Dereferenced Null Pointer

THE SCUTTLER

Final Report

Team Members

Scrum Master: Jonathan Chun

Luis Garibay

Nicholas Karalis

Attila Koksal

William Wang

Overview

Top-down extraction-based shooter set in a Dead Space-inspired world. Players start on their home ship, acting as a central hub to venture and loot abandoned ships, settlements, and mega-structures. Gameplay sees the player explore the stage, looting valuables and killing zombies, space pirates, and other horrors along the way. Players make risk vs reward decisions, balancing their limited inventory size for either profit or firepower to protect themselves against increasingly difficult enemies. Players should extract before midnight to sell their loot and upgrade their arsenal to venture into more difficult stages. Extra features include a fog of war element while exploring, a large variety of weapons with unique attributes, and enterable buildings.

Game Description

Characters

You play as the titular character, **The Scuttler**, as you travel the galaxy looting abandoned planets.

Gameplay

Players are deployed into increasingly dangerous stages to collect resources and survive. They can equip a range of weapons (handgun, shotgun, pistol), use items like stims, and interact with the environment. Extraction before the timer runs out is crucial, or enemies become more dangerous. Fog of war makes exploration tense, especially at night, when visibility is low. Enterable buildings, destructible environments, and ambient noise heighten immersion.

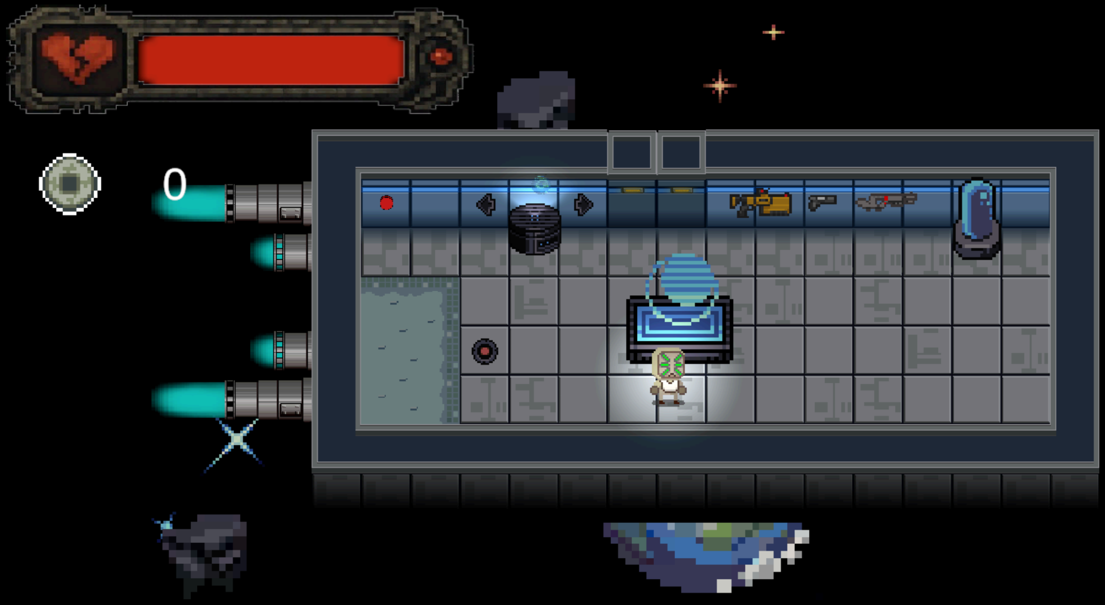
Levels

There are three levels of increasing difficulty, From spooky forests on Elarin Reach, to Abandoned Space Colonies in the Derelict Echo Zone, to an eerie abandoned town in the Virelia Prime Sector. A variety of environments to test the scuttler’s survival skills.

