

CISC 226

Game Design Project Proposal

Group 16

January 25th, 2021

Joshua

Kennan

Kevin

Table of Contents

[1.0 Introduction](#)

[2.0 Gameplay](#)

[2.1 Broad Goals and Activities:](#)

[2.2 User Interface](#)

[2.2.1 Example Pixel Art](#)

[2.3 Towers](#)

[2.4 Enemies](#)

[2.4.1 Example Pixel Art for Enemies](#)

[2.5 Crafting System](#)

[2.5.1 Example Pixel Art for Resources:](#)

[2.6 Levels](#)

[2.7 Controls](#)

[2.8 Survival](#)

[3.0 Administrative Information](#)

[3.1 Target Platform](#)

[3.2 Development Tools to be Used](#)

[3.3 Group Coordination Plan](#)

1.0 Introduction

The game is a tower-defense game where the player is also a character on the map and has to interact with the towers and the waves in real time.

You see the world from a top-down perspective. Your goal is to stop enemies from traversing through a series of rooms and reaching the end. You have both the ability to fight enemies in hand-to-hand combat, or by setting up turret “towers” to do the work for you. These towers will require maintenance and reloading. Enemies drop different kinds of resources, including raw materials, weapons, and armour, which can be used to construct new towers, repair or upgrade existing towers, or fortify yourself. The player progresses through several levels, each of which consisting of a collection of rooms. Both the player and his friend have health that is depleted by being attacked by enemies, or by letting enemies get to the end room. The player has a certain number of lives or chances, which once depleted results in a game over and the level restarting. Enemies appear in waves, with each level having a predetermined number appearing at displayed intervals. Once all waves have been defeated, the player may move on to the next level, carrying over a small percent of their currently-owned items. Stronger gear and towers appear as the player progresses. The game has no definitive end; each level gets progressively harder and harder.

Our game is a combination of two genres: a dungeon crawler and a tower defence. The player can run around in a dungeon like in a dungeon crawl, but one of their main abilities is to place and maintain towers, which allows the player to play the game like a tower defence. He needs to physically walk up to a tower in order to interact with it.

We want this game to appeal to different game styles. The player could either focus on building and strategically placing towers and having the player be an aid to the towers, or focus on upgrading the player abilities and having the towers support the player. At the beginning of the game, we plan to have some levels that “force” the player to use one playstyle or another, so that the player can have a taste of both and see which one they like better.

2.0 Gameplay

2.1 Broad Goals and Activities:

The game takes place during the medieval period. The main setting of the game is within a pseudo-randomly generated dungeon. You and a friend decide to explore this old dungeon, only to find a blockage in the end of one of the rooms, preventing any further exploration. As your friend begins attempting to free the blockage, you travel back towards the entrance, feeling defeated. Upon reaching the entrance, however, you notice a wave of enemies making their way towards you. Perhaps they were angered by your intrusion?

The only option is to fight - you must protect your friend in hopes he'll clear the blockage and allow you to escape further. Using any spare materials from within the room, you craft some crude weapons to defend yourself. While you're preparing, your friend discovers some building plans left there from previous explorers. This allows you to build powerful towers that may aid your fight, as well as armour and weapons to help you defend yourself and your friend.

After several waves of enemies, your friend manages to get through the blockage, and you progress to another set of rooms. Before leaving, you were able to salvage a few materials from any towers you had built, which will help you in the next few rooms.

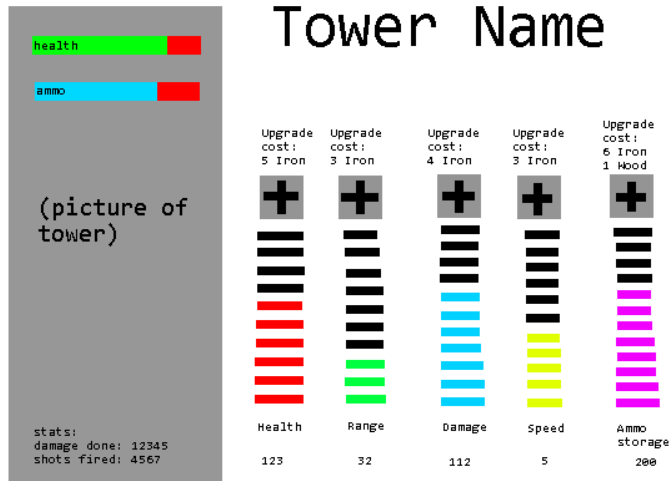
Your goal is to continue through these sets of rooms, learning the secrets the dungeon hides, getting stronger weapons, and ultimately looking for an escape. You and your friend may only lose your health a total of two times before its lights out.

