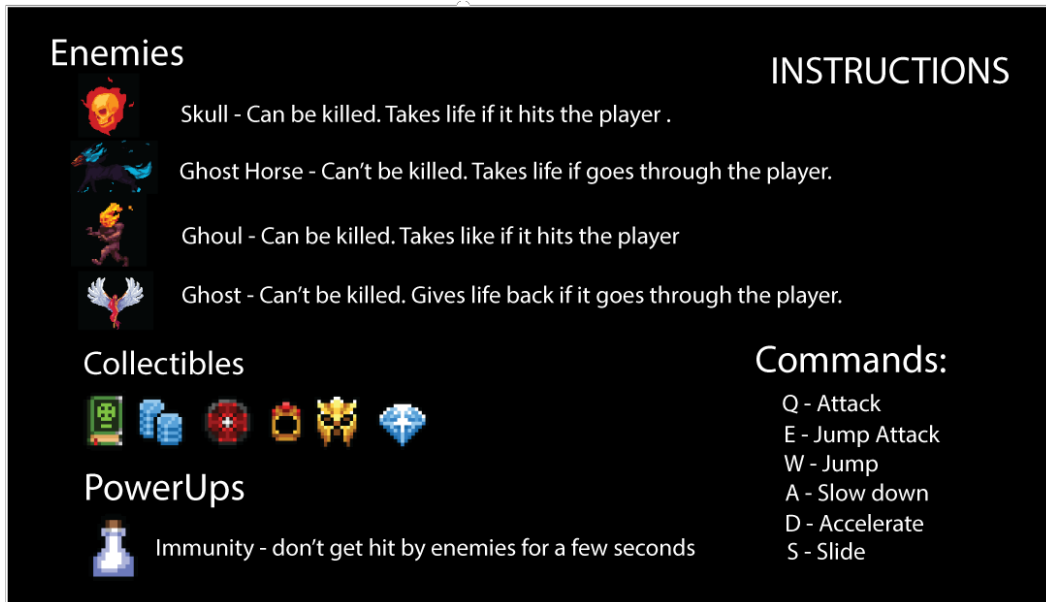


Documentation

Concept



Health System

The player **takes/receives damage** from passing through ghost enemies. Normal enemies **give damage** once and then disappear.

The game ends when the life bar is depleted.

Main Menu Support

The player starts the game in the **Main Menu**, presses '**SPACE**' to start the game, Presses '**P**' to pause/unpause, then '**SPACE**' to return to the main menu from the **Game Over Screen**. Instructions can be accessed through mouse hover over the specific icon in the **Main Menu** and **Pause Screens**.

Scrolling Background

A scrolling background composed of **wall tiles** and **floor tiles** is generated in the same manner as the **enemies**, **power ups** and **collectibles**. It follows the player's speed, it is generated in front of the current screen and disappears when it reaches a certain position behind the screen.

Heads-Up Display

The **enemies disappear** when they hit the player if they are implemented that way, a **life bar** can be seen **increasing/decreasing** at the top, along with the **current score** and **time**. Found **collectibles** are also displayed. Each time the **power up** is active, it can be seen for its entire duration to alert the player of its existence.

Scoring System

Hittable/Collectible enemies/powerups/collectibles give score points each time they collide with the hero. The score is also composed of the **distance** run by the player(as an extension to the time scoring system). The game **SPEEDS UP** as more time passes, so the score is based on the **speed** and the **time** passed.

Power Ups

The game implements **IMMUNITY**. It lasts 8 second after it is collected by the player and it allows it to not get hit by the enemies for that amount of time. A **SUPER ATTACK** power up is also **fully implemented** in the game, but a last minute code crash made it unusable. It destroys/collects everything on the visible screen with any hero attack type.

Leaderboard

The game over screen displays the **best 3 scores** obtained in that session. The **current score** can also be seen and can be **found on the list**.

Game Implementation

The **GameTest** Class has an instance of the **GameStateClass**, which implements all of the screens(**GameOver, MainMenu, Game, Pause**). The **GameTest** Class manages all the screens. They both Instantiate, Update and Render all of objects.

Hero

The Hero class was a Component Implementation experiment. It takes User Input, has multiple sprites and actions.

Graphical Entities

All the sprite owning entities outside of the hero are implemented based on **INHERITANCE** and **TEMPLATES**. There are Graphic Entity Classes that are then stored in Manager Classes that contain a vector of that entity type.

```
+ // ...
+ class CEntity { ... };

+ // ...
+ class CBackgroundPiece { ... };

+ // ...
+ class CStaticEntity { ... };

+ // ...
+ class CCollisionEntity { ... };

+ // ...
+ class CEnemyEntity { ... };

+ // ...
+ class CCollectibleEntity { ... };

+ // ...
+ template<class T>
+ class CGraphicsEntityVector { ... };

+ // ...
+ template<class T>
+ class CStaticEntityVector { ... };

+ // ...
+ template<class T>
+ class CBackgroundVector { ... };

+ // ...
+ template<class T>
+ class CCollisionVector { ... };

+ // ...
+ template<class T>
+ class CEnemyVector { ... };

+ // ...
+ template<class T>
+ class CCollectibleVector { ... };

// Sprite Containing Game Entity Vector Managers
Hero* m_hero;
CEnemyVector<CEnemyEntity>* m_skullVectorManager;
CEnemyVector<CEnemyEntity>* m_ghoulVectorManager;
CEnemyVector<CEnemyEntity>* m_ghostVectorManager;
CEnemyVector<CEnemyEntity>* m_ghostHorseVectorManager;
CBackgroundVector<CBackgroundPiece>* m_backgrounWallpaperVectorManager;
CBackgroundVector<CBackgroundPiece>* m_backgroundFloorVectorManager;
CCollectibleVector<CCollectibleEntity>* m_collectibleVectorManager;
CCollectibleVector<CCollectibleEntity>* m_powerupVectorManager;
// Screen display foun/not found collectibles and powerups
CStaticEntityVector<CStaticEntity>* m_displayCollectiblesManager;
CStaticEntityVector<CStaticEntity>* m_displayPowerupsManager;
```

Improvements

The **next course of action**, if it were not for the time, would have been fixing the existing SUPER ATTACK Power Up, code clean-up over the Hero class, more power up/enemy reaction implementations, finding more sounds that work and a full Component Implementation.