

Project 1

Saturday, October 10, 2020 9:22 PM

The Game:

Player enters a dungeon and encounters monsters that she can either escape or defeat. The dungeon has rooms based on a map that connects the room to either one or two additional rooms. The player must make her way through the dungeon until she exits the dungeon.

The capabilities of the monsters are randomly generated with two main characteristics: speed and strength

The user can choose to either fight and defeat the enemy or make an escape. There is always luck involved with whichever choice the player makes. Should the player defeat a monster, there is a reward that improves the player's health.

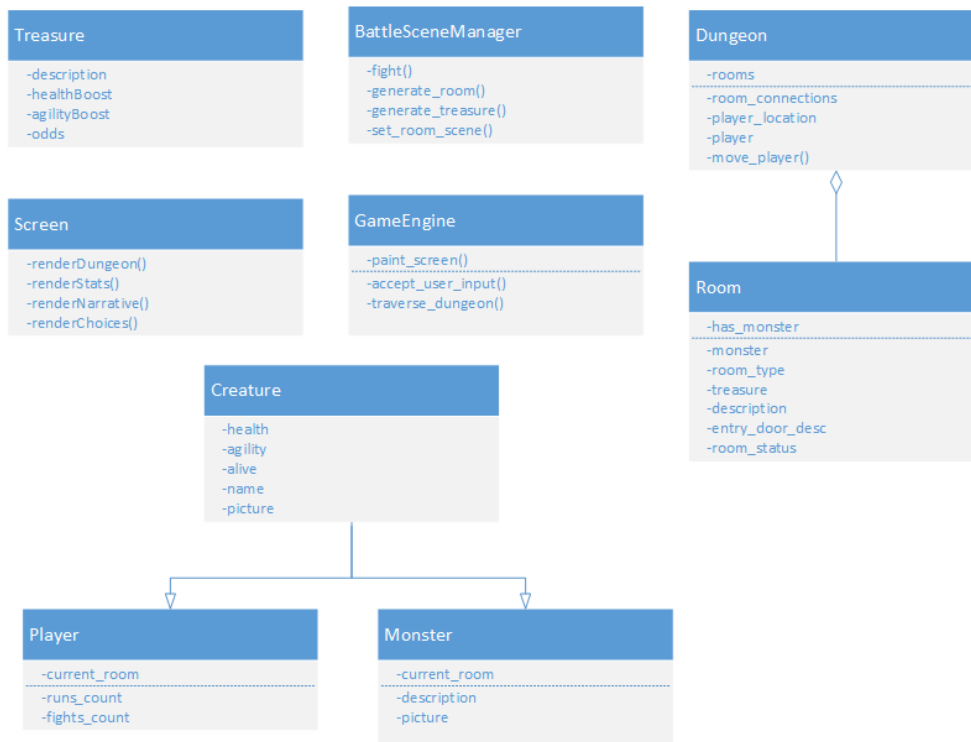
User Interface:

Player will always be shown her health status and the health stats of the monster

After or during every battle, the player will be given the option to fight or flee (which sometimes can be a smart thing to do) and will be given either one option (to proceed) or two options (choose a path).

Classes:

1. Dungeon
2. Room
3. Monster
4. Player
5. Treasure
6. Battle
7. Screen
8. GameEngine



Room

A room can either be empty, have random health points (a prize) or have a monster.

A room can also be of type exit which marks the end of the game

has_monster	Indicates if room has a monster. Only one, has_monster or has_treasure, can be true
has_treasure	Indicates if room has a treasure Only one, has_monster or has_treasure, can be true
room_type	Room can be either an exit, or entry or murky A murky room can have either a monster or a treasure in it
monster	Reference to the monster object in the room
treasure	Reference to the treasure object in the room
room_status	If room has been visited (and is now empty) or needs to be initialized This is useful when room is being rendered on the screen
description	Colorful description of the room
entry_door_desc	Colorful description of the entry door of the room

Dungeon

Contains rooms that are linked to either one or two other rooms

The dungeon map is crafted so that there is at max 6 rooms linked to each other

rooms	List of all rooms in the dungeon List is initialized when the class is initialized
next_rooms	Connections from one room to either one or two other rooms This map is created when the class is initialized
player	Reference to the player object
player_location	Room in which the player is right now
move_player	Updates location of player in the dungeon based on the user input
move_choice	Provides the rooms with entry door(s) that user must choose between using the door description

Creature

Contains common properties for all creatures in the dungeon whether human player or monster

alive	Is this creature alive or dead
name	Name of the creature
health	Health level
agility	Agility level

Monster

Generated randomly with random statistics for agility and health and associated with a room upon generation

Is sticky to one room, i.e., is a property of the room

description	Description of the monster
room	Room in which the monster lives
picture	(optional) path to image of the creature

Player

Traverses the DungeonMap

Resides in a room temporarily before moving to the next room

current_room	Reference to the room that the player is in right now
runs_count	Number of times player ran away from fights
fights_count	Number of times player engaged in fight with monster

Captures fun stats about the type of player

Treasure

description	Colorful description of the treasure
healthBoost	Max health boost
agilityBoost	Max agility boost

BattleSceneManager

Controls the battle sequence between the player and the monster

A battle can last one or two rounds depending on the stats of the monster or the player

fight()	Executes fight between player and monster Uses random odds to influence fight while keeping it generally reflective of the agility level of the player and the monster Agility influences the speed of the strikes, hence higher the agility, more likely the damage (reduction in health) is going to be Returns the outcome of the fight and updates player and monster with stats, updates the room status as well
set_room_scene())	Determines if the room will have a monster or treasure in it
gen_monster	Creates the monster including the stats
gen_treasure	Creates the treasure including the stats

Screen

The class that renders the DungeonMap and the Stats for the player at all times

renderDungeon	Shows the dungeon on screen
renderStats	Shows the stat bar for the player and monster (if room has monster)
renderNarrative	Shows the narrative (e.g. monster description, options on the two doors ahead)
renderChoices	Shows the questions to ask the user for next steps to take

GameEngine

paint_screen()	Calls the screen methods to paint the screen and set the scene for the player
accept_user_input()	Has the loop that presents user the options to select between (or to quit at any time)
traverse_dungeon()	Moves user through the dungeon by calling the dungeon class, player and rendering rooms