Features List

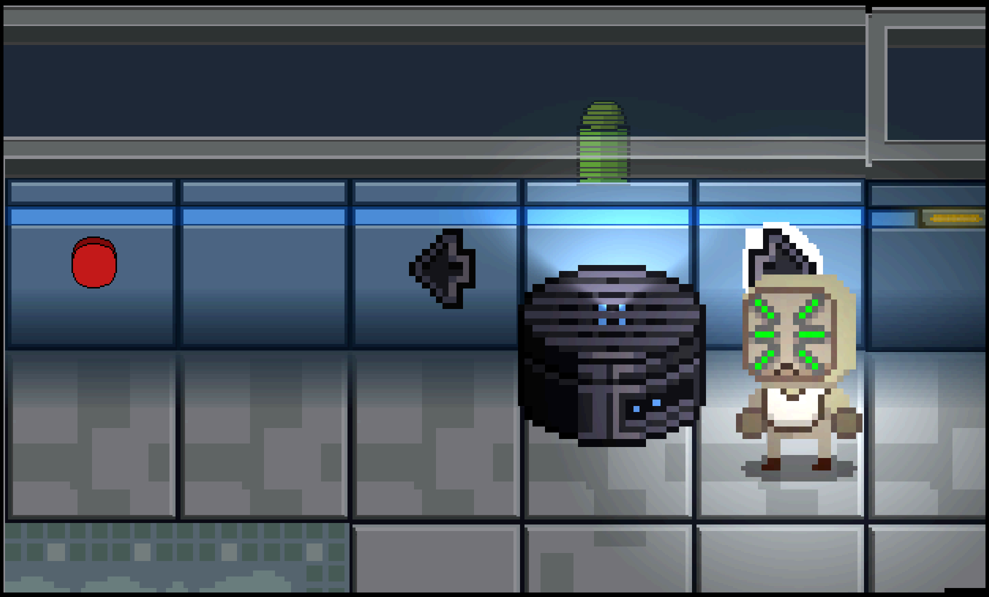
Player Hub World

Between levels, the player can enjoy the serene and calming atmosphere of the hub world. Here, the player can grab his bearings, buy and sell items, purchase new levels, and test basic player functionality.



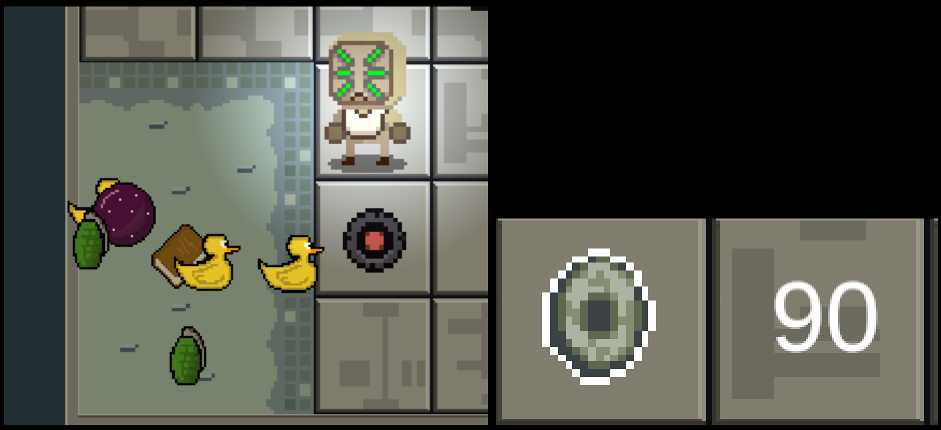
Item Shop

The item shop is located on the ship. The player can flip between a custom catalog of items that the player can purchase. Upon purchasing, the cost is deducted from the player’s wallet and the item is automatically inserted into the player’s inventory.



