



**FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University**

BACHELOR THESIS

Name Surname

Thesis title

Name of the department

Supervisor of the bachelor thesis: Supervisor's Name

Study programme: study programme

Study branch: study branch

Prague YEAR

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date
Author's signature

Dedication.

Title: Thesis title

Author: Name Surname

Department: Name of the department

Supervisor: Supervisor's Name, department

Abstract: Abstract.

Keywords: key words

Contents

1	Introduction	4
1.1	Game Genre	4
1.1.1	Strategy	4
1.1.2	Tower Defense	4
1.1.3	Rogue-like	5
1.1.4	Combining the Two	5
1.2	Original Vision	6
1.3	Current Scope and Goals	7
2	Game Design	8
2.1	Design goals	8
2.1.1	Slay the Spire	8
2.1.2	Plants vs. Zombies	8
2.1.3	Strategic Depth in Each Battle	8
2.1.4	Strategic Depth in Each Run	9
2.1.5	Make Various Strategies Viable	9
2.1.6	Force Exploration	9
2.1.7	Provide a Challenge	10
2.2	Battle	10
2.3	Attacker Movement	10
2.4	World	12
2.5	Resources	12
2.5.1	Materials	12
2.5.2	Energy	12
2.5.3	Hull	12
2.6	Graphical User Interface	12
2.6.1	Waves Left and Fuel	13
2.6.2	Hull	13
2.6.3	Wave Preview	13
2.6.4	Materials and Energy	13
2.6.5	Blueprints	13
2.6.6	Info Panel and Selection	13
2.6.7	Highlights and Range Visualization	13
2.7	Attackers	13
2.8	Buildings	14
2.8.1	Economic buildings	14
2.8.2	Towers	14
2.9	Abilities	14
2.10	Camera controls	14
2.11	Future Features	15
3	Analysis	17
3.1	Game Engine	17
3.2	Procedural level generation	17
3.3	Random number generation	17

3.4	Path generation	17
3.4.1	Path visualization	17
3.5	Terrain generation	17
3.6	Resources and obstacles	18
3.7	Terrain types	18
3.8	World builder	18
3.9	Attacker wave generation	18
3.10	Simulation	18
3.11	Visuals and interpolation	18
3.12	Attacker targeting	18
3.13	Game events	18
3.14	Range visualization	18
3.15	Selection	18
3.16	Blueprints	18
3.16.1	Dynamic descriptions	18
3.16.2	Attacker stats	18
4	Developer Documentation	19
4.1	Unity	19
4.2	Scenes	19
4.2.1	Battle	19
4.2.2	Loading	19
4.2.3	Main Menu	19
4.3	???	19
5	User Documentation — Designer	20
5.1	Terrain Types	20
5.2	Blueprints	20
5.2.1	Buildings	20
5.2.2	Towers	20
5.2.3	Abilities	20
5.3	Attackers	20
5.4	???	20
6	User Documentation — Player	21
6.1	?Introduction	21
6.2	?Controls	21
6.3	?Mechanics	21
7	?Playtesting	22
	Conclusion	23
7.1	Goals	23
7.2	Future Work	23
	Bibliography	24
	List of Figures	25
	List of Tables	26

List of Abbreviations	27
A Attachments	28
A.1 First Attachment	28

1. Introduction

Games exist. They are good. I like games. I will make a game.

1.1 Game Genre

There are a plethora of games. Each is unique in its own way, but there are many similarities among them. One of the ways to categorize games is by their genre. A genre can encompass many characteristics of a game, most often its mechanics, but also its theme, art style or the medium it is played on. Genres have no exact definitions or strict boundaries and similarly, any individual game is usually a mix of different genres.

1.1.1 Strategy

One major genre is strategy games. Strategy games focus on tactics and long-term planning. They require a lot of thinking. There are various kinds of strategy games, but most often, players compete against each other to reach some goal. These players can be real humans or artificial intelligence agents. Strategy games often utilize hidden information or rely on the unpredictability of other players' actions to create an environment where there is no single best way to reach the goal. This means that players have to have a good understanding of the game and be able to adapt to the situation at hand.

