

Void Run() - Final Report

Alfie Thomasson - 40312683

B/Sc (Hons) Games Development

Ciaran A. Brady - 40312302

B/Eng (Hons) Software Engineering

SET09121– Games Engineering

Word Count:1500

Introduction

Void Run() is a turn based combat dungeon crawler game with roguelike and JRPG elements to it. The game itself was designed to draw upon these elements to create a predominantly combat focused game for the player to fight through an abandoned spaceship and kill aliens as they make their way towards the game's final boss: The Alien Lord.

The game has different enemy types that are randomly selected from when a player moves room, as well as each having different unique abilities that can be randomly determined - So even if you're fighting two of the same enemy, they might have a completely different attack pattern and moves! There are also treasure rooms the player can run into, giving the player a stat improving item, in addition to stat increases after beating an enemy and abilities gained on level up. At the end of the game, the Alien Lord fights the player, who has 4 different behavioural patterns that change periodically depending on the state of the battle (Being more likely to be aggressive as it starts to lose health, for example).

Our game draws inspiration from many traditional JRPGs for the turn-based combat, such as the Pokémon game series, older RPGs like Chrono Trigger and the early Final Fantasy games, as well as more recent takes like Slay the Spire and Paper Mario for our UI design and perspective. Our theming took heavy inspiration from the science fiction genre, particularly the Alien series, and we wanted to emulate a similarly eerie atmosphere to these games.

The game can be found at <https://alfiethomasson.itch.io/void-run>

Changes/Omissions from GDD

The biggest change between our Game Design Document and final product was the removal of the mapping system, which had initially been expected to be a fairly major part of the game. We removed this partially due to time constraints but also due to it being a relatively superfluous addition that would not have added very much in terms of actual functionality, as there was little value in going back to empty rooms.

We also had to remove several mentioned room types as we did not have enough time. Most notably, the game design document mentions trapped rooms that the player would have to work their way through to avoid being hurt, as well as decision rooms that would offer the player choices about how to proceed, such as helping a dying scientist's last request, or breaking into a weapons cache. Ultimately, whilst we wanted these features, we did not have enough time to include them.

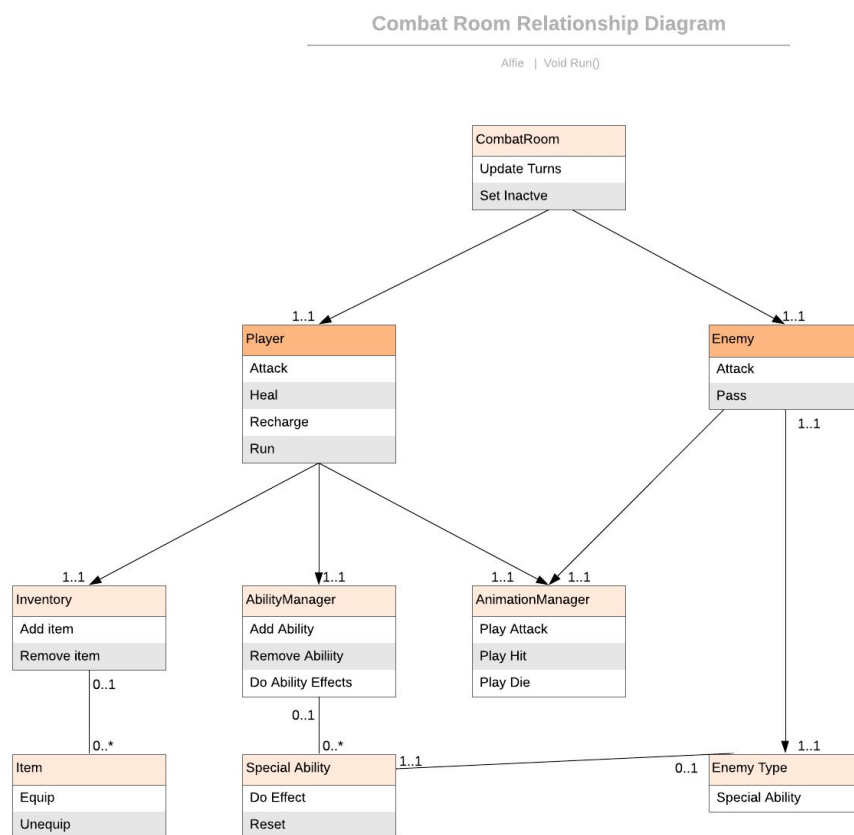
The only major gameplay feature that was changed was a decision we made relatively late into development, which was to use an Action Point system and assign costs to the player's different abilities, instead of the cooldown based system we originally planned. This is because due to how short some of the combats could become, having cooldown-based abilities meant that they would either have to carry over into other rooms, or be once per

