters, and after you click on your choice, clicking Start will send you to the gameplay screen. On this screen there will be a few enemies that you can run into. When you collide with them, you will "fight" them. Your character's stats will be in the bottom left corner at all times, while the enemy's stats will appear in the bottom right after you collide with them. The "fight" will be a slightly altered version of the turn-based combat system we've previously used. It will be altered so that, when the collision event happens, a turn passes. Also, when a collision happens, a grunt or some similar sound will play. When you beat all the enemies on the screen, you will be sent to a "You Win!" screen. If you lose, you will be sent to a "You Lose…" screen.

When the game begins, the intro screen will be loaded. It will explain that you will run into your enemies to fight them. It will also explain that once you click on your character, you will be sent into the game. The Win and Lose screen will contain buttons that will either redirect the user to play again, which will send them to the start screen, or to quit, which will exit the game.

Game States:

Intro -> game -> win or lose -> Quit or Intro

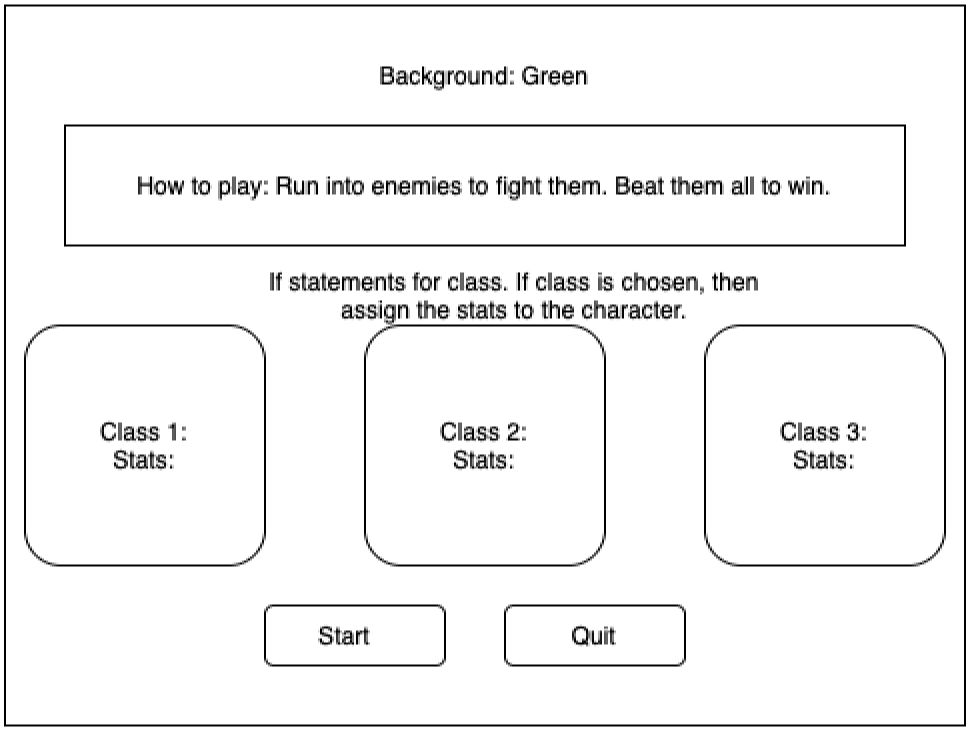
State Transition Diagram:



This game will be very linear in its progression. The user will begin the program, pick their character in the intro screen, play the game, and either receive a win or lose screen after, which will then lead to the end of the program. Each state will be a subclass of the simpleGE Scene class. The intro screen will display instructions and 3 buttons to choose from. When the button is clicked, it will close the intro screen, load the game screen, and load the details of the user's choice to their character. The choices will be different classes for their character to be, which will contain different stat values that will be assigned to their character when chosen. When the user's character has 0 health, or all the enemies on screen have been defeated, they will be sent to the next screen, which is dependent on their outcome. After that, the user will be allowed to quit. A potential start over button may be included as well, which will start the user at the intro screen.

