Name/UIN:
Nabeel Sheikh/325001678

Name of application:
ASCII Dungeon (title work in progress)

Objective of the game:
The game is a ascii turn-based rouge-lite. It is your job to survive endless procedurally generated levels filled with enemies around the corner, traps, and precious loot all while maintaining your energy level, wealth, and health. Every move is a decision and what risks are you willing to take to survive the ascii dungeon.

Rules of the game:

1. You will spawn on a starting level. This level will be procedurally generated. (Note: levels will generally get larger and contain more objects the farther the player gets).

ASCII VISUAL TABLE

@	#	-	X
PLAYER	WALLS	DOOR	LOCKED DOOR
$	K	H	?
SHOP	KEY	HEALTH POTION	CHEST
o	/	$!$	*E*
COIN	OPEN DOOR	SACRIFICE	EXIT

2. Each move costs energy. A single move up,down,left,or right will move time by one unit. The more you move, the greater chance an enemy may attack. Remember you only have a limited number of moves per level (dictated by your energy level). This will get harder and harder each level.
3. A console will help understand what is occurring and will update with each move.

CONTROLS(subject to change)

1. **MOVEMENT.** WASD. Up,Down,Left,Right respectively.
2. **USE POTIONS.** 1 & 2 buttons will be used to use respective potions in backpack.
3. **EXIT GAME.** Press Q twice. :(

YOU MAKE A MOVE. Things that may occur.

1. **You encounter an enemy.** You will transition to the battle screen. You will take turns attacking. The first to zero health wins. You will gain new attacks at the shopkeeper or chests. Remember, stronger attacks use more energy but hit harder.
2. **You receive an item.** The item will be permanently stored in your backpack. Gather more to get stronger.
3. **Encounter a shopkeeper.** You will have a choice between three items. These items can give passive benefits or be an active item (i.e sword, fire spell, etc.).
4. **You reach the end of the level.** The next level is loaded. This time harder than the last.
5. **You die.** Game will restart and you will start fresh. Better luck next time.

OBJECTIVE:

Score. Score is calculated at the end of each level. Score = ENERGY LEFT + COINS COLLECTED + HEALTH LEFT + ENEMIES KILLED

Jack class outline:

1. Game *//Controls flow of game.*
2. Screen *//Responsible for visual rendering of game objects.*
3. Player *//Keeps track of player stats, items, and attackset.*
4. Backpack *//Holds active items.*
5. Level *//Creates level, spawns items, and has player location.*
6. Item *//Contains stat affects.*
7. AttackSet *//contains attacks.*
8. Attack *//attack attributes i.e damage, energy cost, type.*

UML ON NEXT PAGE -->

GAME	SCREEN	PLAYER
+score -isDead +currentLvl		+health +energy +attackSet +backpack +position
-GameOver()*//end game if player is dead* -Battle()*//transition to battle screen state* -Shop()*//transition to shop screen state* -LvlChange()*//starts next lvl* +update()*//updates game every tick* +getLvl()*//end game if player is dead* +quit() *//quits game*	-BattleScreen()*//prints battle screen* -ShopScreen()*//prints shop screen* -PrintLevel()*//prints level* -PrintPlayer()*//prints player stats* -PrintConsole()*//prints console*	+getHealth()*//returns health* +getPos()*//returns position* +getStat()*//returns data values of player*

LEVEL	ITEM	ATTACK
+lvlArray	-damage -health -energy	+damage +energyCost
-playerMovement()*//controls player movement* -enemyMovement()*//controls enemy movement* -createLevel()*//creates level* -spawnItems()*//spawns items in existing lvl*	-addToPlayer()*//adds stats to player stats.*	+getDamage()*//returns attack damage* +getEnergyCost()*//returns energy cost*

BACKPACK	attackSet	ENEMY extends PLAYER
+item1 +item2	+attack1 +attack2 +attack3 +attack4	-lvl -hitChance
+useItem()*//calls item addtoPlayer()*		+getLvl()*//returns level* +getChance()*//returns attack success chance*