2.2 User Interface

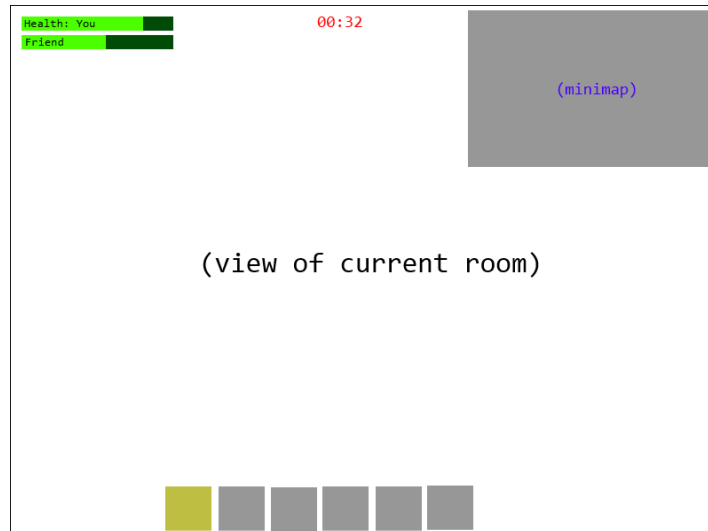
The game is a top-down, 2D game. The game will be pixelated in nature, but at a higher resolution. The music and sound will be in the style of the SNES and other 16-bit consoles.

The camera follows the player around and shows an area that is big enough to contain the current room and part of the surrounding room. The player has health indicators for both them and their friend. At the bottom, there is a hotbar containing towers the player can place and items they can use. Hovering the mouse over a tower gives some information like it's range (represented by a circle around the tower), and how much ammo it has left. A timer will occasionally be shown, which represents how much time is left until another wave approaches.

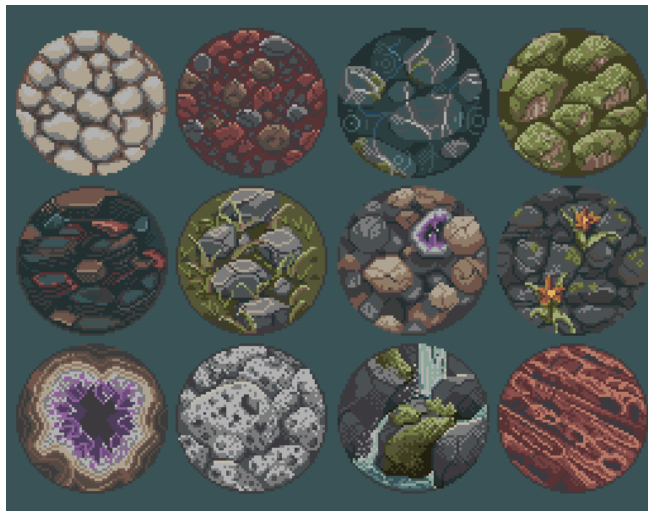
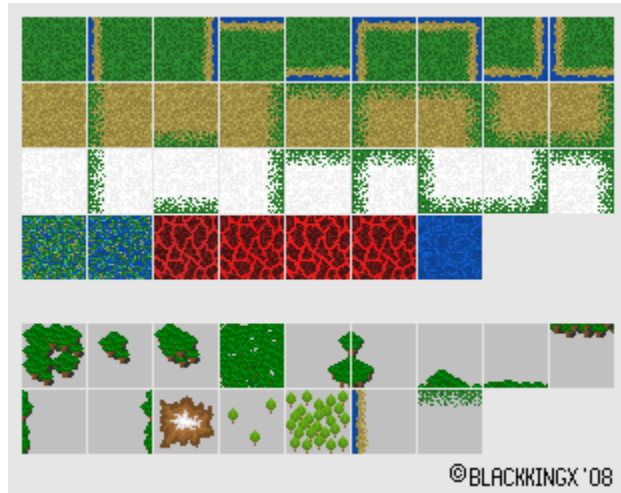
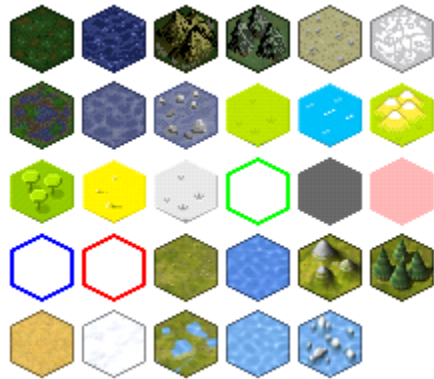
Also shown is a minimap, which gives a general view of the whole map. It shows which room the player is in, which room your towers are in, which are damaged, and which room enemies are in. The player can interface with the minimap and select rooms to view them as if he was there. Here is an example interface for the towers:



Example of general user interface:



2.2.1 Example Pixel Art



Note: These illustrations are examples from Google of the types of designs we are aiming to use for the game - these are not the finale pictures.

2.3 Towers

Towers are a key aspect of game play because they are a helpful tool to help you defend yourself against the monsters in the dungeon. The concept behind the towers is you are able to craft, repair, and upgrade your towers based on the resources available to you. The player is able to collect resources that are left in the room or that are dropped by enemies in the waves. As the game progresses different resources will be dropped allowing for different types of towers to be built (fire, tesla, sniper, etc.). Since the player is in the dungeon for them to build, repair, destroy, or upgrade towers they will have to go to the tower's location to complete the action. Players can build or upgrade towers to improve their usefulness but they must also be aware that the towers are susceptible to being attacked by enemies and will need maintenance. Maintenance includes both reloading the towers as well as repairing the towers due to damage

done by enemies. Maintenance can happen during waves or in between waves however, the player must be at the tower's location to repair it to make mid-game maintenance challenging. If a tower is destroyed, it will drop some of its resources which can be picked up by the player. A tower can be destroyed by both enemies in the wave as well as the player which gives them the option to scrap towers for resources. To salvage resources from the towers the player will attack the tower to destroy it themselves.

Some example tower types (note that some of them would be stretch goals):

Name	Cost (rarity of resources required)	Projectile type	Range	Shot speed	Damage per shot	Damage type	Maintenance	Effect
Dart	Low	Basic dart	Medium	Low	Medium	Single	Low	
Rapid	Medium	Small dart	Medium	High	Low	Single	Low	
Fireball	Medium	Fireball	Medium	Medium	Low	Sustained area	High	Sets enemies on fire
Cannon	Medium	Explosive	Low-Medium	Slow	Medium	Area	Medium	
Tesla	High	Lightning	Low	Low	Medium	Sustained area	Medium	Stuns enemies
Flamethrower	Medium	Fire	Low	Medium	Low	Sustained	High	Sets enemies on fire
Sniper	Medium	Bullet	High	Low	High	Single	Medium	

2.4 Enemies

There are many different kinds of enemies, each with different weapons, move speeds, damage, and health. The enemies get progressively harder the higher the level gets. When the player successfully defeats an enemy, it drops resources that can be used to craft items and towers, and also food that heals the player a little bit. Some enemies may be more resistant to certain types of attacks and are best killed with different kinds of attacks, e.g. an ice enemy is resistant to ice, but vulnerable to fire. When an enemy moves, it chooses a target in order of precedence. It only attacks the player when it is within a certain range. If it can't attack the player, it can attack a tower, again only if it's within range. If it can't attack either, then it progresses toward the end of the level until it reaches it or it gets within range of something to attack. One stretch goal would be to have certain enemy attacks cause debuffs to you or the tower, e.g. setting it on fire or poisoning you. Examples of enemy types:

Name	Health	Damage	Attack speed	Movement speed	Attack effect
Swarm	Low	Low	Medium	Medium	

Brute	High	High	Slow	Slow	
Standard	Medium	Medium	Medium	Medium	
Fiery	Low	Low	Medium	Medium	Sets things on fire

2.4.1 Example Pixel Art for Enemies



Note: These illustrations are examples from Google of the types of designs we are aiming to use for the game - these are not the finale pictures.

