The Blind Forest

By Gold Sharkfins

"The Blind Forest" is a 2.5D action-platformer with a focus on mastering of new abilities and exploration into a vast and challenging world. Battle ferocious foes on your journey into the world with the tight and fluid gameplay to get the stone keys and unlock the gates to the temples.

Fight and defeat the guardian to retrieve the last key and open the secret temple.

The Gold Sharkfins Team

Jesse Flanary

Game Development
Enemies
Boss
Boss Level
HUD
In Game Menus
Obstacles

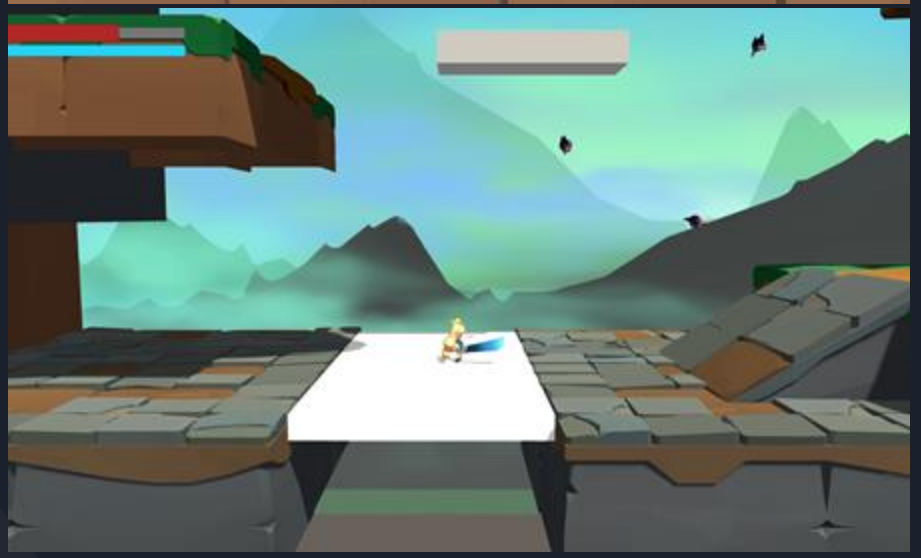
Juan Diego Lugo

Game Development
Game Design
Level design
Gameplay programing
AI programing
Characters design
Artist
UI
VFX
Sound engineer

Prototype and Alpha Progress

Prototype

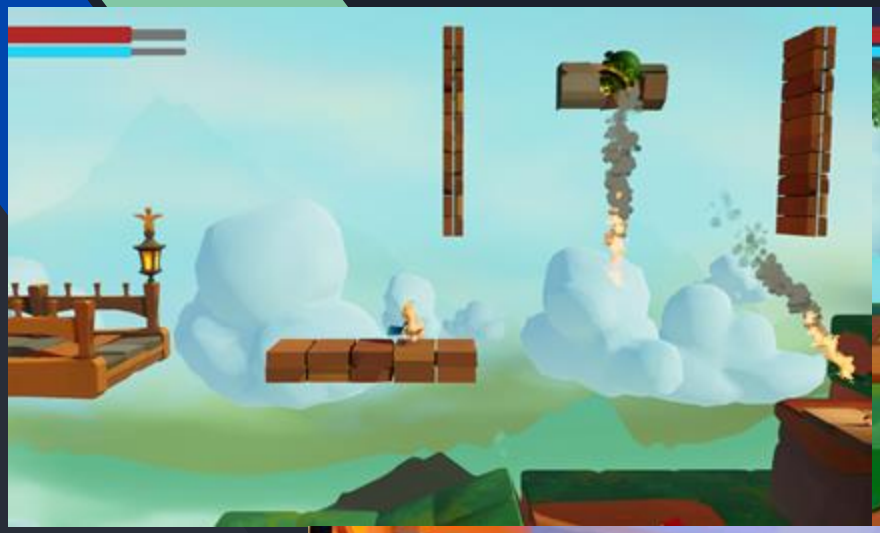
Alpha



After fixing the player model, many of the changes were game for the enemy, player feel and the first version of the boss was introduced(not shown)

Beta Progress

Final



Tips/Tricks

- Take a break
  - Don't be afraid to take breaks. Taking a break will reset your mindset
  - You may come back and realize that what you have been doing either won't work, or will be hard to iterate on.
- Be open to change and improve
  - Don't be afraid to take something that you already made and change it in order to improve or make your project better.
  - Take complicated parts and simplify them. It will help you improve on it later.
  - A unique game feel is in the details! Be able to have simple but effective details to add that will make the game more alive and the gameplay way more fun to play.
    - For example: Play feedback, change of reaction like animations, particles effects on action, hit, death, knockback, screenshake and sounds.
- Analyze system interactions
  - Before starting any major work in open production, think of every way a system, or systems, will interact with other pieces of the project.
    - For example, an enemy interacts with the player. Think about how it finds, chases, and hits the player. What happens when it gets hit and the sequence of events afterwards?
  - Also think about how systems compare.
    - If you have a lot of enemies, think about their similarities(i.e. health, attack power, etc) and decouple those properties from that specific enemy and have another script that you will apply to every enemy that provides basic functionality(such as taking damage).

What Went Right

- We finished the game!
  - Going into month three, the project was on thin ice. There were a lot of bugs and the team health was… hostile. Despite all of our differences, we manage to produce a good and playable game.
- The last minute implementation of the boss was successful
  - We originally were going to cut the boss out. But after needing more work, we decided to do a mini version of the boss. It didn't have all of our original plans but it worked well with the player and their abilities.
- An appealing game
  - Making a game that looks good and makes you want to play it is really hard! We think that we succeeded on that part. The Game design, assets, music, vfx, and render light all came together to make the game look and feel great.

What Went Wrong

- Communication
  - We struggled to agree on major features. We overcame this through an understanding of each other's parts and a trust in each other's abilities to complete their tasks.
- Compromising and losing of team members
  - The project started with four members but ended with two by the end of the second month. Due to a failure to compromise on work and a failure of some members to complete their work, our group was cut in half. We assigned the loose work accordingly.
- Problems during the implementation of assets
  - Certain assets were implemented in a way that caused features to not work in our builds. This was fixed by properly implementing them from scratch .
- Problems with the repository(GITHUB)
  - We encountered a lot of problems building the project in the repository. We determined where the issues came from, addressed it, and eventually fixed it.
- Unused assets
  - Many of the asset packs had a lot of extra files that were never used that increased the total size of the project and the builds. We didn't really have a fix for this, we just have to deal with a slightly larger build size.