There are many qualities players enjoy in strategy games. Of course, it feels great to outsmart your opponent or conquer a challenge. But strategy games also provide a sense of progression and accomplishment because of their depth. They are often very replayable because of the different situations that can arise from the game's mechanics.

One way to categorize a strategy game is whether it is real-time or turn-based. In a turn-based strategy, players take turns to make their moves. This allows for a slower pace and more time to think about the best move. In real-time strategy games, however, the environment evolves continuously and players have to react and make decisions quickly.

1.1.2 Tower Defense

Tower defense is a subgenre of real-time strategy where the player has to defend against waves of enemies. The player has to build towers, which attack the enemies as they approach, to defend their base. The attackers are very predictable and usually follow a set path. The player has to build their towers in a way that maximizes their effectiveness against the attackers. (do I want images and examples?)

The game usually consists of multiple levels, each presenting a different challenge. The player has to adapt to the different attackers and the different terrain. Sometimes the levels make up a campaign, where the player has to progress through harder and harder levels to reach the end. Other times, the levels are standalone and the player can choose which level to play, where the goal can be

to survive as long as possible. This can take a few hours and may become very repetitive.

The attackers can have unique abilities or be resistant to certain types of towers. The composition of each wave can be predetermined or randomized to a varying degree. In this way, the game can force the player to adapt and use different strategies. To make their decisions interesting, the player has various towers and upgrades at their disposal, each with different abilities, often complimenting each other.

Players can get resources to build their defense passively, but some tower defense games also include an economy system. Here, the player has to build economic buildings to generate the resources. This adds another layer of strategy to the game, as the player has to balance their economy with their defense.

1.1.3 Rogue-like

Rogue-like is a subgenre of role-playing games. In role-playing games, the player takes on the role of a character and goes on an adventure. The character can grow stronger by acquiring new abilities, items or experiences. The player has to make decisions about how to upgrade their characters to overcome the challenges they might face. This high level of customization and the sense of progression makes role-playing games very engaging.

Rogue-like is named after the game Rogue, released in 1980([citation!?](#)). The game was known for its high difficulty and the fact that the player had to start from the beginning if they died. The game is usually procedurally generated, meaning that the levels are created by an algorithm, rather than being designed by a human. This is important because the player can't memorize the levels and has to rely on their skill and knowledge of the game's mechanics to progress. Since all in-game progress is lost when the player dies, the player has to learn from their mistakes and improve their skill to advance further. These are defining characteristics of rogue-likes that separate them from other role-playing games.

In most rogue-like games, you explore a dungeon, where you have to fight enemies and avoid traps. You are then rewarded with items and other resources to improve your character. The game is usually turn-based, making it more strategic.

Many games break the traditional rogue-like formula but still share many of its characteristics. They might have a different theme, where you don't even fight monsters but instead compete in another task. They might include a progression system, where you unlock new items or upgrades for your future runs. Real-time gameplay is also common, where more skills are tested than just the player's ability to think strategically. These games are often called rogue-lites, however, we will not make such a distinction since all genres can be bent and blended in many ways. ([do I include more examples here?](#))

1.1.4 Combining the Two

I enjoy playing both rogue-likes and tower defense games a lot([should this be here?!](#)). There aren't many games that combine these two genres. It is possible this combination doesn't work well, however, it is worth exploring, because these

two genres seem to complement each other. This is why this game will be a blend of these two genres.

The main gameplay loop will be a tower defense game. The player will play through many short levels. Tower defense can get a bit stale if you find a strategy that can be used every time. This is how some rogue-like elements can help. Each level will be procedurally generated featuring different terrain and attackers. Additionally, the player will start with a small arsenal of blueprints — defensive towers, abilities and economic buildings. They will acquire new ones from a randomized selection as they progress. This means that the player will likely use a different strategy in each run.

1.2 Original Vision

The game will be for the PC. This is because it is going to be pretty complex and PCs usually have screens large enough to display all the information the player needs. The mouse and keyboard allow for precise control and a lot of buttons to be used. PC players are also more likely to enjoy challenging strategy games, which is what this game aims to be.

