

Thi Vo

Tempestt Neal

COP 3331

22 April 2019

Flex Project Submission 2: Trivia RPG

Flex Project Submission 2 Updates (look out for stuff in bold!)

- **Hello. I understand that changes here were rather drastic compared to what was laid out with Submission 1, but I was under a time crunch when I didn't expect this program to be as complicated as it would be if I didn't make some changes. I did not expect to heavily rely on interactions between the Enemy and Avatar class, which lead to massive changes in parameters. The Avatar class needed more functions to shorten itself, even though it was already lengthy. While the Avatar class may have too many functions, it is pretty offset by the Boss and Enemies class having little to no functions now due to Enemy being the abstract class. Thank you for your understanding!**
- Added main file to document and functionality
- Changes made to Avatar Class
 - Clarified how many values were used in stats data member (7). Speed stat removed from vector<int> stats. Considered unnecessary complications.
 - equipment changed to vector<pair<string, int>>.
 - items changed to int<pair<string, int>>.
 - private int temp_defense added.
 - 1 constructor added: Avatar(string = Xenon).
 - Added get_name() function.
 - Added get_level() function.
 - Added action Enemy&, int &) function.
 - attack() function has Enemy& parameter.
 - questions() function has Enemy& parameter. Only manipulates enemy stats now. No analyze functionally due to analyze() function being removed.
 - flee() function now takes in integer.
 - die() function returns an int instead of being void.
 - level_up() function now has an Enemy parameter. Mention how current EXP stat is changed in level_up function.
 - Added equip() function.
 - Added print_items() function. Also writes out to "Inventory.txt" file.
 - Added drop_item() function.
 - Added pick_up() function.

- Added valid_choice function.
- fight function now takes in two reference parameters of Avatar and Enemy.
- Changes made to Enemy Class
 - Considered an abstract class now because random_choice() is now a pure virtual function.
 - Moved here EXP static int data member from Enemies.
 - Added get_name() function.
 - Moved here get_item() function from Enemies.
 - Moved here get_EXP() function from Enemies.
 - defend() function removed. Considered not important in Enemy AI.
 - analyze() function removed. Considered not an important advantage for the user.
- Changes made to Enemies Class
 - Removed all data members.
 - Created a constructor that takes in an int.
 - Has overriding random_choice function.
 - die() function removed. Considered unnecessary as the function is already in the Enemy Class.
- Changes made to Boss Class
 - Created a constructor
 - Added overriding random_choice function.
 - die() function removed. Considered unnecessary as the function is already in the Enemy Class.
 - fight() function removed. Function was edited in Enemy.cpp to work for both Boss and Enemies objects.

Important Requirements

- Encapsulation
 - Where: Enemy.h
 - Line #: 21 – 24
- Inheritance
 - Where: Enemies.h
 - Line #: 6
 - Where: Boss.h
 - Line #: 8
- Polymorphism
 - Where: Enemy.h
 - Line #: 30
 - Where: Enemies.h
 - Line #: 10
 - Where: Enemies.cpp
 - Line #: 80 - 99

-
- Where: Boss.h
- Line #: 12
- Where: Boss.cpp
- Line #: 21 – 33
- Static Data Members and Functions
 - Where: Enemy.h
 - Line #: 24, 29
 - Where: Enemy.cpp
 - Line #: 16-19
- Friend Functions
 - Where: Avatar.h
 - Line #: 52
 - Where: Enemy.cpp
 - Line #: 67 – 122
- Overload Operators
 - Where: Enemy.h
 - Line #: 34
 - Where: Enemy.cpp
 - Line #: 62 - 65

Summary

You are a character who is playing through an entire RPG dungeon. This entire RPG adventure is all text-based (it's up to the user to fill in the gaps with their imagination). This dungeon is crawling with various enemies and one final boss. However, you will gain EXP, collect items, and use OOP and trivia knowledge to help you along the way. The game will end when you die or you defeat the final boss.

