BOMB JACK ™

[SPOOKY EDITION]

CREATED BY JASON HEFLINGER AND BEN SALAK

A CSSE220 PROJECT MADE WITH

BLOOD, TEARS, AND CAFFEINE

(SORTA BUT NOT REALLY)

PRESS ENTER TO PLAY

Origin and Objective of Game

- Design a game that showcases the coding skills learned in CSSE 220

- Implement the design using correct object oriented design principles

- Learn something new

Our Characters

| The Hero | Spooky Scary Skeleton | Angry Head | Even More Angry Head |

Our Levels – Level 1

Our Levels – Level 2


```
| BGBA      EBGB |
|  __        __  |
|                |
|       HB       |
|       __       |
|                |
| BGBA      EBGB |
|  __        __  |
|                |
|   B        B   |
```

Our Levels – Level 3

Our Extra Features

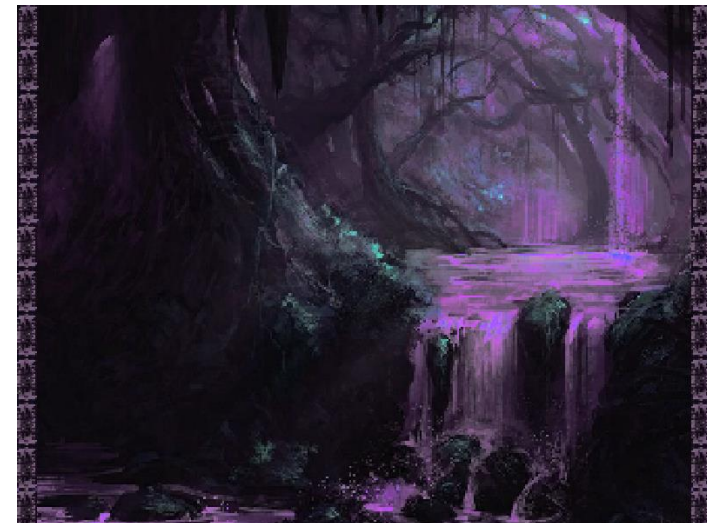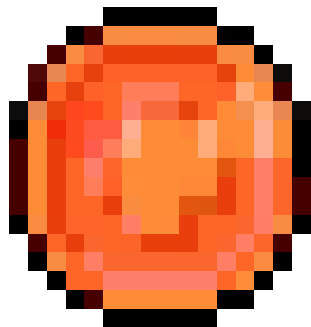- Images for Player, Aliens, and Environment

- Bombs that time out

Our Extra Features

- Images for Player, Aliens, and Environment

- Bombs that time out

Our Extra Features

- Help screen

- Title and Game Over Screens

INSTRUCTIONS

use arrow keys to move

press u/d to to advance/return to a level

press r to restart

Press space to exit instructions or wait 2 seconds

[SPOOKY EDITION]
CREATED BY JASON HEFLINGER AND BEN SALAK
A CSSE220 PROJECT MADE WITH
BLOOD, TEARS, AND CAFFEINE
(SORTA BUT NOT REALLY)
PRESS ENTER TO PLAY

GAME OVER

BONUSES EARNED:
+ZILCH LEVEL FAILED
+ZILCH DEAD
+ZILCH YOU FAILED!!!!! SORRY!!
+0000 TOTAL POINTS
FINAL SCORE: 200

PRESS ENTR TO RESTART

Our Extra Features

- Animated level transitions with bonuses and motivational comments!

- Animation of sprites

Our Extra Features

- Animated level transitions with bonuses and motivational comments!