It will be a single-player game. As stated, the small-scale gameplay will be a tower defense, but on a larger scale, the game will be rogue-like. This means that it will consist of individual runs, where the player will start from scratch every time. Their goal is to get as far as possible, trying to reach the final level and beat the game.

The game should have some story to motivate the player on their journey. The details of this story are not yet decided, but the game will likely be set in space. The player will travel forward through an unexplored galaxy trying to reach a destination specified by the story. The levels will take place on different planets, each with its own theme and unique terrain. This sci-fi theme allows for a lot of freedom in the designs of the levels, buildings and attackers. It also works well with the mechanics of the individual levels or, as we will call them "battles".

The goal of a single battle is to gather enough fuel to continue to the next planet. The faster the player gathers the fuel, the sooner they win the battle. Fuel is gathered passively, but additional buildings can be built to speed up the process. In the meantime, the player has to defend against waves of attackers by building towers and using abilities. All of this costs materials and energy — resources, which are generated by economic buildings. Thus, the player will have to balance their use of resources among defense, economy and gathering fuel.

On their way, the player will encounter various events, shops and other anomalies. These will provide the player with opportunities to gain resources, blueprints or other bonuses. They will also provide the player with choices, which can have different outcomes. The player will have to decide which choice is the best for their current situation. Some information about the upcoming battles will be given to the player, for example, the attacker types and general difficulty of the battle. Harder battles will provide better rewards. The path the player takes will be non-linear allowing them to decide which battles to fight and what to interact with. This way they can choose to take a bigger risk for a bigger reward or play it safe.

I probably forgot to mention something

1.3 Current Scope and Goals

The scale of the full game is quite large. Too much for the scope of a bachelor's thesis. Furthermore, we will not consider the story, art or sound design as that is not our area of expertise. Instead, the goal will be to create a functional prototype, which can be used to playtest the core gameplay. It will require some basic content — several attackers and blueprints — so the game can be properly tested. The prototype will be prepared for future development so that more content can be added later.

Playtesting is very important because it can provide valuable feedback about the game's mechanics. This prototype will allow for testing the game and adjusting the design, tweaking the mechanics or making things more clear to the player. It is important to start with playtesting as soon as possible to avoid wasting time on content that doesn't work.

The prototype will consist of all systems and mechanics necessary to play through a battle. This includes attackers, towers, abilities and the economy. It will also include procedural generation of the battles including the terrain, attacker paths and the makeup of the attacker waves. The player will be able to progress through battles and collect blueprints. But there will be no map view to choose their path because the progression will be linear. There will also be no events or shops.

All the art and sound assets will be placeholders. But care will be taken to make everything as clear as possible to the player.

Additionally, the prototype will include a very rudimentary tutorial to explain the game's mechanics to the player. This is important because the game will be complex and the player needs to understand how the game works to be able to play it properly.

The main goal of this thesis is to design, and implement this prototype of the game, so it can be playtested. It will also focus on the design of the game's mechanics and systems, and the decisions behind them. The game will be implemented in the Unity game engine, using C# as the programming language. The approaches to implementing the game's mechanics and systems will be discussed in detail. Notably, the algorithms that were used and why they were chosen.

2. Game Design

don't just document the design, but also the decisions behind it

- We want the game to be good, so we should think about the things that will make it good. It is for a specific audience.

2.1 Design goals

- these are the goals the game should achieve (each will have explanation — what and why)

1. strategic depth in each battle
2. strategic depth in each run
3. make various strategies viable
4. force exploration
5. provide a challenge

- some of our goals are achieved very well in other games of the relevant genres, so we can look at them for inspiration

2.1.1 Slay the Spire

- explain the game, use the following images



Figure 2.1: A fight in Slay the Spire.

2.1.2 Plants vs. Zombies

- explain the game, use the following images

2.1.3 Strategic Depth in Each Battle

- StS — you have to balance defense and offense — some enemies get stronger throughout the fight - refer to the fight image



Figure 2.2: Slay the Spire — map screen.