Outline

1 Main + 4 Classes

- **Main**
 - **Read in “Intro.txt”**
 - **Tell user about default name. Ask if they want another name, which will lead to creating the one Avatar() object.**
 - **Explain by reading in “Tutorial.txt”**
 - **Allow user to choose between using the Menu or Exploring. This is repeated until user dies or is level 10.**
 - **If the user reaches level 10, then a Boss() obj will be created. The user can either die from the fight, or win and read in the “End_Credits.txt” file.**
 - **read_file(const string&) function reads in any text file regularly.**

- **valid_choice(char, char, char) function checks to see whether an entered character matches either of two required characters.**
- **Enemy**
 - Superclass of Boss and Enemies
 - **4 Data Members (Protected)**
 - Name (string): Contains the enemy name.
 - Stats (vector<int> (**6**)): Contains the current HP, max HP, critical HP, Attack stat, Defense stat, and Grip stat.
 - Item (pair<string, int>): **An item that the enemy will have and can use.**
 - **EXP (static int): EXP amount the Enemy can drop.**
 - **9 Member Functions**
 - **Get Name: Returns enemy's name.**
 - **Get Drop: Gives the drop data member to the user. Occurs when enemy dies and has a drop.**
 - **Get EXP (static int): Transfers set EXP to user. This function is static as it's returning a static data member.**
 - **Random Choice (Pure Virtual).**
 - Attack: Uses a normal attack.
 - Use Item: Only turn-based move that isn't random and uses **the enemy's healing item.**
 - Die: **Returns an int to see whether current HP <= 0.**
 - Operator[]: Allows easier accessing of values in the Stats vector via subscripting.
 - Fight (friend definition): **Allows a fight to occur between an Avatar and an Enemies objects. Also allows a fight between an Avatar and a Boss objects.**
- **Boss**
 - Subclass of Enemy
 - Will spawn once the user reaches level 10.
 - Updated Data Members of Enemy Superclass
 - Name (static string): Ragnarok
 - Stats (int vector) – Max HP: **500**, Critical HP: **125**, Attack stat - **300**, Defense stat – **300**, Grip stat: **9999999** (No escape)
 - Item (pair<string, int>): Will 100% be a Med-kit.
 - **1 Constructor + 2 Member Functions**
 - **Boss() : Initializes data members and seeds random number generator.**
 - **Random Choice: Calls the roll dice function or use item function if current HP <= critical HP.**
 - Roll Dice: Certain events will occur depending upon the value of the dice (2-12).
 - 2 (Snake Eyes) – Reduces user HP to 1.

- 3 (Attack) – Normal Attack.
 - 4 (**Swap**) – **Swaps Avatar and Boss' Stats. Attack and Defense swapped only.**
 - 5 (**Unexpected**) – **Boss damages himself.**
 - 6 (**x1.03**) – **Give the boss a slight boost in this Attack and Defense stat (x1.03).**
 - 7 – (**Laser Beam**) – Performs a normal attack with 2x the damage.
 - 8 (**Triple Attack**) – **Performs a normal attack 3 times. (Doubles temporary defense for user to half damage)**
 - 9 (***Snap***) – Halfs the user'
 - 10 (**Attack**) – **Normal Attack.**
 - 11 (HP Steal) – Steal some HP from the Avatar. (**20% of current HP**)
 - 12 (Restoration) – Heals half of the Boss' HP.
- Enemies
 - Subclass of Enemy
 - **1 Constructor + 2 Member Functions**
 - **Constructor: Takes in avatar level parameter. Initializes values by reading in enemies from "Enemies.txt" file. Also seeds in random number generator.**
 - **Random Choice: Chooses which function to perform during an enemy's turn. Will 100% choose Use Item if current HP <= critical HP.**
 - rps: Chooses a random Avatar stat to affect. Plays rock-paper-scissors with the user. Will continue until a tie doesn't happen. Who ever wins gets a stat increase or decrease in favor.
- Avatar
 - **6 Data Members**
 - Name (string): Contains avatar name. Default name is Xenon. Can choose name at the start of the game.
 - Stats (vector <int>(7)): Contains the current HP, max HP, current EXP, EXP cap, attack stat, and defense stat, and evasion stat.
 - Level (int): Contains the avatar level. Default level is 1. Max level is 10.
 - temp_defense (int): Will serve as a temporary defense stat. Only functionally used in defend() function.
 - Equipment (**vector<pair<string, int>> (3)**): Contains the weapon, helmet, and body armor. These will increase the attack and defense stats respectively.

