My Game: Dungeon Hunter

Requirements

Intro:

Given the scale of a platformer, and the time I was given for the assignment I was unable to make a full platformer. By this I mean there are not several levels, no story, no story beats, or real level design. At the current state my game is a tech demo that could be used to create a full platformer.

I wonder consider this game in an Alpha version. Everything implemented works but could use minor tweaking so be prefect and fluid.

This game comes with 2 levels. The first level is a test level. This level features every item currently in the game within close proximity to each other. This space was where the game was made is used for testing new assets.

The second level is the Demo level. This is one simple level of a platformer created using all the current game assets. This level is very easy to complete but gives you and idea of most of the things you could do.

Base:

This game is a platformer where you player as the Hunter, a cloaked rogue equipped with a sword. The Hunter treks through minotaur filled dungeons in search of loot and his new equipment. The Hunter is well prepared for these dungeons are armored. It takes more than bumping into an enemy to hurt him. Unfortunately, the minotaur’s punches pack a lot of force. Minotaur are not the brightest creatures, and their best sense is sound. However once heard, if someone was to stand completely still the minotaur would be unable to detect them. They can only see movement.

Parts to the Game:

There is quite a few parts to this game each will be listed and broken down.

To Start we are going to cover the custom components

Hover Component:

* The hover component is a component that can be added to any pickup of your choice.
* It is meant to draw to players eye to the pickup
* The hover component causes the pickup so slowly bounce up and down like it is floating

Health Component

* Is a component that is responsible for handling health
* The Heath Component can be added to any actor.
* It holds the health, respawn health, and max health values
* It handles take damage and healing
* I handle death
* It keeps track of the player’s current checkpoint.

Inventory Component

* Is a component for handling inventory
* A Component that can be attached to an actor
* Only used on the player
* Handles the players inventory (weapons)
* Allows to player to equip and unequip guns
* Updates Hud icon
* Keeps track of current weapon

Next, we have our Actors

The Hunter Player

* Is our player. Can move around the level, jump shoot, and swing a sword. All actions are animated as well as given sounds
* Can move with A and D
* Can Jump with spacebar
* Can melee attack on left mouse click
* Can pick up a weapon with E
* Can Shoot on right click if they have a gun
* Can equip the next weapon with C
* Can equip the previous weapon with X

The Enemy

* Is our only current enemy. A minotaur that patrols an area looking for players to attack. All actions are animated.
* Can be idle
* Can patrol between two points
* Can hear the player
* Can see the player
* Can punch the player

The Enemy spawner

* Spawns an enemy blueprint on the start of a overlap even and does not trigger again until the end over lap has been trigger before trigger start overlap again

The Tiles

* Wall and Floor tiles to for the player to traverse

The Weapon

* A pistol that the player can acquire, it holds six shots and it slow to shoot. It spawns a bullet that deals double the damage of the sword.

The Bullet

* A projectile that can be fired from a weapon with a set damage amount

Moving Platforms

* Platforms that move between two points repetitively

Pick ups

* Is something the player picks up either by pressing E or colliding with the object.
* Parent Pick up class
* Ammo pick up that adds ammo to the player
* Health pick up that heals the player

Controls:

A – Move Left

D – Move Right

Space – Jump

LeftMB – melee

RightMB – shoot

E – Pick up gun

C – Next Weapon

X – Previous weapon

Files included

CRC Cards.

UML diagram.

Game code file.

Packaged game.