combat things. Adding Ability Points to the game made for another level of balancing for the players and we believe improved the game. We also did not include the ability to have consumable items, due to time constraints.

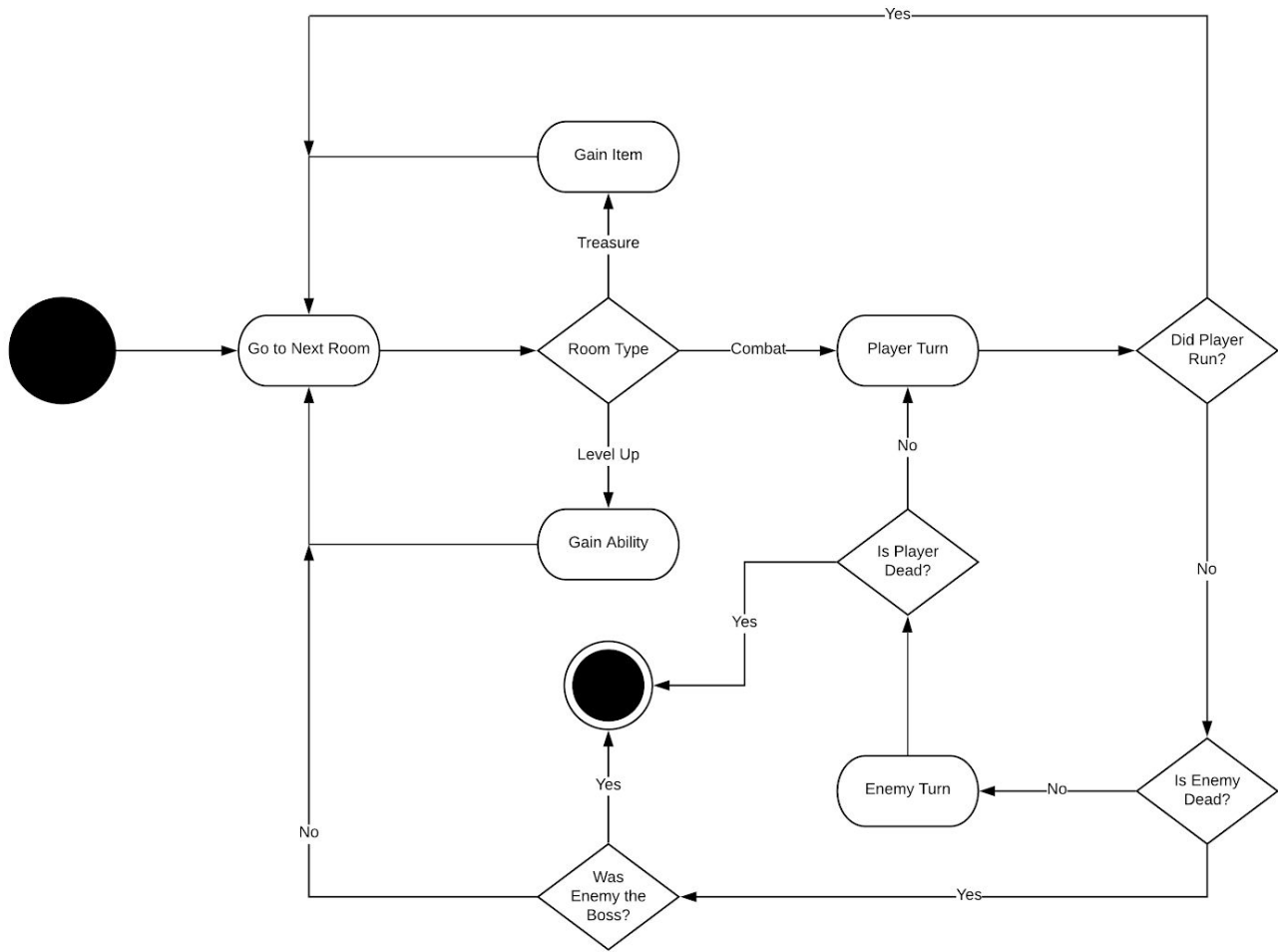
We had minor user interface changes, like moving the locations of a few things compared to the screen mockup in the game design document. Notably the bottom part, where we initially placed the inventory and character details on one side of the action bar, and the map and options on the other. Instead, we placed the action bar in the centre, and relegated the option button to one corner, then placed the player details on the left side of the screen with the items on the right.