- Items (**vector<pair<string, int>> (5)**): Items you can use while in exploring-menu or in-battle. Limit of 5 items in inventory at a time not including equipped equipment.
- **1 Constructor + 18 Member Functions**
 - **Constructor: Uses 1 string as a parameter that initializes name. Initialize other data members. Seeds in random number generator.**
 - **Get Name: Returns name.**
 - **Get Level: Returns level.**
 - **Action: Allows a user to choose from a set of battle actions: attack, questions, use Item, defend, or flee.**
 - Attack (option): Use a normal attack.
 - Questions (option): You are first asked what stat you want to manipulate(**always enemy**). You are asked a question about OOP or just questions about anything (4 options). If successful, the desired stat decrease will occur. If unsuccessful, the opposite of the desired stat decrease will occur.
 - Use Item (option): Use a **healing** item from inventory.
 - Defend (option): Increases the defense stat for one turn. **Uses the temp_defense data member.**
 - Flee (option): Try to escape current fight. Succession depends upon evasion stat and grip stat of the enemy. 0% success with the boss.
 - Menu (exploring-option): Allows you to view stats, equipment, and items. Furthermore, you can change equipment. Moreover, you can drop items **or heal**. Will be given as user reaches level 10, **which is the reason for using the reference parameters to Avatar and Boss.**
 - Explore (exploring-option): Chance of finding and possibly collecting item. **If inventory is full, user will be allowed to replace an item in inventory.** There is a chance to random encounter with basic enemy. Nothing can occur.
 - Die: **Returns a int to determine if the game ends. Returns 1** when current HP <= 0. **Returns 0 otherwise.**
 - Level Up: Occurs **always after each fight win, actual leveling up occurs** when current EXP + get EXP(enemy) >= EXP cap. Increase max HP, EXP cap, attack stat, defense stat, speed stat, and evasion stat. Sets current HP to max HP. **Set current EXP to (current EXP + get EXP(enemy)) % EXP cap.**
 - Fight (friend): Occurs randomly when using the explore option. It's a friend function so that every enemy can easily receive the user's stats without using parameters. Also, enemies engage fights, not the user. **Defined in Enemy.cpp file.**

- **Equip:** Allows user to switch out equipment (or equip when having no equipment). Equipment will have a visible effect in fights and when viewing stats in menu.
- **Print Items:** Prints out a list of items in inventory. Furthermore, this list is written out to the "Inventory.txt" file.
- **Drop Item:** Drops an item in inventory.
- **Pick Up:** Allows a user to pick up an item from exploring or an enemy drop.
- **Valid Choice:** checks to see whether an entered character matches either of two required characters.

Items will be read in from text-files

- Weapons
 - Boxing Gloves (+10 Attack)
 - Diamond Pickaxe (+20 Attack)
 - Shotgun (+50 Attack)
 - Drill (+80 Attack)
 - **Time Harp (+100 Attack)**
 - Holy Lance (+130 Attack)
 - Bane of Light (+160 Attack)
 - Lightning Hammer (+200 Attack)
 - Lightsaber (+250 Attack)
- Armor
 - Red Cap (+10 Defense)
 - Overalls (+20 Defense)
 - Metal Helmet (+20 Defense)
 - Knight's Armor (+50 Defense)
 - Iron Suit (+160 Defense)
 - Mask of Terrible Fate (+100 Defense)
 - Yellow Jumpsuit (+200 Defense)
- Healing
 - **Sunrise Tomato (+50 HP)**
 - **Bandages (+100 HP)**
 - **Elixir (+150 HP)**
 - Med-Kit (+200 HP)

