For 15 extra points on this milestone--all or nothing-- you will brainstorm ideas for future CS 2340
projects. In a written document, please include:
• A high-level overall description of the proposed project (~5 sentences).
• A brief description (2-3 sentences) for each of the 5 iterative milestones and specific requirements detailed similarly to the current project's requirements. See the Background/Purpose and Implementation sections of the given milestones for examples

Extra Credit Project Proposal
We propose a dungeon crawler/looter game, much like Risk of Rain or Terraria. Overall, future students should design a game with either a single or various small maps. The player's job in the dungeon crawler game is to explore the "dungeon" to find various loot and then exit. Along the way there are enemies to defeat that drop loot as well, and at the very end there can be a final boss to beat with the best loot in the dungeon. The player wins once they leave the dungeon, and loses once they die. Other game engines (i.e Unity or Unreal engine instead of being limited to Android and JavaFX tools) would also be a great plus, and would teach a student and their team how to interact with existing APIs and objects.

M2:
To set up a basis for the game, the team should set up a simple beginning screen to start or quit a game. They must also design classes and objects to handle behavior and interactions between the player, enemies, and items. The focus of this milestone should be to introduce and plan out the project for the team and what design principles and patterns to follow.

M3:
Building off of the beginning screen, the team should implement player and map navigation in such a way that the project doesn't suffer from technical debt. There should be unit testing for this milestone and for the following milestones as well. The team should also create SSDs and use cases to show how the user (player) will interact with the system's menus and objects.

M4:
Once the player can move around the map, the next milestone should focus on item collecting, giving the player a sort of task to do during the game. The player should have an inventory for items, which functionality will be implemented in a later milestone. The focus of this milestone should be sequence diagramming to plan out future interactions and the overall pacing of the game.

M5:
After the player can move and collect items, the team should start implementing player and enemy interaction. The player should die when they are killed, and attacking the enemies is an optional function should the team decide to go for a more adrenaline-packed horror game. The player-enemy interaction should also come with a code smell writeup, where the team is able to see if cross-object interaction does not introduce unnecessary code smells and keeps the code simple to navigate and read.

M6:
Once the player is able to evade/defeat enemies, the team should focus on the winning condition and boss implementation, as well as item upgrades. These items can range from anything to just for money gain or for helping/harming clearing out the dungeon (e.g: extra damage, lower speed, etc.). By adding the final boss and item upgrades, this milestone will test the team's ability to add on to previous milestones to see if they tried to minimize technical debt.