<https://alfiethomasson.itch.io/void-run>

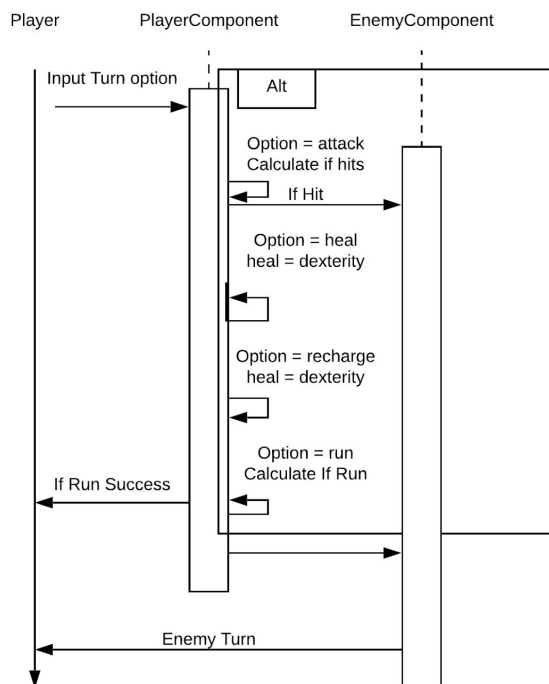
Software Design Models



Relationships of entities in combat rooms.



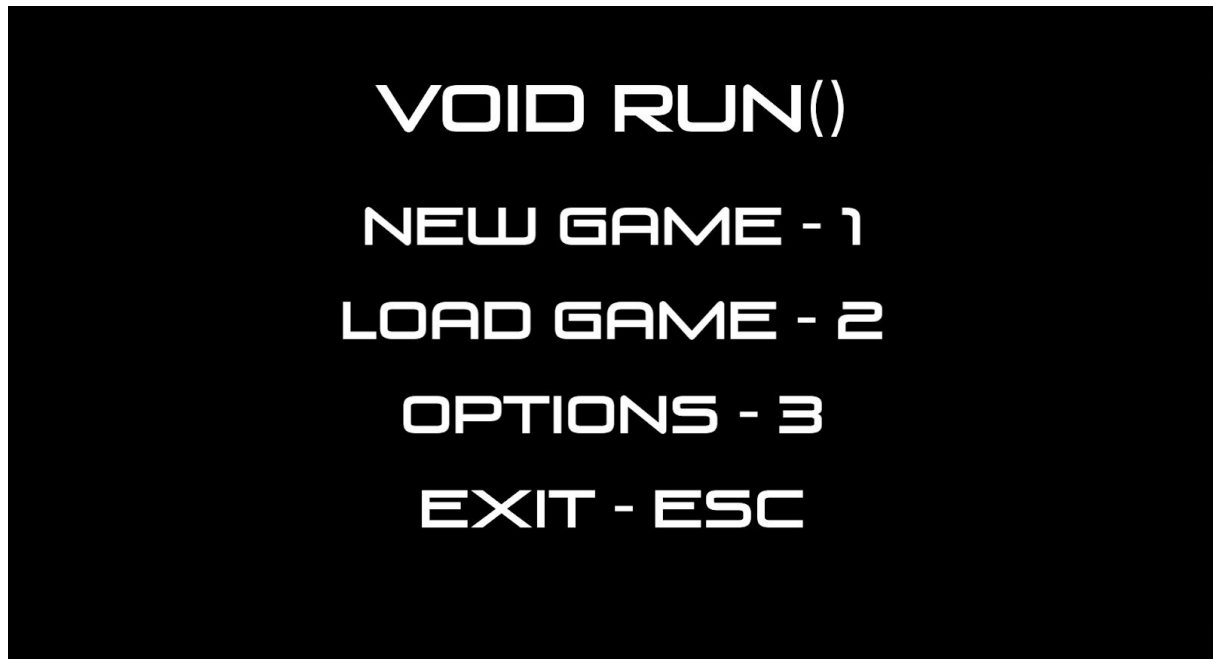
Flowchart showing combat loop



Sequence Diagram for player input

Game Implementation

When the game first starts, the main menu screen opens.