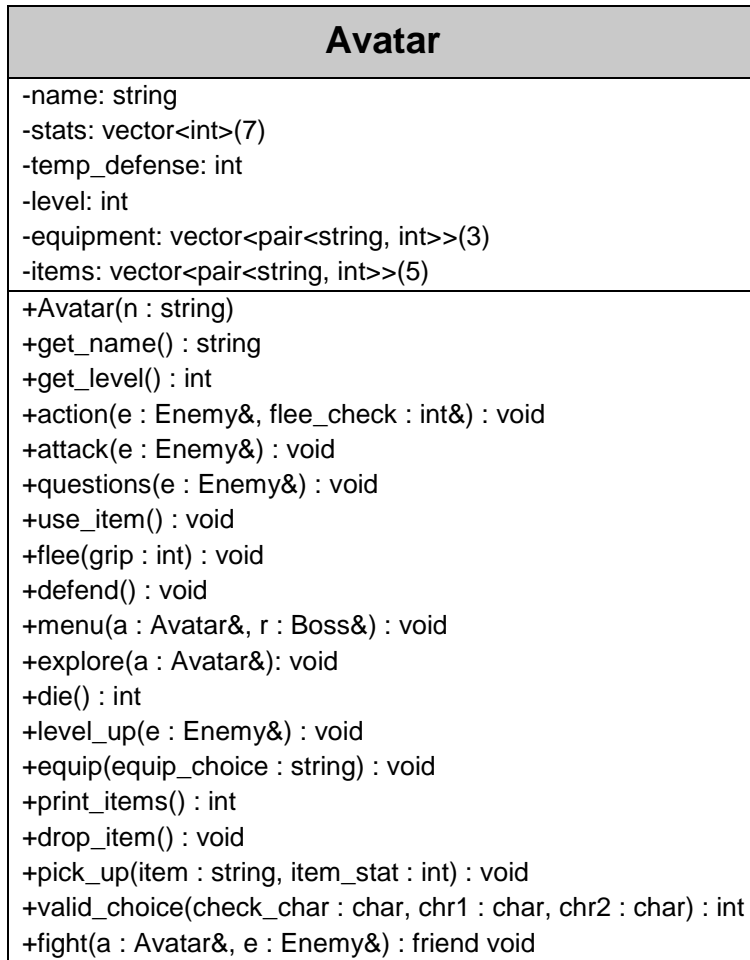
Enemies (will also be read-in from a text file) **(Stats in-file)**

- Filthy Peasant
- Noble Raccoon
- Walking Tree
- Whimsical Owl
- Mighty Shovel

- Ultra Ghost
- Purple Guy
- Elite Penguin
- Energetic Cat-Girl

End Credits

- Will be written to a text file and console.

UML diagrams

Enemy
#name: string #stats: vector<int>(6) #item: pair<string, int> #EXP : static int
+get_name() : string +get_drop() : pair<string, int> +get_EXP(): static int +attack(a_HP : int&, temp_defense : int, a_name : string) : void +use_item() : void +die() : int +operator[](index: int): int& +fight(a : Avatar&, e : Enemy&) : void

Enemies
N/A
+Enemies(avatar_lvl : int) +random_choice(a_stats : vector<int>&, temp_defense : int&, a_name : string) : void +rps(a_stats : vector<int>&, temp_defense : int&, a_name : string): void

Boss
N/A
+Boss() +random_choice(a_stats : vector<int>&, temp_defense : int&, a_name : string) : void +roll_dice(a_stats : vector<int>&, temp_defense : int&, a_name : string): void