Programming 4  
Assignment 1: 2D Roguelike

1. **Are your player, items and enemies the same class, different classes in the same family, or completely different classes?**The player, items and enemies will all be descendants of the Sprite class. The player character will have special methods to enable him to pick up gold, items etc. There will be an Enemy class which will be used to implement both enemies (different pictures and hit parameters will be set for each type of enemy).   
   The player, items and enemies were all just Sprites – no inheritance was used. A Staircase Sprite was also used to get to the next levelCollision detection methods were used to enable the player to pick up items, fight monsters and get to the next level.
2. **What logic will you put into your Form class? What logic will you put into your Game Manager class?**The Form class will have an object of type GameManager which will contain the logic for implementing the game. The Form will call the RunGame() and SetupGame() methods of the GameManager class on every timer tick.  
   The setUpGame() method was not called on every timer tick – just when the dungeon was initially set up or a new level was generated
3. **What class(es) do you need to implement the dungeon? Briefly explain the job of each class, list the data members it must hold, and the methods it must expose. How do the Dungeon and the TileMap communicate?**Sprite class – will hold Sprite sheet, location and velocity parameters, frame height and frame width, a bounding rectangle, a Tilemap and an enum of directions. It will have methods to draw itself, move itself, check whether it is out of bounds, set the sprite sheets, change direction, detect collisions with other sprites and terrain. May be an abstract class.  
   PlayerCharacter class (a descendent of Sprite class) – will hold HP, score, and will have methods to pick up treasure and item and a combat method for fighting enemies.  
   Enemy class ( a descendent of Sprite class) will hold HP.   
   Treasure class (a descendent of Sprite class) – will hold scoreIncrease. Will erase itself when player character walks over it and score will be increased by the scoreIncrease variable.  
   Potion class (a descendent of Sprite class) – will hold HPIncrease. Will erase itself when player character walks over it and will increase the player character’s HP.  
   Damager class (a descendent of Sprite class) – will hold damage. Will erase itself when player character walks over it and will decrease the player character’s HP.  
   Room class – will hold types of areas that are and are not walkable.  
   Dungeon class – will hold a collection of Room classes.  
   Tile class – will hold a bitmap and an IsWalkable Boolean.   
   TileList class – will contain an array of tiles and will hold methods to get the tile image, set the tile array entry and determine if the tile is walkable.  
   Tilemap class – will hold a TileList, a two-dimesional array and a Graphics object. It will have methods to set and get map entry, draw the map, and determine if a tile is walkable.  
   Classes were modified as mentioned above.
4. **What data structure(s) do you need to hold collections of enemies and items?**Enemies and items will be held in linked lists specific to each class.
5. **Does the dungeon need pointers to its sprites? Why or why not?**As far as I am able to tell, the dungeon does not need pointers to its sprites as the GameManager class will handle the appropriate functionality.
6. **Does the sprite class need a pointer to its dungeon? Why or why not?**  
   As far as I am able to tell, the sprite class does not need pointers to its dungeon as the GameManager class will handle the appropriate functionality.
7. **What enum types (if any) do you need?**There will be an enum to hold the direction of the Sprite class and its descendants.
8. **Does the player sprite need access to the collection(s) of enemy sprites?**Yes, the player sprite will need access to the collections of enemy sprites in order to make combat possible.
9. **What class is responsible for creating the collections of enemies and items?**The GameManager class will be responsible for creating the collections of enemies and items.
10. **If you are using an FSM, what class calls the FSM methods of the sprites?**  
    A FSM will not be used.
11. **At each game cycle, you need to perform collision detection between the player character and each enemy and item in the dungeon. What class or classes hold a method to compare the areas of two entities to check for collision? What is the function header of this method? What other classes are involved in the collision detection logic?**  
    The Sprite class will hold a method to compare the areas of two entities to check collisions. The function header for this method is as follows:  
    bool Sprite::CollidedWithMe(Sprite^ otherGuy)
12. **If you are implementing Line of Sight what algorithm will you use (i.e. room-based or field-of-vision)? What methods are needed, and which class holds each method?**  
    I will not be implementing Line of Sight.