Goblin’s Keep is a 2D tile-based escape game set in a castle infested with goblins that hunger for human flesh. You play as a courageous knight navigating a dangerous maze, evading traps and goblins along the way. To win, the player must first collect all the keys scattered across the map, then locate and activate a lever to unlock the exit. Only then can the player escape. But beware — if a goblin catches you or your score drops below zero, you’ll be captured and devoured in the goblins’ long-awaited feast. Escape quickly... if you want to survive.

We remained faithful to our original vision during development, though we did make some significant omissions — notably, the removal of multiple floors and power-ups. Our initial concept imagined a knight entering a castle rumored to contain treasure at its top level, guarded by goblins on every floor. This narrative remains compelling for a 2D maze game, but during development, it became clear that building and debugging even a single map was a time-consuming challenge. Once we finalized the first level, we had no time to expand further, and thus, multiple floors had to be cut. However, we still believe that realizing this full concept — with multiple floors and an ultimate treasure — is essential to turning *Goblin’s Keep* into a complete, player-enticing experience. We hope to pursue this vision after the course ends.

With the decision to stick to a single level, we also removed power-ups from the game. Originally, we wanted goblins to become faster on later floors, and had designed power-ups (speed boost and freeze) to counterbalance this increasing difficulty. These power-ups would have allowed players to temporarily run faster or freeze nearby goblins — offering tactical advantages in high-stakes situations. While we were excited about this gameplay layer, it didn’t make sense to include it with only one level and uniform goblin speed. We were disappointed to let these features go, but we believe they can be excellent additions in future iterations.

Regarding code structure, our early UML diagrams were drafted with zero prior game development experience. Looking back, we noticed several changes. For instance, the UML's StaticEntity is now MainObject, and GameEntity, GameLoop, and MazeGame roles are now handled by GamePanel. The only major structural change was moving the Punishment logic into the MainObject. The MapGenerator class underwent significant evolution during development. Initially, it was responsible for both generating the map and managing gameplay logic (like time progression, bonus management, and collisions), which made it overly bloated. To improve modularity and separation of concerns, we introduced a MapHandler class to handle gameplay updates, object interactions, and win/loss conditions — resulting in cleaner and more maintainable code.

Throughout this project, we learned valuable lessons that will serve us well in future work. One of the biggest takeaways was how to effectively work as a team. In the beginning, we were unorganized, inexperienced, and lacked communication. By the end, we had developed strong collaboration skills, communicated our ideas clearly, and built a shared understanding of the codebase. This project was also our first time writing tests, which helped us appreciate the importance of comprehensive testing to ensure correct functionality and code reliability.

Developing *Goblin's Keep* has been one of the most rewarding and eye-opening experiences of our time at SFU. We had the chance to work on a compelling idea, see it come to life, and learn what it takes to build a game from scratch. We're proud of what we've created — and we're even more excited for what comes next. We now feel equipped with the tools, skills, and experience needed to pursue our goals as software developers.

To see a tutorial/demo, check out our trailer on YouTube: 🎬 game tutorial.mp4