Alexander Phillips (1000024961)

PHILAS1

Planning Document

Rougelike c++

Programming 4 Assignment

## Are your player, items and enemies the same class, different classes in the same family, or completely different classes?

Current implementation from previous lab puts all my ‘things’ that move are drawn onto the screen using the sprite class.

Better implementation would be inheritance approach with a superclass thing and player, enemy, item branching off.

## What logic will you put into your Form class? What logic will you put into your Game Manager class?

Form will hold interaction with form components such as movement of the player character using key input

Whereas the game manager class will contain logic to run the game, compared to the form which will mainly output the game.

## 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?

Dungeon: Will need to be able to generate the dungeon as a 2D array which will then be passed to the tilemap

Room: Will need to be able to set its starting coordinates on the 2D array holding width and height

Cell: Will contain the information of the cell, which will be stored into the 2D array.

The tilemap could have a method which asks for converted 2D array from dungeon which then draws it.

## What data structure(s) do you need to hold collections of enemies and items?

Linked list! Previously created architecture will be used to manage the enemies & items. Flexibility of this architecture seems the most logical to use.

## Does the dungeon need pointers to its sprites? Why or why not?

Possibly for creating correct phasing of player location when portal to next level is stepped on

## Does the sprite class need a pointer to its dungeon? Why or why not?

Yes, it needs to understand the bounds which it can be created within.

## What enum types (if any) do you need?

Direction of sprite.

Type of Cell.

## Does the player sprite need access to the collection(s) of enemy sprites?

Shouldn’t, this hopefully will be handled within the game manager which will deal with any interactions between

## What class is responsible for creating the collections of enemies and items?

Current implementation from previous submissions is in the form but game manager should be the ideal data structure to create these objects.

## If you are using an FSM, what class calls the FSM methods of the sprites?

Currently a FSM is not planned out to be used.

## 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?

Game manager should compare the rectangles of each enemy with the main character.

Check Collision ( Sprite^ MainSprite, Sprite^ EnemySprite )

## 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?

Room based method sounds the simplest to implement, currently unknown what implementation will be needed for this.