-reading in from file, displaying map July 20
-implement Chamber partitions for easier generation July 20
-implement basic player generation July 21
    -implement movement & input July 21
-implement basic enemies generation July 22
    -implement movement & turnOrder July 22
-implement multiple floors July 23
-implement combat & update input July 23
-implement basic treasure generation/enemy drops July 24
-implement basic potion generation July 24

-implement different player races July 25
-implement different enemies with different abilities July 25
-implement different potions July 26
-implement different treasure generation July 27
    -implement dragon July 27

Shiny DLC
-implement fog of war July 28
-implement basic enemy AI July 28
-implement Druid race July 29
-implement Domesticated Owlbear animal Companion July 29

Everything done by Mingrui Zou, because no partner :(

Player Character

Question. How could your design your system so that each race could be easily generated? Additionally, how difficult does such a solution make adding additional classes?

Answer. I would have a player class that holds all the default properties of the races as fields and has virtual functions for when the player does attacks, uses a potion, picks up gold, etc. I would then have subclasses for the different races, implementing each with its own set of functions for doing things. This allows me to easily add new races by simply creating another subclass and defining its behaviour through its methods.

Enemies

Question. How does your system handle generating different enemies? Is it different from how you generate
the player character? Why or why not?

Answer. I have one superclass for all enemies and subclasses for each type of enemy. I have other classes to randomly determine where and which enemy is spawned. It is different from player generation because the type is random and there are multiple enemies. I am using a singleton pattern for the player so there is only one.

Question. How could you implement special abilities for different enemies. For example, gold stealing

for goblins, health regeneration for trolls, health stealing for vampires, etc.?

Answer. Similarly to the Player class, I have one Enemy superclass with virtual methods and subclasses for each type of enemy. I then define the behaviour of the enemy with its methods.

Potions

Question. What design pattern could you use to model the effects of temporary potions (Wound/Boost Atk/Def) so that you do not need to explicitly track which potions the player character has consumed on any particular floor?

Answer. I could use the observer pattern, such that when a potion gets used it notifies the player and updates the players stats accordingly. Upon entering a new floor, all stat bonuses would be reset.

Treasure

Question. How could you generate items so that the generation of Treasure and Potions reuses as much code as possible? That is, how would you structure your system so that the generation of a potion and then generation of treasure does not duplicate code?

Answer. I have class GameObject that encompasses anything that can exit on the board. GameObject has subclass
Item, which has subclass Potion and Gold. Generation of Gold objects and Potion objects are very similar in that their respcective object is created and assigned to a cell. I have class Map that is responsible for world generation, so that includes players, potions, treasure, enemies, stairs, etc.