Intro Screen:

The intro screen will give us details on how to play and how your character's stats will look.


Background: Green
How to play: Run into enemies to fight them. Beat them all to win.
If statements for class. If class is chosen, then assign the stats to the character.
Class 1: Stats:
Class 2: Stats:
Class 3: Stats:
Start
Quit

This scene has four main components:

- **howTo –** a stock simpleGE multiLabel containing instructions for how to play
- **classBtn –** multiLabels that show the different stats you can play with. Will be buttons as well.
    - Will make 3 of them
- startBtn – a stock button indicating “Start”
- quitBtn – a stock button indicating “Quit”

I want it so that when a class button is pressed, the information is automatically applied to the character’s stats. Whan start is clicked, then this screen will disappear

The initializer will create all attributes and set up the sprite list:

init():

Super initialize

Fill screen background with the color green

Set caption as RPG

Define sprites on the screen:

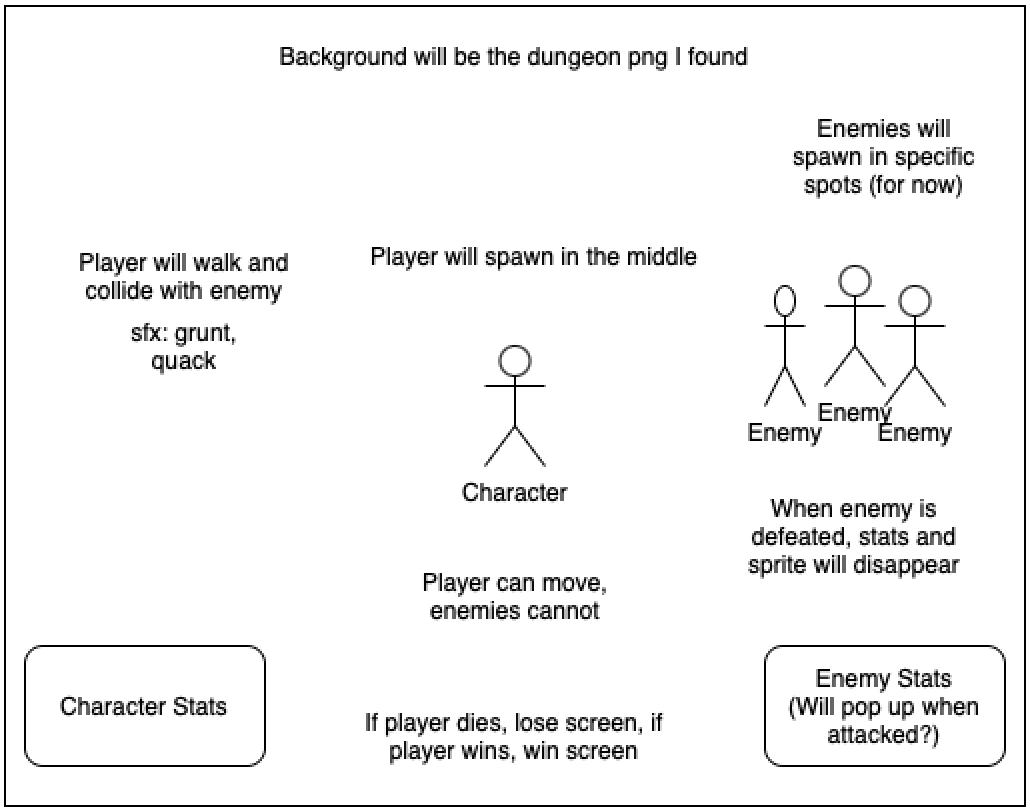
- Make howTo a multiLabel that is meant to just be read
    - Add textLines containing instructions
    - Align it on the screen as wanted
- Use Button Class in simpleGE
    - Create buttons for Class 1, 2, & 3
        - When clicked, stats will be applied to Character’s sprite
        - Default stats will be made for if user chooses none
    - Create buttons for Start and Quit
        - Start will end the intro screen and lead to game play
        - Quit will end the intro screen and the program

Add howTo, classBtn(s), startBtn, and quitBtn to sprites(?)