2.5 Crafting System

Crafting will be an essential aspect of the game to allow the player to improve the battle statistics of their towers as well as their own battle stats. You will acquire resources that are either within the room or that are dropped by enemies within the waves. As the game progresses the player will get access to different types of resources which will allow them to craft new items or upgrades. New items and upgrades will be pointed out at each relevant level by your buddy. The only requirement to craft an item will be to have the resources; clicking the button will craft the item in an instant. In order to craft, the player will select the craft menu, which will always be accessible, which will bring up the items list and all items that can be crafted will be highlighted. Besides acquiring resources from drops or destroyed towers, you can destroy an item to salvage its resources. To destroy an item you simply open inventory and then select one of your current items and select the scrap option. It is important to note that resources are used for both towers and personal improvement; this means the player has to make a decision on their play style: use resources to improve their own stats or their towers stats. This dilemma will prompt the player to actively make decisions and choose between playstyles. Within the current level the player will have unlimited space in their inventory. Example resources (note that some of them would coincide with stretch goals):

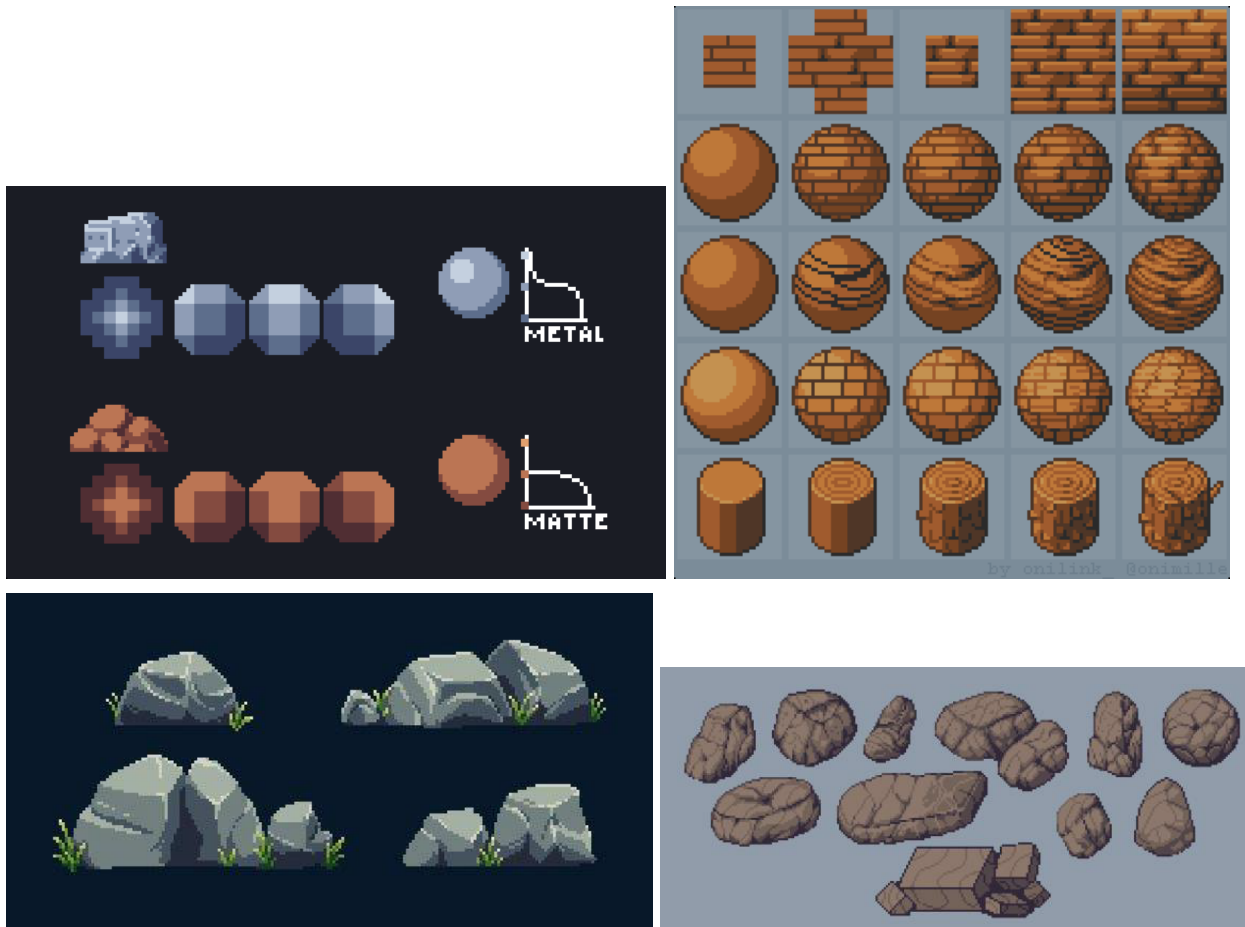
Name (not finalized)	Rarity (probably will decrease with progress)	Use
Wood	Very common	Weaker/cheaper items
Iron	Less common	Fairly strong items
Tungsten	Rare	Fire-resistant items, towers that shoot fire
Cobalt	Rare	Items/towers involving ice
Gunpowder	Uncommon	Items/towers that involve explosives
Uranium	Uncommon	Explosive-resistant armor, towers that involve explosives
Copper	Uncommon	Towers/items involving lightning/electricity
Titanium	Rare	Very strong items