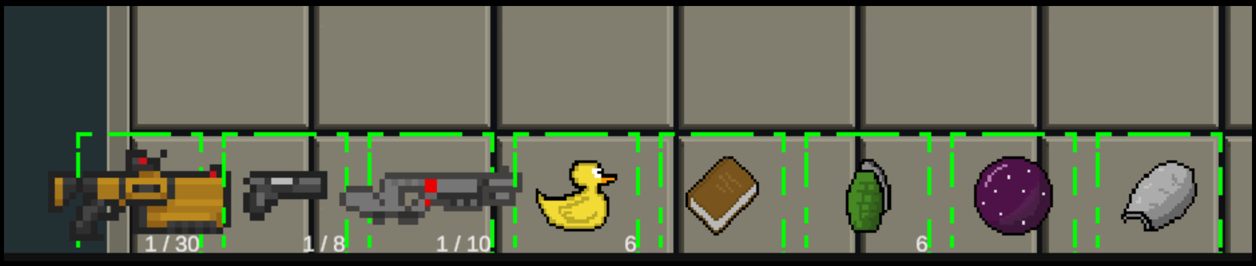
Selling Items

On the ship there is a selling platform where the player can sell his items. Upon dropping the items on the platform, a button can be pressed to convert all items on the platform into currency, which is then automatically added to the player’s wallet.