The menu screen has buttons to start a new game, load a save, or open options. The options menu displays accessibility options. The player can change the resolution between 720p, 1080p, and fullscreen. They can control the volume of music and sound effects, or use a Master Volume to alter both. The player can rebind controls by hovering the mouse over one and pressing a key. The Reset Controls button resets the control scheme.



If the player loads the game, it checks for a text file with the appropriate name and pulls the numbers from it. These values are placed into the player's stats, giving the same health, level, items, abilities, etc, as when they left off.

Alternatively, the player can start a new game. Once loaded they are in the core gameplay loop:

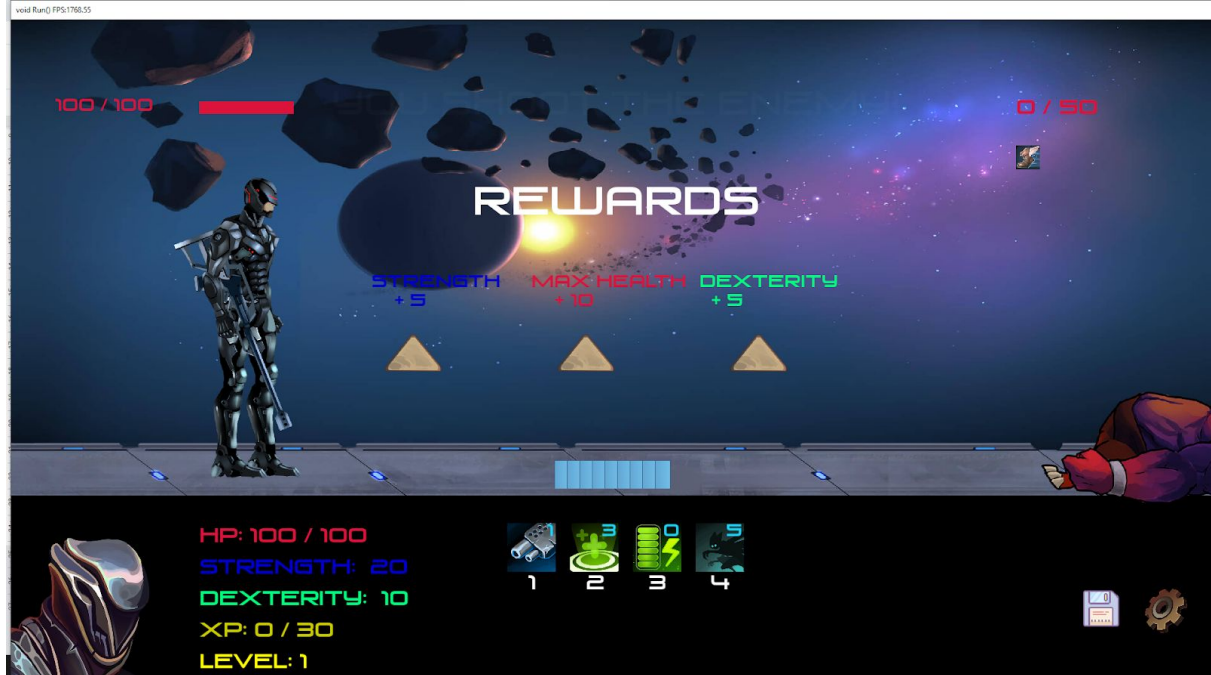


When the player enters a room, an enemy is randomly selected, and a special ability is randomly selected.

