Roguelike Development Fundamentals from Scratch

Matilda Lerner

Contents

1 ABSTRACT

My amazing abstract would go here!

2 INTRODUCTION

Every aspect of a game stems from its mechanics, a notion which fascinates me to no end. For this reason, I endeavored to design a complex video game from scratch as my senior thesis, studying and applying various algorithms in order to create the foundation for deep and engaging gameplay.

My decision to apply this learning to a "traditional rogue-like" stems from my longtime fascination with the genre. Multiple of my favorite games can be classified as nothing other than a traditional rogue-like while boasting distinct mechanics which set them apart from any other game, or creative endeavor for that matter. My hope was to understand at least the foundation of how these games operate, applying them to my own attempt at a traditional rogue-like.

What defines a rogue-like is the forcing of a player back to the start of the game whenever they lose, a mechanic called permadeath, a shorthand for permanent death. The earliest rogue-likes strived to capture the essence of tabletop role-playing games by simulating every detail of every game object to create a world deep with possibilities, using permadeath as a means of forming a bond between a player and their character. Contemporary rogue-likes, while maintaining the permadeath qualifier, have foregone complexity in favor of quicker and more reactive mechanics. In other words, contemporary rogue-likes are more akin to their arcade ancestors, while contemporary developers of traditional rogue-likes continue to experiment with deep mechanics (or, in the case of some developers, introducing these deep mechanics to the current generation).

Finding this medium where developers are actively experimenting with the scope of what video games mechanics are capable of is what draws me to traditional rogue likes, and why I wanted to learn how to develop the fundamentals of one. There are of course tools and libraries available to facilitate the development of a rogue like, but I chose to forego them in the hope of learning how these core mechanics function. This was the foundation into which I entered the development of my game, which I have chosen to call Leaflings.

3 BACKGROUND AND RELATED WORK

The concept of a "rogue-like" was defined in the 80s, as many developers were releasing derivatives of the original *Rogue*, released in 1980, due to its appealing flexible nature and open-ended design. The genre still sees refinement and interesting adaptations with many rogue-likes being released by indie developers every year, some of which garner massive acclaim and success.

Contemporary rogue-likes tend to give players an incentive to continue playing between losses by allowing special unlocks, regardless of a round's victory. If a player makes significant progress but dies before reaching the game's end, their progress in that round must be lost (as dictated by rogue-like tradition), but they should still be rewarded for getting far. This reward both incentivizes the player to try again, and it also allows the developers to make the later parts of their game more difficult, with the knowledge that the player can come back with steadily better unlocks. (*Risk of Rain*)

Contemporary developers of traditional rogue-likes tend to adhere to the archaic fundamental mechanics of ancient rogue-likes. Some games find a hybrid space, such as *Tales of Maj'Eyal* which conforms to traditional gameplay while allowing progress between games. The most noteworthy contemporary rogue-like developers, however, tend to stick to the fundamentals which made original rogue-likes so fun. (*ToME*, *Cogmind*)

My project adheres more to the latter methodology. While I am fond of many contemporary rogue-likes, the mechanics required would have been too complex for me to work at a low level, forcing me to use a game engine, which would have foregone the learning experience. As such, much of my design methodology was strictly bound to foundational rogue-like mechanics.