Weapon examples (note that some of them would be stretch goals):

Name	Damage	Attack speed	Effect
Wood pole	Low	Medium	
Iron sword	Low-medium	Medium	

Firey iron sword	Low-medium	Medium	Sets enemies on fire
Simple bow	Low	Low	
Tungesten sword	Medium-high	Medium	

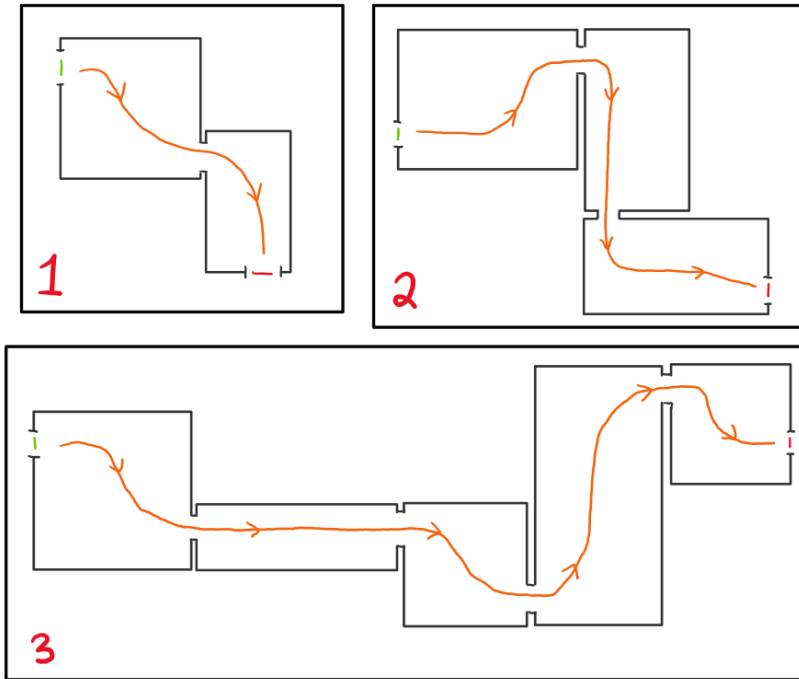
2.5.1 Example Pixel Art for Resources:



Note: These illustrations are examples from Google of the types of designs we are aiming to use for the game - these are not the finale pictures.

2.6 Levels

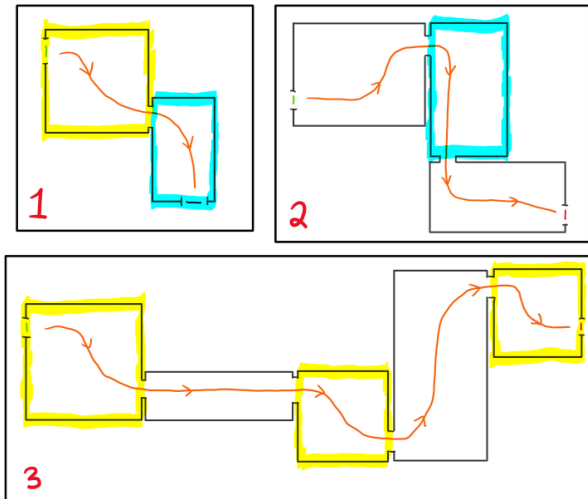
The game has infinite levels, each consisting of an ever-increasing series of connected rooms (see figure below).