- Animation of sprites

Good and Bad Aspects of The Game Design
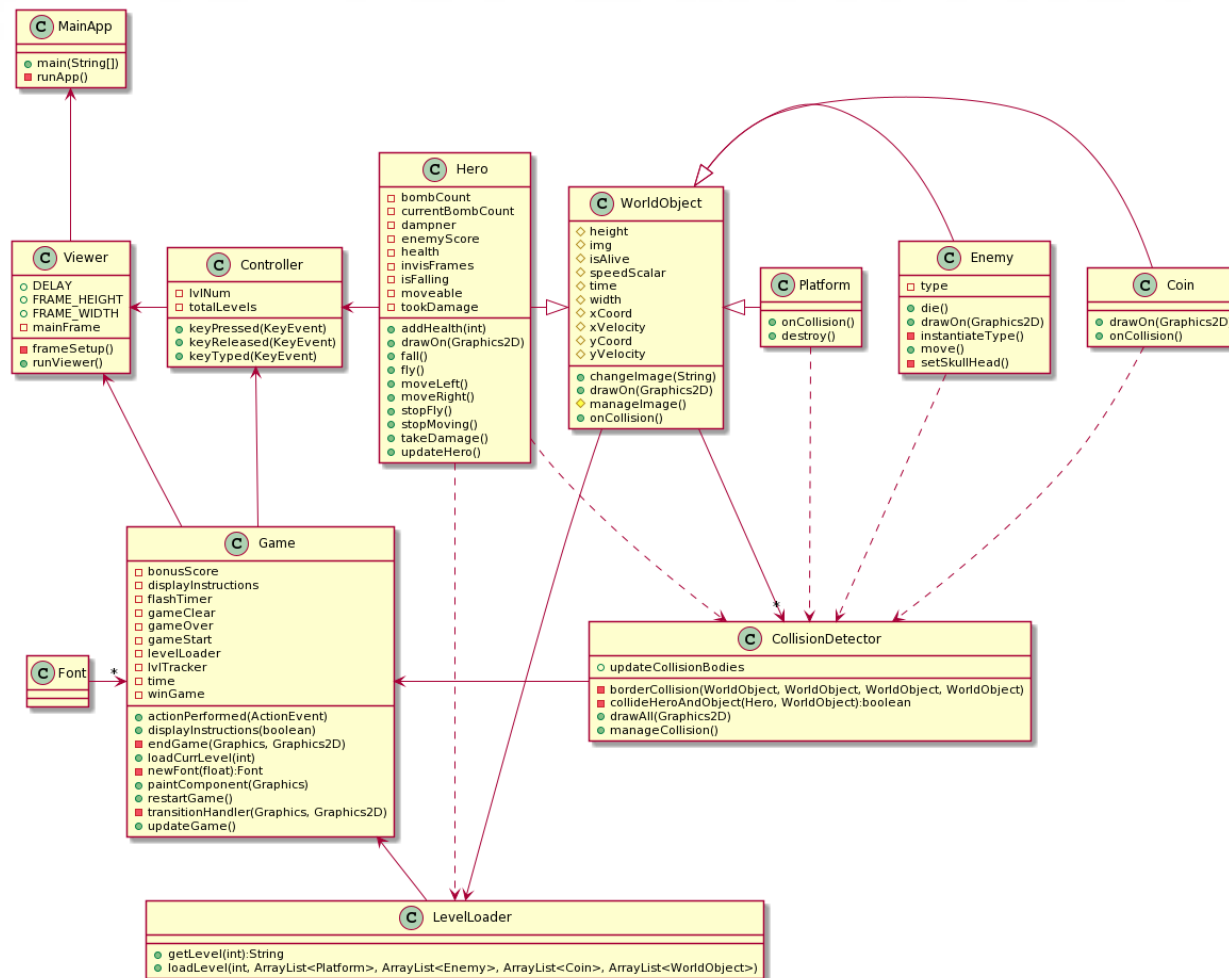
The Good
- Use of Inheritance to reduce duplicated code in world objects
  - Enemies, platforms, and coins (bombs) all use the inherited methods
- Classes kept small and focused on one task
  - 11 total classes
  - Keeps coupling low

The Bad
- Originally code did not follow object - oriented software design principles
  - Hero class handled collisions
  - Refactored code to resolve issues
- Accidentally wrote duplicate methods to handle the same thing
  - 2 ways to restart levels
- Collision
  - Implemented multiple times
  - Based off position
  - Based off velocity

UML Diagram

Questions?