

For each part that changed, explain why and what didn't work about your original plan. For parts that didn't change, explain how the design decision ended up helping you.

In our original plan of the maze, our program included a 2D or 3D rendering of a randomly generated maze with a page that allowed for user inputs. The maze generator implemented a Backtracker class that we created to ensure that each cell would be visited and there would be a unique solution. A JPanel for front page graphics allowed the user to input dimensions, a key, and decide if they wanted a 2D or 3D maze. In our 3D graphics class, our initial design consisted of the bare bones of the maze without any of the "flashier" graphics. The maze walls were all the same color and there were no potential obstacles to finish the maze.

After this initial build of the program, we began adding different shading techniques to give the depiction of a more "dimensional" maze and make it seem as though the user is really moving through it. We also added the choice of enemies and health in the 3D rendering, where the user could "attack" enemies with key listeners on the space bar. If they were not quick enough, their health would decrease, and hearts around the maze could increase their life. Later, we added code that allowed the user to see the enemies and hearts without being in the cell that contained them. We also made it possible to use the health later if the user decided to come back to it. In terms of navigating through the maze, we chose to include a compass that helped guide the user to the end of the maze as well as shading on the floor to indicate the user's position in relation to the finish (the closer to the finish, the more red the floor would become). We also added a leaderboard that displayed the user's name and time into a txt file if the user chose to submit their score. This would then be displayed in a separate GUI.