Process method:

- process():
- If classBtn1 is pressed:
    - Assign classBtn1’s stats to Character’s stats
- If classBtn2 is pressed:
    - Assign classBtn2’s stats to Character’s stats
- If classBtn3 is pressed:
    - Assign classBtn3’s stats to Character’s stats
- If the start button is pressed:
    - Set response to “Start”
    - Stop the scene
- If the quit button is pressed:
    - Set the response to “Quit”
    - Stop the scene

The Game class:


Background will be the dungeon png I found
Enemies will spawn in specific spots (for now)
Player will walk and collide with enemy
sfx: grunt, quack
Player will spawn in the middle
Enemy
Enemy
Enemy
Character
When enemy is defeated, stats and sprite will disappear
Player can move, enemies cannot
Character Stats
If player dies, lose screen, if player wins, win screen
Enemy Stats (Will pop up when attacked?)

simple.GE.Scene.

Game class's visual elements:

- **characte**r – will be a sprite that can move in cardinal directions. When it collides with enemies, Character Stats and Enemy Stats will be affected
- **enemy** – will be a sprite that cannot move. Will be subject to collision with character
    - Will most likely make 3 enemies (for now)
- **charStats** – Will show character's stats and stay on screen. Only thing that should change is their health. If they take damage, it should carry over into the next battle, not be reset. multiLabel?
- **enemStats** - Will show the enemy's stats. Should only appear when a collision happens. When enemy loses all health, their stats and sprite should disappear. multiLabel?

BTS:

- **enemCounter** - Will keep track of how many enemies there are

- **Fight –** A slightly altered tbc system that will affect the stats of character and enemy

```
Initializer:
Set background to dungeon png
Initialize fightSnd to collision sound effect
Load Character sprite png
    character gets Character (will be a class, sprite)
Load Enemy sprite png
    enemy will get Enemy (will also be a class, sprite. Similar to Character)
Set enemCounter in here too
Create charStats instance?
Make multiLabel that receives input after collisions.
Create enemStats instance?
Add character, enemy, charStats, enemStats to sprites.

Event handling:
process:
If character collides with enemy:
    Play fightSnd
    Display enemStats
    A round of turn-based combat will happen (which should affect both entities health)
    If character dies
    Exit game screen and send user to the Lose screen
    If enemy dies
    Remove enemy sprite from screen and remove enemy stats from screen.
If enemCounter reaches 0
```

Exit game screen and show Win screen

Components of Game Class:

- Collision
- MultiLabel for stats to pop up on screen

Also Needed:

Character Class:

Will be connected to the sprite the user moves around

```
    (process) If directional arrows are clicked
        Move Character in that direction
```

Enemy Class:

Might not need one?

If anything, it may be a list of enemies so that we can make multiple in the game

Enemy only needs to be loaded onto screen, which can be done in game. enemSats is what will be affected by collision, which means that we will need to code something like:

When character collides with enemy:

Character fights Enemy

(This will only affect stats until one of the two dies)

Stats Class: (Where the magic happens)

Define variables in play (hitPoints, hitChance, maxDamage, armor, name)

These need to be communicated with labels displaying the info on screen

I think I want the enemies to have randomized stats (might be done in Game class)

Win Screen:

simpleGe scene

Label saying “you win!”

Button that says Start over

    If start over button is clicked, close scene, load intro screen

Button that says quit

If quit button is clicked, close scene

Lose Screen:

simpleGe scene

Label saying “you lose…”

Button that says Start over

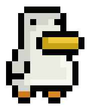
    If start over button is clicked, close scene, load intro screen

