

Workload

As the screen and screenElement classes are taken from assignment 1, those are not divided up.

Zach

- Player class - Will code and test
- Enemy class - Will code and test
- Room class - Will code and test
- Entity class - Will code

Broderick

- Item Class -Will code and test
- Weapon - Will code and test
- Armour - Will code and test
- Potion - Will code and test

Darsh

- Chest Class - Will code
- Battle Class - Will code
- Testing main program

Jared

- World Class - Will Code

Nathan

- Coding main program
- Testing various classes
- Factory class - Will Code

Coding Conventions

File Headings

At the top of every file, we will include the name of the author, the date that it was started, and a rough overview of what the file does

Ex:

Name:

Date:

Description: This is is an example

Naming Conventions

Object and class names will begin with a lowercase letter, and any other words in the name will be capitals.

Ex:

screenElement, playerSpawn

Differences from initial plan

There are many differences from our initial plan that we had to change for this stage.

Below is a list of all changes for classes:

-Entity class added

Player Class:

- Added locationX and locationY
- Added a vector of items
- Put updateHP under the entity class
- Now inherits from the entity class

Enemy Class:

- Now inherits from the entity class
- Added void enemyDamage(int attack)
- Added void enemyAttack(player &p)

World Class:

- Changed int roomNumber to a vector of class room.
- Added new functions (startBattle, allowSave, allowLoad)

Item Class:

- Has been completely redone. Here is the new version of it:

Item:

This class is an abstract base class

Attributes:

Behaviours:

Virtual Void open(player &p) = 0; - Adds the item to the player p's inventory.

Weapon class

This class is derived from the Item class

Attributes

- string name = the name of the weapon
- int atk = the attack strength of the weapon

Behaviours

- weapon(const string &s, const int &i);
- void open(player &p) - Adds the weapon to the player p's inventory.

Armor class

This class is derived from the Item class

Attributes

- string name = the name of the armor
- int def = the amount of additional defence which is added to the player.

Behaviours

- armor(const string &s, const int &i);
- void open(player &p) - Adds the weapon to the player p's inventory.

Potion Class

This class is derived from the Item class.

Attributes

- string name = the name of the potion
- int HP = The amount of HP that the potion recovers

Behaviours

- potion(const string &s, const int&i);
- void open(player &p) - Adds the potion to the player p's inventory
- void use(player &p) - uses the item. Ex: potion.use() calls player.updateHP(potion.hp)

Chest Class

- Chest class has been updated to reflect the new item class.

Battle Class

- moved allowSave and allowLoad into the world class
- added new function fight which is where the actual battle information takes place.

-Room Class added

- This class controls all of the information of a room.

-Factory Class added

- This class is a factory class used to create entities and items.