Energia powered ‘22
C++ and OOP

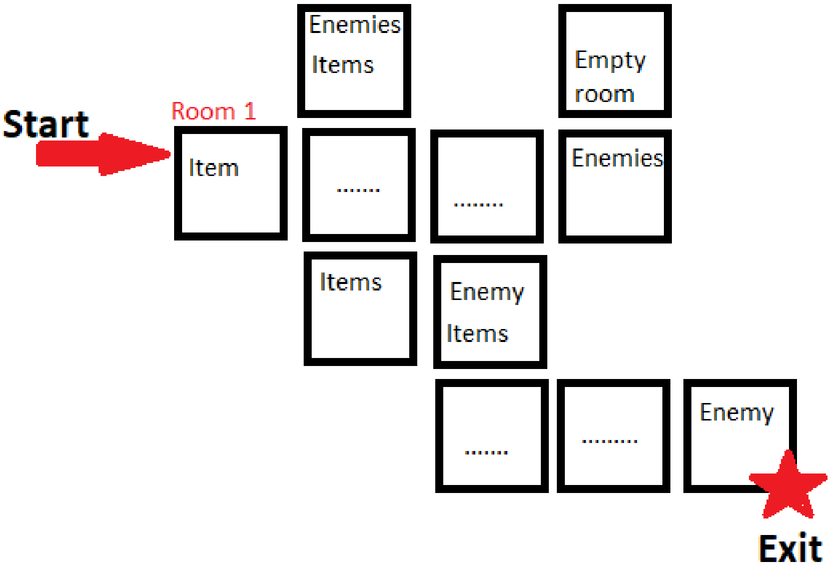
Final Project

Text-Based C++ Game

It is a game application that uses texts as input and output instead of bitmaps or 3D graphics.

Dealing with GUI needs much deeper understanding with C++ and OOP this is why a text-based application is suitable for this level.

Which game we are going to build?



Sample description:

It's a maze game that consists of rooms, each room can contain enemies to fight, items to collect or it can be an empty room. The goal of the player is to move from the start point (Room 1) to the exit, in order to win the game, and surely he must fight all the enemies that appears on his way and not be killed by them.

As we said before, we're not going to deal with GUI's, so everything in the game should be described, every action and movement should be asked and taken from the user:

- Health, attack and defence of every game characher (enemy and player)
- What appear in the entered room
    - ex: You enterd a room and you find a big monster with _health, _attack, _defence.
    - You entered a room and you find a gun that increase your health with___, and your attack with___, and your defense with ___.)
- Action to be choosed by the player
    - ex: fight enemy or retreat
    - loot Item
    - move to another room (Move up, down, left, right, depending on the location of the room)

Before writing your code you need to:

- Think about the objects that exist in the game in order to name the classes you're in need.
- List the main data member of every class in addition to the member functions (You don't need to list all of them at this step as through coding you may discover missing functions or missing data, now you're just putting the basics of your game)
- Identify the relations between every class, in order to know which class need to be constructed first and what class depends on the other. (Composition, aggregation, association and inheritance)
- Think of an idea to make your maze map, in other word, how will you order the rooms in the game, will you put all of them in an array and give every room an index? Will you put them in a 2D matrix? ....? etc
  (Don't worry if it's not clear, it will be discussed later but you need to put this questions in your head for now)
- In the maze class, this is where all the logic happens, means that the game loop function will be written in this class, so think about the main steps that need to be written to be able to run the game continuely.
- Know how you will terminate the game or break the game loop(When wining or when killed by an enemy)

- After giving yourself enough time to think about the answers and analyze the project (30 minutes minimum) you can now jump to the next section where most of questions are answered.

(Don't skip this part, in order to get the most benefit from the project)

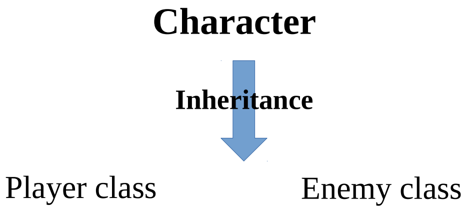
Program classes:

Item class

An object of Item class will be used in the room, or grabbed by the player, it should hold a name, a value to increase the player's health when he grabbed it, value to increase attack and other to increase defence.

Character class

It’s The base class of all the charachter in the game (Player and Enemy), it holds the same data member of Item class in addition to a value to keep track of health and maxHealth and some other functions (To take a damage when fighting and to check if dead after fighting)



(No need for a new class called enemy as nothing new need to be added as the player class, so when you need to make an enemy object just do it using class charachter)

Room class

A room should has a position, a flag to indicate if it's the exit or not, and as we said before, a room can contain a list of enemies or items or both.

(We will discuss the concept of position later)

Player class

It's a derived class from charachter as a player had to keep track of his current room and his previous room in order to be able to return back to it when retreating from fighting the enemy, in addition to a list that contains the item collected by him. And finaly some additional member functions will be added to this class (Think of them :D)

Maze class

This is where most of the logic is written, a maze contain a player and some rooms (no need for items and enemies again as they are already included in the room class).

The most important function in this program is the game loop function:

It is an infinite loop with a breaking condition (After the game is over or wining the game, you may ask the user if he wants to continue or not if not then break this loop and terminate the program), any other functions in the code, most probably are called by it.

In order to write this function you need to think about, What is the first thing you need to do when starting playing the game?

Answer: Enter the room (I hope you guessed it right :D)

- So now you need a function called enterRoom that will be called at the begining of this game loop.
- Then, check the content of the room (room with enemies, items or emty) and take an action from the user depending on this content.
  - ➔ if enemie: call a function that handleRoomWithEnemy, it asks the user to take an action if he wants to fight enemy or retreat. According to the action taken, if he wants to fight so call another function to handle this event.
    - ▪ FightEnemy ---> it increases the status or kill the player

  if he wants to retreat ---> go to the previous room

  - ➔ if items ----> loot items
  - ➔ if empty ----> nothing to handle
- Finaly, ask the user which room he wants to move to (changeRoom).
- Loop again.

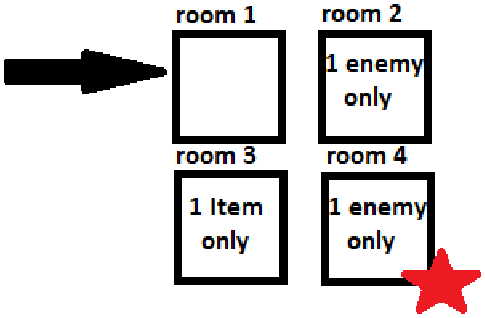
In source.cpp: (Where the main function is written)

You start by making your game scenario and objects, i.e. create maze object and player, rooms, items, enemies, and place them in any way you want, not necessary as the picture in the first page of the document, then call the game loop of maze class.

But make sure that any scinario created should work on your code, make your code more abstract and not to a specific scenario only.

Rules to be followed:

- Every function should be dependent, no function does 2 funtionalities or more at the same time,.
- Start step by step, think of the game in a simple form as possible as you can, i.e.



Start by making the maze 4 rooms only, every room contains only one charachter or only one item or emty, then continue increasing your program (in other word, imagine your game is only as the photo shown above, write a code for this maze and then get the stuff cooler and more challenging)

- If a class has a room as a data member it should be a pointer.
- If you pass any room to a function then pass it by reference.
- For the room position, you can give every room a number as the photo above and put them in an array, but this method is not so efficient as it doesn't work for all scenarios, but you can start by this method first, then after writing the game logic and understand well your program, think about other methods. (2D array, pointers, ...)
- Make your code well organised in order to be able to debug when getting any errors