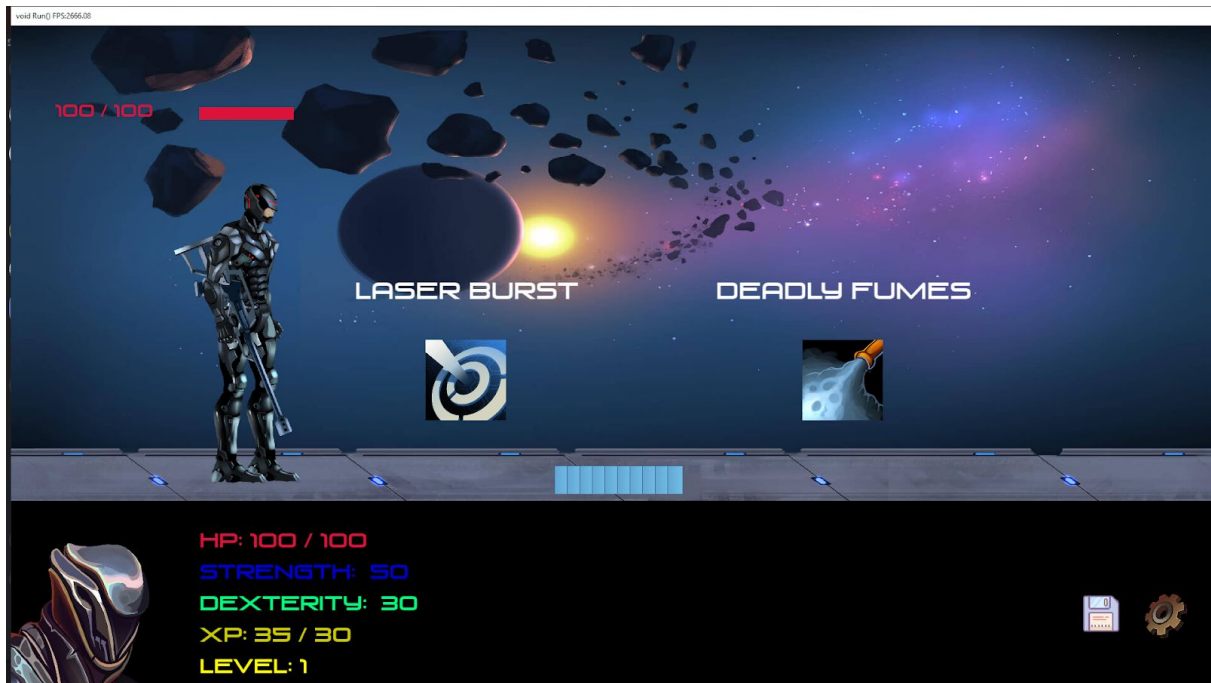
During combat, the player can select from attack, heal, recharge, or flee. There are other abilities available upon level up. Each ability has an AP cost, which the player must pay to use. They regain one point per turn, but can use recharge to gain more.

Attacking performs a calculation to work out their chance to hit based on the player and the enemy's dexterity. Success deals damage equal to the player's Strength. Healing heals by the player's dexterity. Recharging restores 6AP. Running calculates success based on the player and alien's dexterities.

If the player's HP reaches 0 they die, if the Alien reaches 0HP the player gets a stat increase and experience.



