Class and Package Diagram for System Finding Trump

Assignment in the course PA1435 Object Oriented Design

2017-04-27

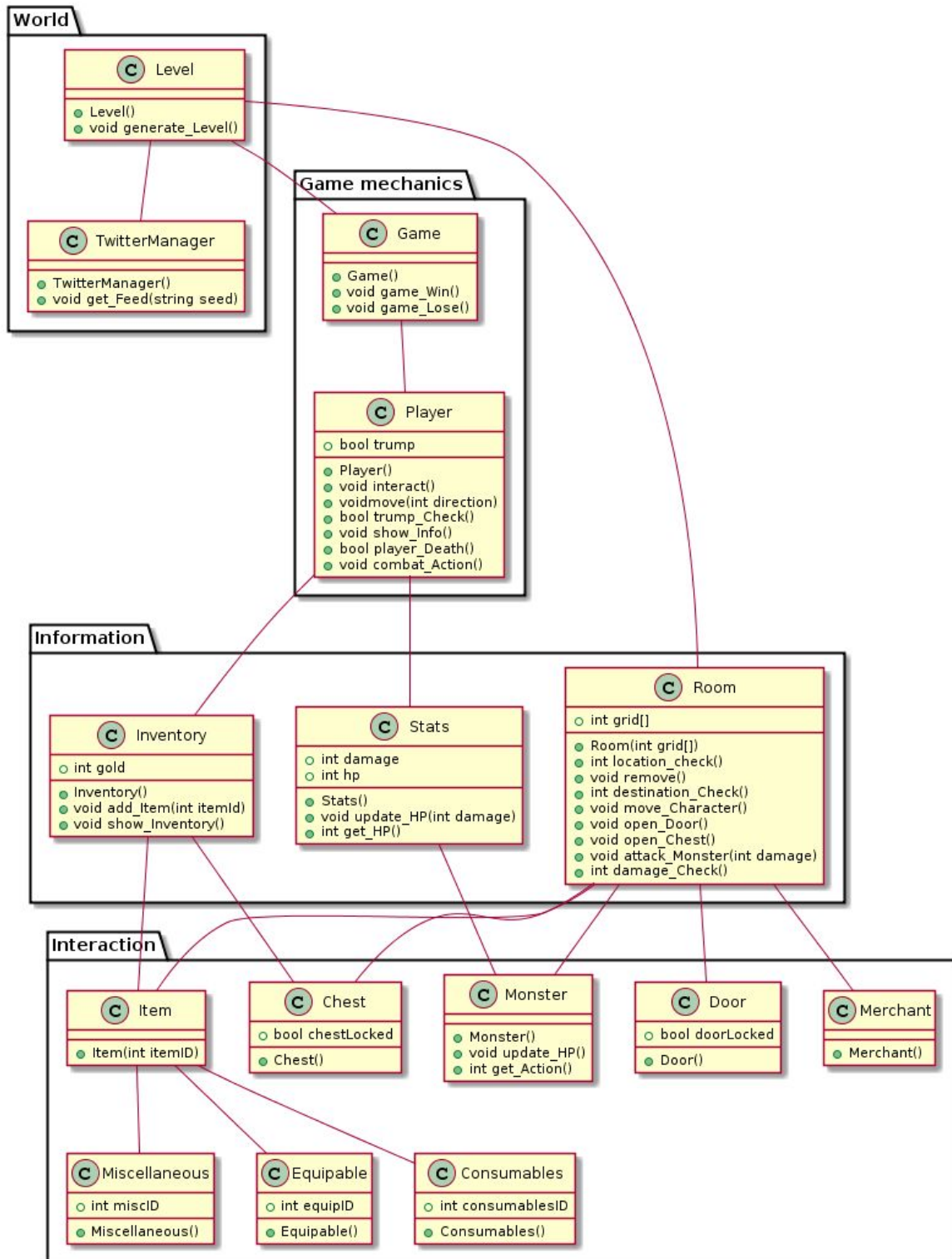| Author Name	Social Security#	Thinking	Writing
Emil Wahl	880729-0351	17%	17%
Fredrik Carlsson	961206-0070	17%	17%
Daniel Cheh	941231-1772	17%	17%
Aksel Grahn	940311-0159	17%	17%
Sebastian Vusak	960406-1110	17%	17%
Gustav Björk	960223-4131	17%	17%

System Description

The system is based on the classic game Nethack. Nethack is a text-based and two-dimensional game. It is a rogue dungeon crawler with a random generated map. In this dungeon you will be able to find treasure and fight monster.

The main difference between our system and Nethack is that our system is connected to Twitter and all the levels, items, and monsters are generated from a seed in the form of a Twitter feed.

Class and Package Diagram

Classes - Class Diagram

World

Level
- Level()
- void generate_Level()

TwitterManager
- TwitterManager()
- void get_Feed(string seed)

Game mechanics

Game
- Game()
- void game_Win()
- void game_Lose()

Player
- bool trump
- Player()
- void interact()
- voidmove(int direction)
- bool trump_Check()
- void show_Info()
- bool player_Death()
- void combat_Action()

Information

Inventory
- int gold
- Inventory()
- void add_Item(int itemId)
- void show_Inventory()

Stats
- int damage
- int hp
- Stats()
- void update_HP(int damage)
- int get_HP()

Room
- int grid[]
- Room(int grid[])
- int location_check()
- void remove()
- int destination_Check()
- void move_Character()
- void open_Door()
- void open_Chest()
- void attack_Monster(int damage)
- int damage_Check()

Interaction

Item
- Item(int itemID)

Chest
- bool chestLocked
- Chest()

Monster
- Monster()
- void update_HP()
- int get_Action()

Door
- bool doorLocked
- Door()

Merchant
- Merchant()

Miscellaneous
- int miscID
- Miscellaneous()

Equipable
- int equipID
- Equipable()

Consumables
- int consumablesID
- Consumables()

Interaction package

Contains classes which players can interact with in one way or another. Responsibility is to fill the world with objects to interact with.

Information package

Contains classes that holds information about entities in the game. Responsibility is to give information between different classes.

World package

Contains classes that hold and create the game world. Responsibility to generate and store the layout of the game world.

Game mechanics package

Contains classes for the core game mechanics. Responsibility to control and manage the game.