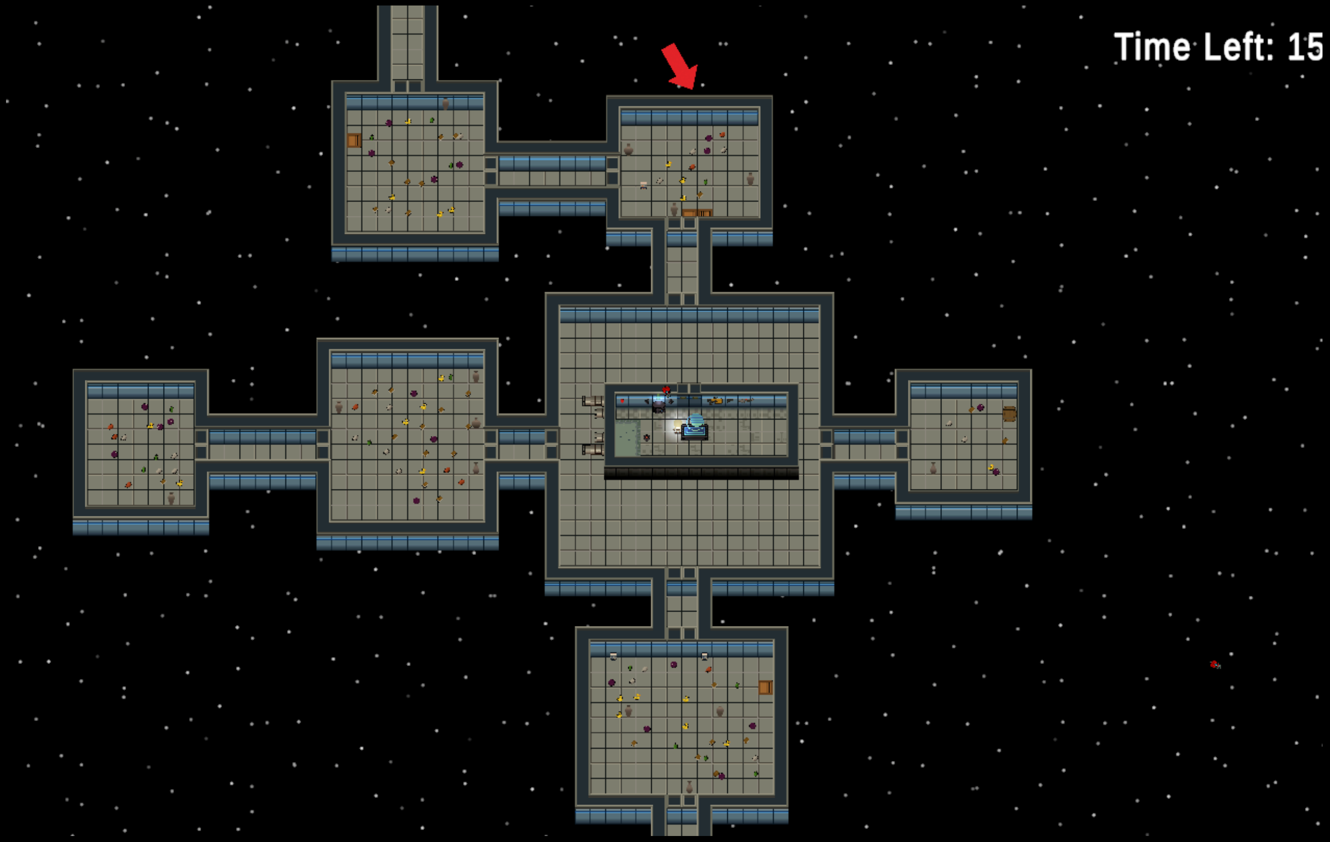
Interactable UI Inventory

Throughout gameplay, the player can make use of an 8 slot inventory located in the bottom portion of the screen. Any weapons and items are placed here. The player can select different slots with the numpad (1-8) or mouse wheel. Items can be dragged between slots as well and stack automatically.



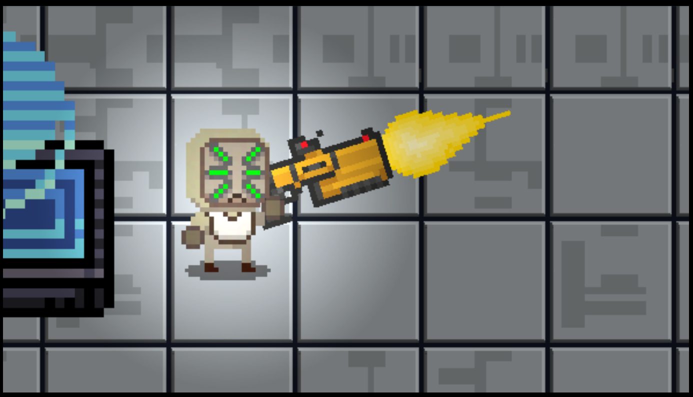
Procedural Level Generation

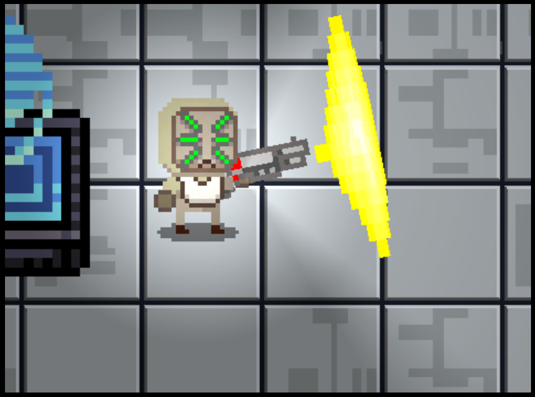
All three levels of The Scuttler make use of several layers of Procedural Generation. Level layouts are generated using the Binary Space Partitioning algorithm, and the result layout is given a specific tileset, enemy spawn points, and item spawns.



Classes of Weapons

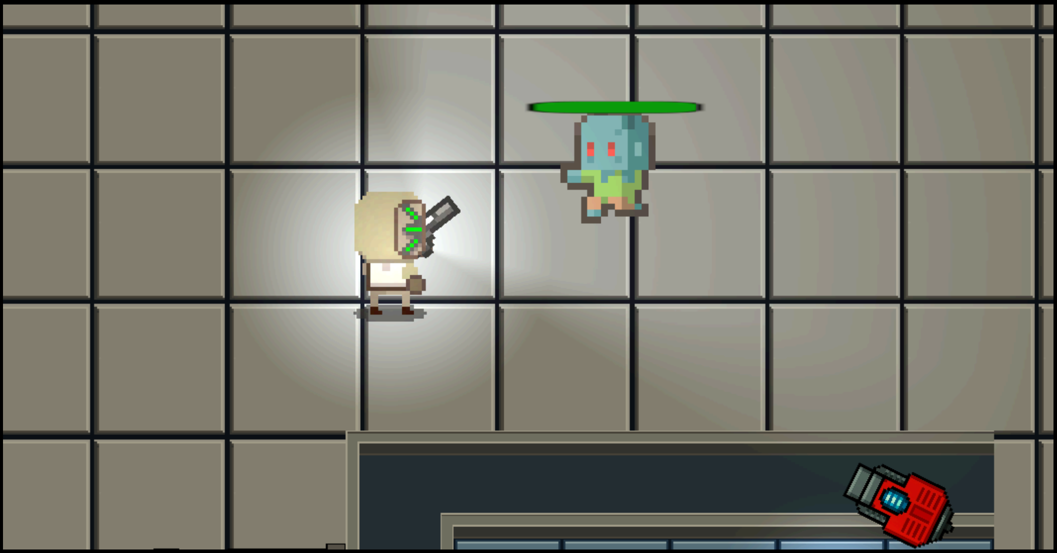
There are 3 classes of weapons that the player can make use of, being the Handgun, Shotgun, and Rifle. Each of these weapons have custom values of damage, bullet spread, and fire rate.