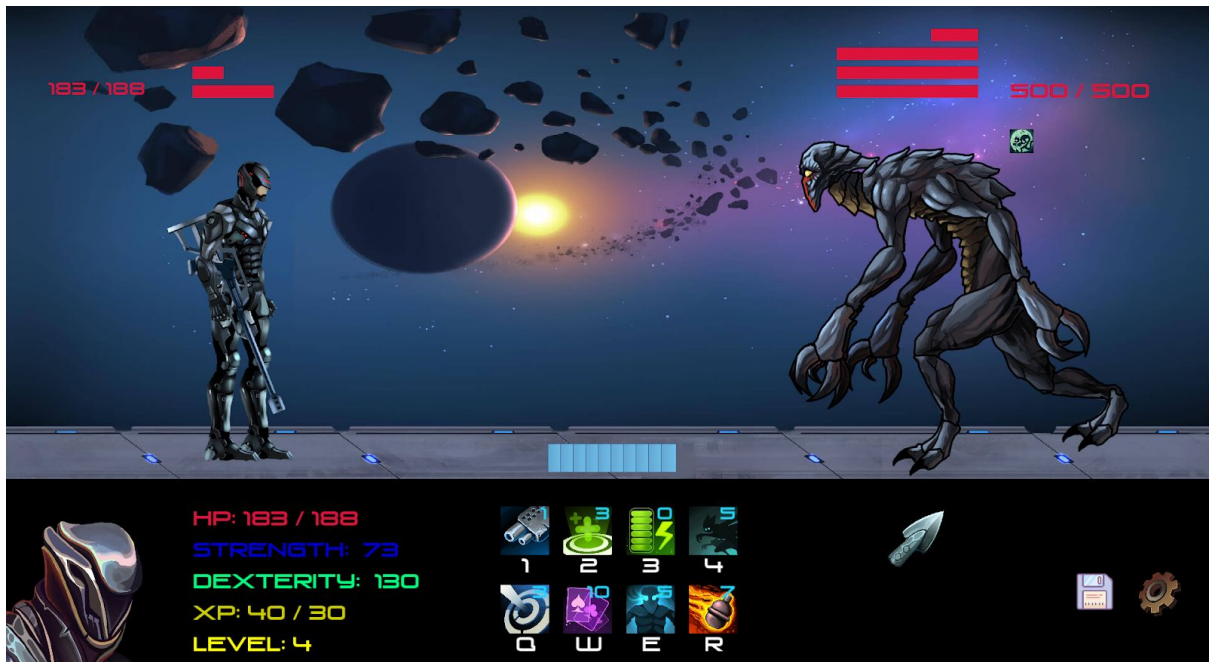
Upon reaching 30 experience, the player levels up and gains a new ability.



There is also a chance of the player entering a treasure room which will randomly give them a new item when opened.



The player can save and open the settings menu from in the game. Once the player reaches level 5, the final boss will attack them.



Upon killing the Alien Lord, the game is won.

Evaluation of implementation

The game meets its original concept very well. We used eerie music and technological sounds for many of the attacks to give the game a creepy, eerie atmosphere. The core gameplay loop is as we intended despite the small changes we made, like trading cooldowns for action points - The game requires a decent amount of thought, especially at the start when you are weak, whilst as you gain more abilities, you have to manage your AP more, giving the game a skill requirement. The game is fun and with the different options of abilities presented, the player feels like they have some degree of control over the situation.

In comparison to other games in the genre, it's clear ours draws inspiration from games like Slay the Spire, and the combat from games like Pokemon and Paper Mario. For the combat, we have captured Pokemon's element of the back and forth moves, and have to choose different options to maximise your potential, quite well. The atmosphere is most similar to Darkest Dungeon, with that constant feeling of not knowing what's coming up, and the constant danger of death. If we were to continue on and improve the game, this is the type of game Void Run() would most likely become.

The game is, overall, of good quality and plays well, but lacks polish. There are minor issues that need to be addressed: Small visual bugs, like the experience counter not updating until after the first turn of combat after levelling up; Minor AI issues, like the Alien Lord continuing to be passive when it should have been desperate; Problems relating to balance where you can randomly spawn an incredibly strong enemy on the first room and die near instantly due to a lack of scaling difficulty.

However, despite this the core functionality of the game works very well, the player is responsive and the keys work well, all of the important functions such as the accessibility features, changing controls, volume, etc work consistently with no problems, and the game has a variety of options for the player to play with, making for a genuinely clean and enjoyable experience with no game-breaking issues.

Possibly improvements, aside from fixing the few minor bugs that remain, would be adding a wider variety of enemies and room types, as well as scaling difficulty. Adding some more abilities or items would be fun too, as there is already a framework in place for items adding special abilities, yet it is not used anywhere. Having trapped rooms or non-hostile encounters could be fun, and it would be nice to have a few more story details, such as a small cutscene showing the alien's death.

Resources/References

Font - <http://typodermicfonts.com/venus-rising-font/>

Characters and Misc Assets -

<https://graphicriver.net/item/complete-futuristic-game-assets-pack-aliens-robots-zombies/17428604?#>

Icons and other Assets -

<https://assetstore.unity.com/packages/2d/gui/icons/spellbook-megapack-109615>

<https://assetstore.unity.com/packages/2d/gui/icons/gui-megapack-101517>

Save Icon - <https://icons8.com/icons/set/save>

Music and SFX - <https://www.zapsplat.com/>