- PvZ — you have to balance defense and economy — in each level, you want plants that help you in the short term while you build your economy and plants which are more expensive, but effective once you can afford them
- in our game you have fuel mining as offense and defensive towers as defense, economic buildings and abilities, so you have to balance short-term and long-term power

2.1.4 Strategic Depth in Each Run

- StS — As you get better, you realize there is this interesting trade-off between short-term and long-term power - you want cards to survive the next few fights, but they will be duds later; you also want cards which will have greater synergy in the future, but don't help you right now.

- examples - iron wave, entrench

2.1.5 Make Various Strategies Viable

- StS — some cards interact in interesting ways that make them stronger ; there are cards and relics which fundamentally change the way your deck works (ex. barricade and entrench)
- PvZ — the interactions are not so strong, but you have to combine the plants such that you have no weak spots — cheap and expensive, for specific zombie types
- make blueprints that have unique effects that change the way you play

2.1.6 Force Exploration

- StS — you have to explore different strategies because you get different cards every time, some enemies ensure you are prepared for everything and intentionally break some strategies
- characters



Figure 2.3: A plants vs. Zombies level.

- PvZ — you have to explore different strategies because you have to adapt to different zombies and level environments, not too deep; after the campaign, three seed slots are selected for you.
- choose from random blueprints, various attackers with abilities and terrain types

2.1.7 Provide a Challenge

- StS — not easy to beat; after you beat the game, you can play on a higher ascension
- we will have something similar to StS

2.2 Battle

- overview viz section 1.2
- build stuff, send waves, use abilities

2.3 Attacker Movement

- free movement with individual pathfinding X
- too unpredictable for the player — The player needs to plan in advance in order to maximize the possibility to take calculated risks - solution: visualize enemy paths — Too many different paths - too much clutter; Still planning at most one wave in advance
- linear lanes (like in Plants vs. Zombies) X - not enough space for interesting building placement



Figure 2.4: Sunflowers and Potato Mines in Plants vs. Zombies. Sunflowers are necessary to fuel your economy, while Potato Mines are a cheap way to deal with the first few zombies.



Figure 2.5: Iron Wave — a card in Slay the Spire.



Figure 2.6: Entrench — a card in Slay the Spire.



Figure 2.7: A small portion of the many interesting cards in Slay the Spire.

- paths based on your buildings X - what if they block off the path?
- predefined paths ✓
- branching

2.4 World

- free placement X
- grid of tiles ✓ - more granularity, easier to develop intuition, same for attacker paths
 - hexagons X
 - squares ✓
 - 3D ✓ - Simple and intuitive way to make the level itself more interesting Some towers won't be able to shoot uphill or downhill - more interesting decisions for the player
 - Terrain types
 - Obstacles

2.5 Resources

2.5.1 Materials

2.5.2 Energy

2.5.3 Hull

2.6 Graphical User Interface

specify controls



Figure 2.8: All the plants of Plants vs. Zombies in the in-game almanac.

2.6.1 Waves Left and Fuel

2.6.2 Hull

2.6.3 Wave Preview

2.6.4 Materials and Energy

2.6.5 Blueprints

- what they represent
 - rarities
 - make them unique
 - lenticular design

2.6.6 Info Panel and Selection

- what it looks like and what's on it
 - select blueprints
 - select buildings
 - select attackers

2.6.7 Highlights and Range Visualization

2.7 Attackers

- move, have health
 - sizes
 - special abilities - passive, repeating, reactive



Figure 2.9: Card reward screen in Slay the Spire. Here the player can choose one of three randomly selected cards to add to their deck.

2.8 Buildings

- how and when to build, one per tile
 - special building have special abilities
 - main types:

2.8.1 Economic buildings

- provide resources

2.8.2 Towers

- attack attackers
 - range
 - targeting
 - projectile types
 - damage types

2.9 Abilities

- used mid-wave
 - usually instant effects
 - free placement, global placement, tile placement, use on a building

2.10 Camera controls

- zoom to look closely, rotate so the terrain doesn't hide stuff



Figure 2.10: Seed select screen in a rooftop level. Here, plants must be planted in flower pots, and most plants cannot shoot uphill.