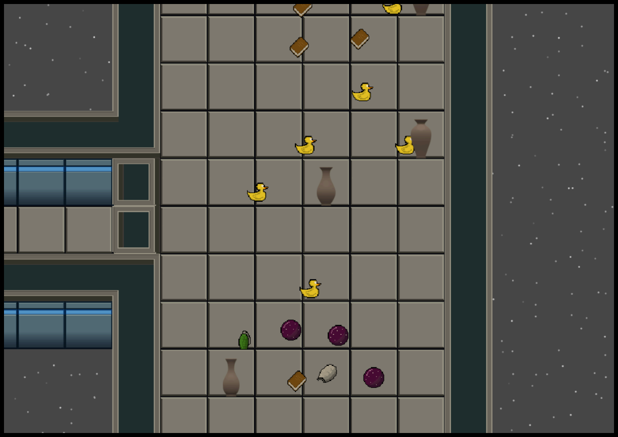
Enemy Variety

There are 3 main types of enemies that the player can encounter in the levels, being the Turret, Melee, and Ranged enemy. Each of these enemies have their own behavior, and spawn at the designated spawn points around the procedurally generated levels.



Item Catalog

A vast list of items has been created that the player can pick up, interact with, and buy and sell. These items spawn procedurally around the levels, and allow the player to accumulate currency to purchase later levels.



Implementation

Game Engine

Unity 6

[Unity 6000.0.44f1]

Scripts: C#

Algorithms: Corridor First Dungeon Generation, Binary Space Partitioning, Inventory Management, Item Interaction System

Tools: GitHub, Unity Hub, Unity Engine

Post Mortem

WHAT TASKS WERE ACCOMPLISHED?

In this project, we finished implementing player movement, player shooting, player aiming, player animation, and an interaction system for the player. In addition, we were able to implement a health system, damage system, and stamina system for the player and the enemies in the game. The inventory system we made allows the player to move, drop, and pick up items in individual inventory slots. We were able to procedurally generate dungeons and towns and generate objects and loot with enhanced graphics. We were also able to implement audio and visual effects to supplement gameplay. We implemented multiple different stages to play on and a menu system to traverse between worlds.

WHAT PLANNED TASKS WERE NOT DONE?

As a group, we were able to finish all the planned tasks we had in mind when developing the Scuttler. However, when we started to play the game more, we realized that more fixes were in need due to bugs.

HOW DID SCRUM WORK FOR YOU?

We conducted weekly meetings to discuss what each of us had finished for the project and what we needed to work on. We designed user stories and assigned them accordingly to each group member to make sure effort contributed was equal.

WHAT WOULD WE DO DIFFERENTLY?

We would try to develop a vertical slice prototype as fast as possible then iterate a demo each week, then use the final few weeks for better audiovisual effects and polish. This way we have a marketable vision for our project every demo, leading up to a fully polished game demo.

Video

The Scuttler - Preview Trailer

Full Gameplay Youtube Link