Each room will have some debris scattered within them, some of which may be able to be broken down into much needed construction materials. The player has a short duration of time, displayed on the screen, to set up their towers and craft any equipment they see fit before enemies begin showing up, travelling along a linear path to the final room's exit. A time will appear at the end of every wave indicating how long the player has to prepare for the next wave. This could include crafting better gear and weapons, making new towers, or managing, upgrading or repairing existing towers. Each wave will be more difficult than the last, with enemies increasing in number, strength, or both. Once all the waves of a level have been defeated, a final timer is displayed indicating the amount of time the player has to leave the level. In this time the player may destroy their constructed towers and equipment for scraps, some of which will be transferred over to the next level, giving them a slight starting advantage. The player is given some lore or information by the friend at the start of each level. The level designs change as the player progresses; each successive level having more rooms, each stylized to represent the area of the dungeon they are in. The player may replay any level they have beaten before without having to replay the previous levels, as well as starting on the current level they have yet to beat.

The first few levels will be manually designed, and are more akin to tutorials. They will introduce the narrative to the player, as well as the mechanics and available playstyles. One level may lack any weapons for the player, strongly encouraging them to make use of towers. Another level may lack many resources needed for towers, requiring the player to use hand-to-hand combat. The goal of this would be to organically demonstrate the strengths and weaknesses of each fighting method, allowing the player to better choose tactics in the future. There must be some balance between playstyles; the player would likely get overwhelmed without the use of towers, and yet still has to physically engage with the battle by maintaining, reloading, and upgrading towers.

Levels after the first few will be generated using prefabricated rooms, with randomized debris generation in each. Several generic room “pieces” will be created, and can be put together like puzzle pieces to form a level. Each room will have unique randomized debris and decor, helping to make each room seem unique. This concept allows for a nearly infinitely scalable level generator. The figure below highlights the re-use of prefabricated rooms.



2.7 Controls

For controls, we ideally would want to have both keyboard and controller support. Movement can be done using a joystick or WASD. Attacks would be one button (mouse button or A), interfacing with towers would be another button (E on keyboard or B), opening the crafting interface would be a third button, and building would be a fourth button. The hotbar can be scrolled through using number keys and the scroll wheel on the keyboard, and using bumpers on controllers. To prevent accidental destruction of towers, there is an extra button to enable/disable tower destruction, although when the level ends and you are salvaging your towers, this is enabled by default.

2.8 Survival

To complete the level, you must survive the waves of enemies without letting too many enemies get through or being killed by enemies yourself. There are two health bars that show how much damage has been done to you and your friend. Damage is done to your friend when the enemies get through the level. You have two “lives” that are shared between you and your friend. When one health bar reaches zero, a life is used to revive that health bar. Running out of health with no lives remaining means it’s game over!

Players can carry multiple types of weapons, both melee (e.g. swords, daggers) and ranged (e.g. bow, crossbow). Players can also wear armor which protects the player from general

attacks to a certain extent. One stretch goal would be to have some types of armour be able to deflect certain types of attack, like fire attacks or lightning attacks. Another stretch goal would be to implement items such as potions that grant the player buffs, like a shield or greater speed.

3.0 Administrative Information

3.1 Target Platform

Our target platform will be WebGL to run on a browser.

3.2 Development Tools to be Used

The game engine we will be using is Unity Game Engine running C# code. For code editing, we will be using VSCode. For 2D assets, we will be using Paint.NET and/or GIMP. We will use Tracktion Waveform DAW and Audacity to create and edit music and sound effects. We will be using Github Classroom for project management.

3.3 Group Coordination Plan

Our group will be using Trello as our project management application, Discord for daily group communication, and we will have a weekly group meeting on Monday at 1:30pm - 2:30pm and Fridays at 2:00pm - 3:00pm. Trello will be used to log and assign tasks, keep track of each team member's workload, and plan the group's progress towards milestones. Discord will be our primary communication tool for all messaging and calling needs. As for our weekly calls Monday's will be dedicated to planning the current week by assigning tasks to each team member. Friday calls will then give us the opportunity to connect at the end of the week, discuss any issues anyone had, and be on the same page going into the weekend.