2.11 Future Features

- run structure
- map
- events and shops
- difficulty levels
- unlocks

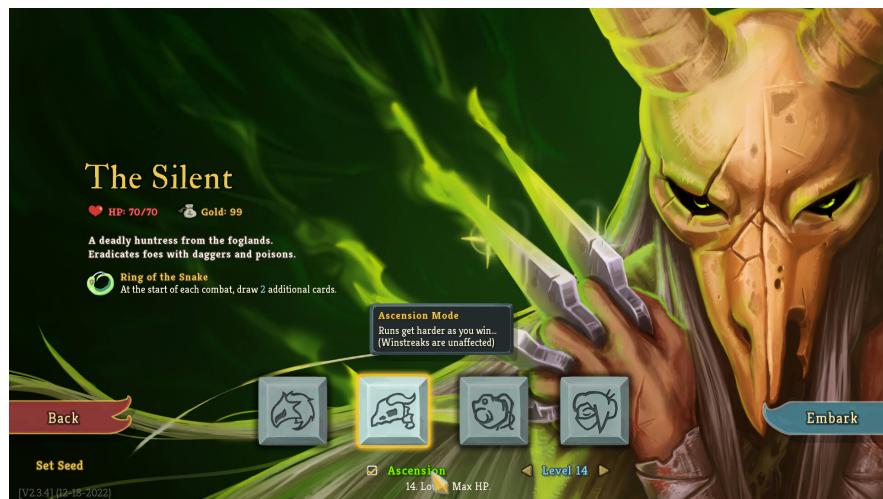


Figure 2.11: Character select screen in Slay the Spire. Here the player can also choose their ascension level, making the game harder.

3. Analysis

3.1 Game Engine

- unity because of familiarity

3.2 Procedural level generation

- make each level unique
 - one random seed dictates entire campaign

3.3 Random number generation

- unity random X
 - system random X
 - custom random ✓
 - why LCG

3.4 Path generation

- describe the algorithm here

3.4.1 Path visualization

- line renderer ✓
 - why does it look this way

3.5 Terrain generation

- fractal noise X
 - WFC ✓

- 3.6 Resources and obstacles**
- 3.7 Terrain types**
- 3.8 World builder**
- 3.9 Attacker wave generation**
- 3.10 Simulation**
- 3.11 Visuals and interpolation**
- 3.12 Attacker targeting**
- 3.13 Game events**
- 3.14 Range visualization**
- 3.15 Selection**
- 3.16 Blueprints**
 - 3.16.1 Dynamic descriptions**
 - 3.16.2 Attacker stats**

4. Developer Documentation

4.1 Unity

4.2 Scenes

4.2.1 Battle

4.2.2 Loading

4.2.3 Main Menu

4.3 ???

5. User Documentation — Designer

5.1 Terrain Types

5.2 Blueprints

5.2.1 Buildings

5.2.2 Towers

5.2.3 Abilities

5.3 Attackers

5.4 ???

6. User Documentation — Player

6.1 ?Introduction

6.2 ?Controls

6.3 ?Mechanics

7. ?Playtesting

Conclusion

7.1 Goals

7.2 Future Work

Bibliography

List of Figures

2.1	A fight in Slay the Spire.	8
2.2	Slay the Spire — map screen.	9
2.3	A plants vs. Zombies level.	10
2.4	Sunflowers and Potato Mines in Plants vs. Zombies. Sunflowers are necessary to fuel your economy, while Potato Mines are a cheap way to deal with the first few zombies.	11
2.5	Iron Wave — a card in Slay the Spire.	11
2.6	Entrench — a card in Slay the Spire.	11
2.7	A small portion of the many interesting cards in Slay the Spire.	12
2.8	All the plants of Plants vs. Zombies in the in-game almanac.	13
2.9	Card reward screen in Slay the Spire. Here the player can choose one of three randomly selected cards to add to their deck.	14
2.10	Seed select screen in a rooftop level. Here, plants must be planted in flower pots, and most plants cannot shoot uphill.	15
2.11	Character select screen in Slay the Spire. Here the player can also choose their ascension level, making the game harder.	16

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment