

Knights Divide

Google Drive Link for deployment and repo:

<https://drive.google.com/drive/folders/17CJsFeGNfGBKM2VUlvcVqD6uJCk8Q7ok?usp=sharing>

Knights Divide is a 2D action-RPG consisting of five levels where the player alternates between playing as Lancelot the melee specialist, and Raevyn the highly mobile ranged character. Levels consist of navigating the sometimes-maze-like dungeons and caverns that make up the map, defeating enemies, and concluding with a boss fight that will gain the player entry to the next level. The player beats Knights Divide once they've ascended to the highest level and destroyed the mysterious final boss.

Timeline

Knights Divide began with our original project proposal. The scope of Knights Divide was slightly wider in the beginning with more levels planned and a more detailed artistic progression of level design and game geography. We started development with the first level which included a full map, Lancelot as the playable character, basic enemies, and a boss fight. The core combat mechanics that were planned for Lancelot like blocking and a counterattack were implemented at this stage and we also experimented with various map components like moving platforms and sections filled with water.



(Figure 1, Original level 1 design with toxic water and moving platforms, moved to later levels)

Development of the second level began after we met and decided to focus more on the core mechanics and push the water and moving platforms to a later stage in development. Knights Divide alternates characters every level, so creation of level two meant the

development of Raevyn as well as new enemies who better matched her style of gameplay and a map design that promoted agility and speed.



(Figure 2, Raevyn standing against a mutant skeleton and the level 2 boss)

The final three levels were built largely at the same time along with the development of new enemies and bosses, compatible with the character that they would be fighting. Development concluded with a final boss fight where we created a unique model for the boss as well as animations and new types of attacks.

Core Mechanics

The initial stages of the core mechanics are implementation of each character's unique abilities. Lancelot has the ability to block and if timed correctly, he can do a perfect block followed by a powerful counterattack. Raevyn on the other hand does not have the ability to block, however has the ability to dash and if timed correctly, she can do a perfect dash increasing her overall abilities. Another thing that sets Raevyn away from Lancelot is that she has the ability to wall jump. Lancelot on the other hand can only progress through levels by climbing on top of the platforms.

Furthermore, Lancelot mainly focuses on melee combat while Raevyn mainly focuses on ranged combat. We initially planned our game to have new weapons added where each player gets to choose from when progressing through the levels, but that idea was soon scrapped when the sprites we got from the internet having a weapon imprinted within the sprite itself instead of a separate sprite. Which created a lot of issues for the scalability of newer weapons for each character. To mitigate this loss, we decided to add skills for each

character. When each character finishes a level, they obtain a unique skill they can use in the future levels.

We also added a level up functionality, that provided the characters defeat enough enemies, the player can level up to which he or she can obtain an ability point. The ability point can be used to increase the player's stats like more health, attack speed, cooldown speed in the stats menu to improve their gameplay.



(Figure 3: the stats menu where player can increase to improve their gameplay)

Testing

Testing was done many times throughout the course of development. Each map had to be tested to ensure the player and enemies could move around properly and to uncover any bugs. During map testing, we found several times that Lancelot would function a bit different than Raevyn. For example, the special audio for enemies would not be detected when Raevyn gets close to the enemies. Furthermore, the animation for each character would get jumbled up whenever someone merges due to plastic SCM overwriting files without checking which file has more functions. It overwrites files based on when the merge happens, and not based on whether the file is updated or not.

Technical Challenges

The development of Knights Divide brought several technical challenges, particularly in collaborative workflows, gameplay implementation, and asset management. Addressing these issues required a combination of systematic problem-solving, effective communication, and strategic use of tools like Plastic SCM. These challenges not only tested the team's technical capabilities but also provided invaluable learning experiences that shaped the final outcome of the project.

One of the earliest and most persistent challenges was managing version control. With multiple team members working on different aspects of the game simultaneously, merge conflicts became a frequent issue. Early on, the team faced accidental overwriting of work due to poor synchronization of changes. To address this, Plastic SCM was adopted as the primary version control system, offering robust branching and merging capabilities. Additionally, a strict workflow was introduced, where updates were pushed to individual branches, followed by peer reviews before merging into the main branch. This approach minimized conflicts and ensured a smoother development process.

Managing the game's growing library of assets posed challenges related to performance and organization. As more textures, animations, and sound effects were added, the project began to suffer from longer load times and increased memory usage, particularly during testing. To address this, the team compressed large assets and streamlined the project by removing unused resources. The synchronization of animations with gameplay mechanics required careful attention. Custom animation events were added to Unity's Animator Controller, ensuring that actions such as damage application and cooldowns aligned perfectly with the corresponding animations.

The screenshot shows the Plastic SCM interface for the project 'Knights Divide'. The left sidebar includes 'Pending Changes', 'Branch Explorer' (selected), 'Changesets', 'Branches' (selected), 'Labels', 'Attributes', 'Code Reviews', 'Sync to Cloud', and 'Merge'. The main area displays a tree view of branches under the '/main' repository. The 'Development' branch is expanded, showing sub-folders like 'Abhishek_Enemies', 'audio', 'Boss', 'DarkLevelAbhishek', 'DarkWizard', 'final_Level', 'LancelotFixesAndStuff', 'levelDesign', 'levelRedesign', 'MainMenu', 'manan_Level_design', 'MananFinalChanges', 'MananFinalChanges2.0', 'Manav_HotbarUI', 'Manav_RaevnyAnimation', 'Manav_RangedEnemy', 'Manav_RevampInput', 'Manav_RPG', and 'Manav_Shooting'. A message at the bottom right says 'Please take our quick survey to share your thoughts on our product and help us improve your experience.' with a 'Take the survey' link.

(figure 4: plastic SCM's branches)

Future Work

While Knights Divide successfully delivers a compelling gameplay experience with its unique features and mechanics. There are several areas where the game could be

expanded and improved. These enhancements aim to increase replayability, improve player engagement, and provide additional layers of complexity and depth to the game.

One of the most promising directions for future development is the addition of more levels and environments. Currently, the game features a single dungeon aesthetic, but introducing diverse biomes—such as fiery caverns, icy ruins, or mystical forests—would enrich the player's experience. Each new environment could come with unique challenges, such as elemental hazards or specialized enemies, further encouraging players to adapt their strategies. Another avenue for improvement is enhancing the game's narrative. While the current version does not provide any backstory or the lore about the game. We can incorporate backstories using dialogues, collectible items and cutscenes in the future.

Finally, improving the game's accessibility and polish would enhance its appeal. Features such as difficulty settings, a comprehensive tutorial, and localization for multiple languages would make the game more inclusive for players of different skill levels and backgrounds.

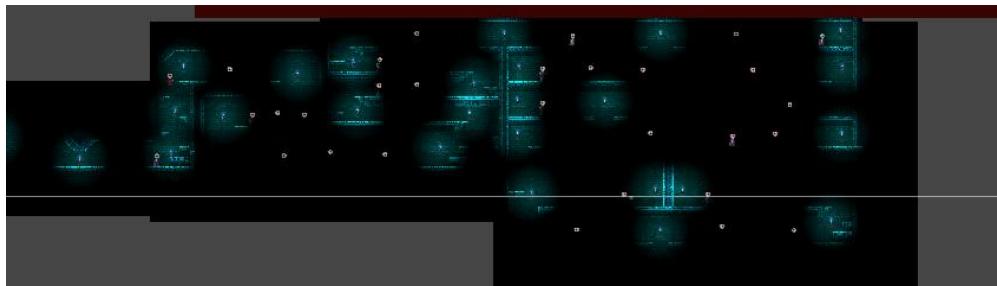
Level Designs

Each level is equipped with a new boss that player must defeat to progress through the game. We currently have 5 levels, 2 levels for Lancelot and the other 2 for Raevyn while the final level being a choice between Lancelot and Raevyn. There are two different level type for both characters, one being the dungeon setting (figure 5) and the second being the dark setting (figure 6). Raevyn's level is a copy of Lancelot levels but flipped with a completely new boss.

Lighting plays a crucial role in level 3 and 4 as it creates the atmosphere needed to feel isolated and alone. Unity's 2D lighting tools were used to craft a sense of depth and immersion, with torchlight illuminating the dungeon's corridors. These lighting effects not only enhance the visual appeal but also impact gameplay, as dimly lit areas conceal enemies, encouraging players to explore carefully. The use of lighting also helps to subtly guide players, with brighter areas often signaling key objectives or paths forward.



(figure 5: level 1 design)



(figure 6: level 3 design)

Program Code Structure

Our program code structure split into sprites, tilemaps, prefabs, sounds, scripts and scenes. The sprite folder contains all the sprites and animations for all characters and bosses. The tilemap folder has all the necessary tilemaps needed to build our game.

The structure for our code is dependent on the type of code we used. The skill functionality uses abstract script that act as a placeholder to create any child scripts that follows an unique skill. Lancelot and the enemies primarily use the event invoking system and animation to simulate attacking and damaging. Raevyn uses the input system and manager to simulate attacking.

Bosses each have their own folder which contain all of the scripts they need to fight which happens primarily with a system of 2d colliders that are used to detect hits. The colliders sit on child objects of the boss and have scripts attached which call on the health script attached to the player to reduce their health every time a collision is detected.

Artistic Aspects

The visual design of the game centers on a pixelated aesthetic. All characters, enemies, and environmental elements were designed using pixel art. Different enemies consist of different animations and attack patterns. Lancelot is designed with armor to show strength, while Raevyn highlights agility. Environmental assets, such as dungeon walls and torches, all fit together like puzzle pieces. Lighting is the key part of the game's visuals, especially in the third and fourth level, where the areas are mostly dark and the only source of light are the torches on the wall, making the game feel more real despite its pixelated style.

The game uses bright colors for enemies and players, and earthy colors for dungeon and background, making it easy to spot the keys prefabs. Users will also find random dungeon prefabs lying around in the dungeon just to make it look more realistic. Similar patterns of tiles have been used in the tile mapping system to maintain consistency. Audio plays a great part in enhancing the feeling of the dungeon, by having different background music, on different levels and spooky sound effects attached to the enemies.

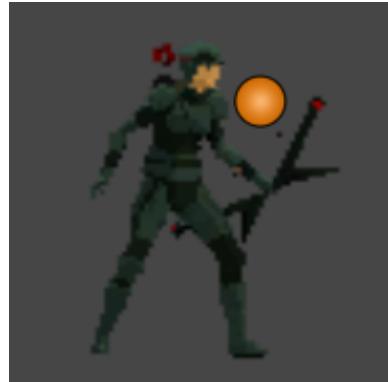
Assets

All characters, enemies, backgrounds, tiles and environmental elements were carefully selected to maintain a cohesive art style. Most of the assets in the game were primarily sourced from free resources available on the Unity Asset Store for example, the audio, tiles for tile mapping, main character prefabs etc. These pre-made assets saved development time while still delivering a polished aesthetic. The final boss, however, was entirely designed by the team to ensure it stood out from others and had a unique touch to final level.

For audio, the game incorporates dungeon-themed sound effects and background music, all sourced from free online libraries, with ominous tones to match the dungeon's dark and mysterious vibe, while combat sounds like sword clash and enemy growls enhance immersion. Our assets also consist of various scenes and Main Menu, teaching the users what the controls are and the option to start or choose any level. Many C# scripts can also be found in the assets of our game, which are further attached to the prefabs and scenes of the game to perform a specific task or action, like character movement, changing levels, audio system etc.



(figure 7: Lancelot)



(figure 8: Raevyn)

Bugs

There were a few different bugs that cropped up over the course of development. Most were removed, however one compiler error which we struggled to diagnose was caused by the moving platforms and sometimes prevented the game from building despite running properly in the unity editor. A few smaller bugs remain in the levels but are mostly graphical.

Lessons Learned

The importance of transparency and communication was the biggest lesson learned over the course of this project. Particularly in the early stages of development, time could have been saved if we each had a better understanding of how other team members were building their respective parts of the game.

Conclusion

Knights Divide began development in mid-September 2024 with a broad scope and a unique take on character progression driven RPGs. It centered around two characters with vastly different playstyles and contained a wide array of enemies. As development progressed Knights Divide focused in on core mechanics and level design tailored to fit the differing playstyle of each character. As we built out our core mechanics the scope of the project narrowed. Things like story development and world building were still done but to a lesser extent than was originally intended. We had also planned a detailed progression of the lighting, background art, and general color-scheme that would be indicative of the player's progress. There was attention to things like the color scheme and art but with time constraints and our narrowing scope we did not build them out the way we had originally hoped. Despite missing some visual components, the benefit of this strategy of time allocation and focus on core mechanics is that combat is very good and control of the characters feels very sharp and consistent throughout the levels. Another part of our original scope that went well was the enemies. We wanted to create many different types of enemies and bosses, and we did so in a way that fits our combat mechanics and is visually appealing. Ultimately, we were able to include most of the gameplay components we had planned in our original project scope, and finish with a strong and fun game.