Button that says quit

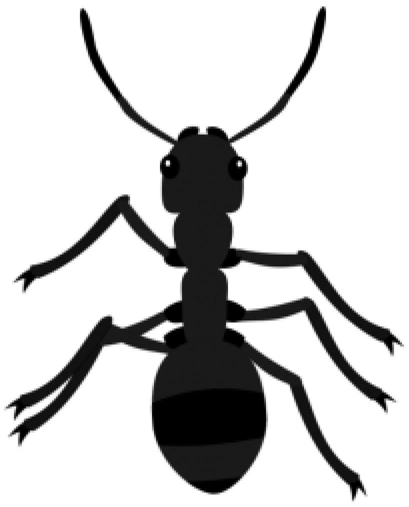
If quit button is clicked, close scene

Asset plan:

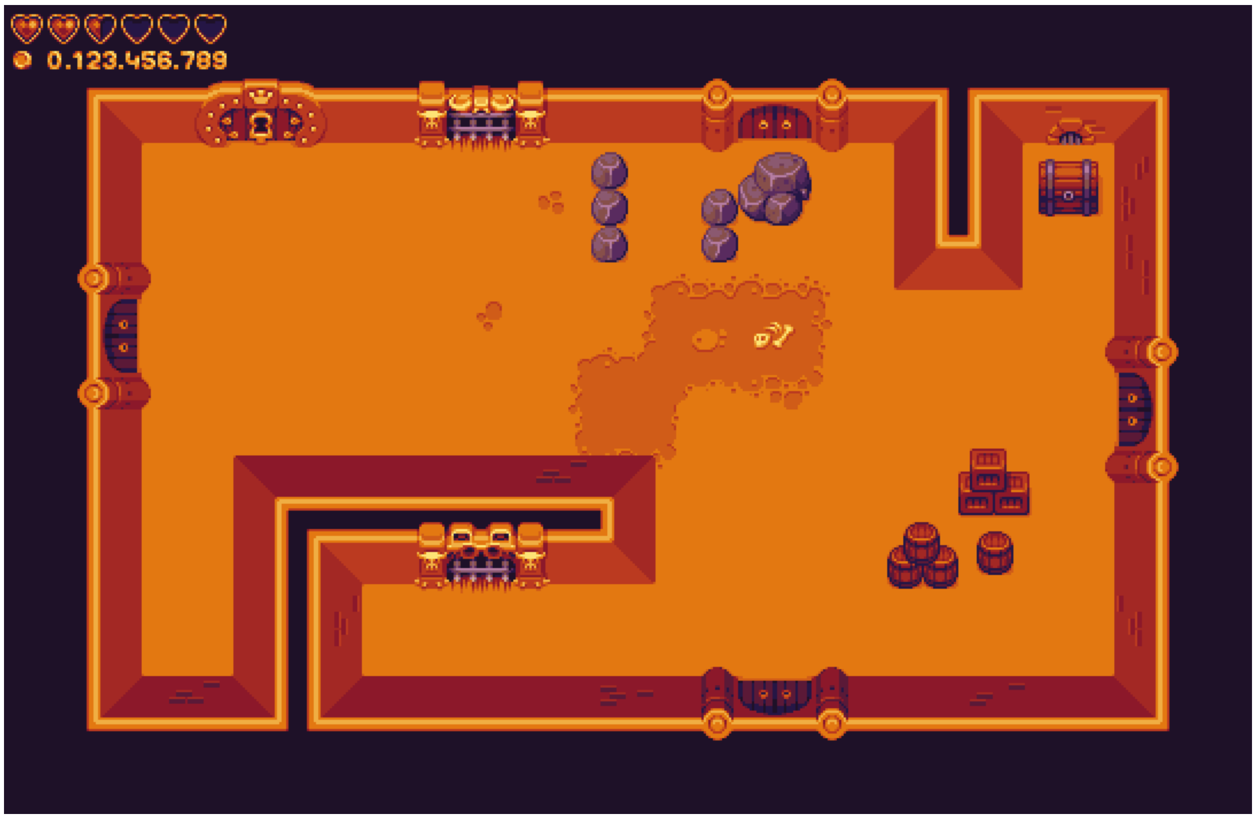
Character png (duck



Enemy png (ant)



Dungeon.png


0.123.456.789