First assignment documentation

Le Minh Nghia - AAOGMU

October 10, 2018

Contents

I

Chapter 1

Assignment's description

Simulate a simplified Capitaly game. There are some players with different strategies, and a cyclical board with several fields. Players can move around the board, by moving forward with the amount they rolled with a dice. A field can be a property, service, or lucky field. A property can be bought for 1000, and stepping on it the next time the player can build a house on it for 4000. If a player steps on a property field which is owned by somebody else, the player should pay to the owner 500, if there is no house on the field, or 2000, if there is a house on it. Stepping on a service field, the player should pay to the bank (the amount of money is a parameter of the field). Stepping on a lucky field, the player gets some money (the amount is defined as a parameter of the field). There are three different kind of strategies exist. Initially, every player has 10000.

Greedy player: If he steps on an unowned property, or his own property without a house, he starts buying it, if he has enough money for it.

Careful player: he buys in a round only for at most half the amount of his money.

Tactical player: he skips each second chance when he could buy.

If a player has to pay, but he runs out of money because of this, he loses. In this case, his properties are lost, and become free to buy.

Read the parameters of the game from a text file. This file defines the number of fields, and then defines them. We know about all fields: the type. If a field is a service or lucky field, the cost of it is also defined. After the these parameters, the file tells the number of the players, and then enumerates the players with their names and strategies.

In order to prepare the program for testing, make it possible to the program to read the roll dices from the file.

Print out which player won the game, and how rich he is (balance, owned properties)
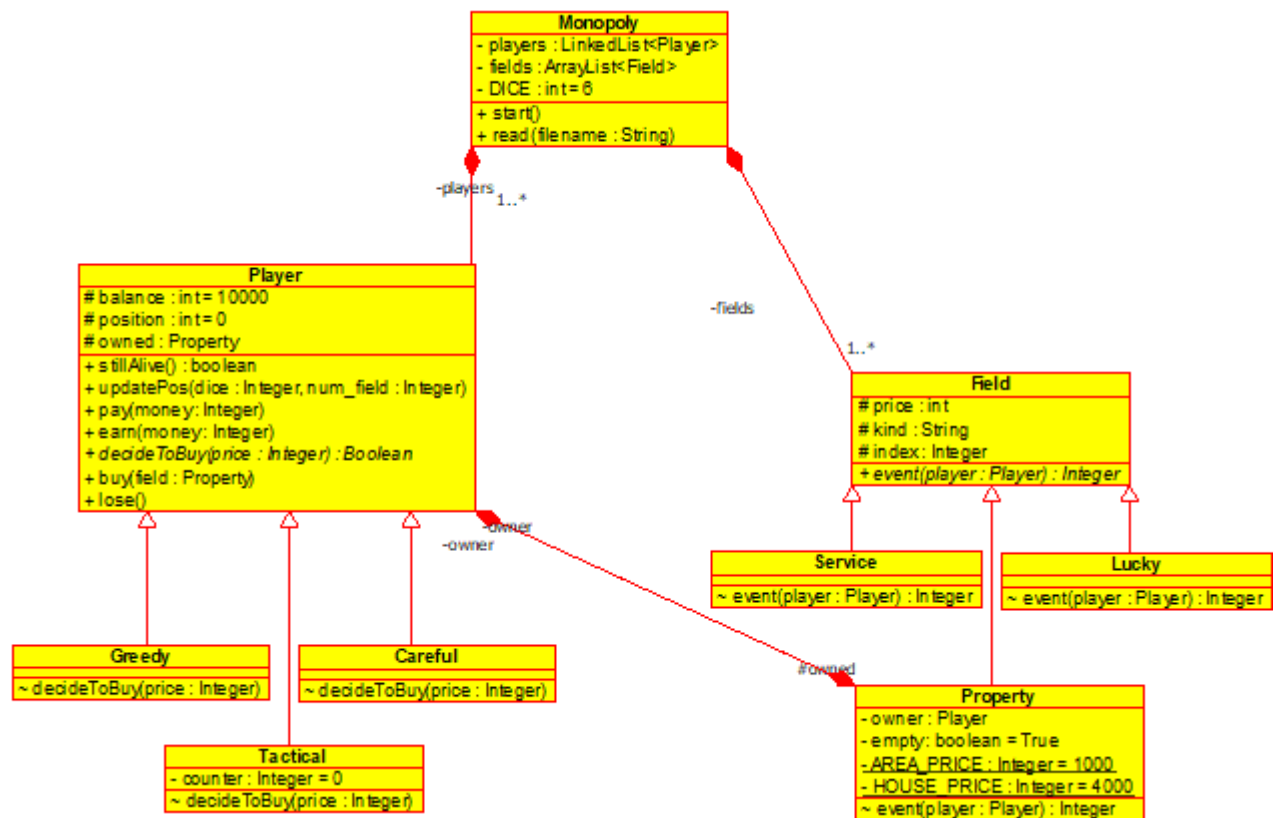
Chapter 2

Usage

The input will be read from *"assignments-1-input.txt"* which is placed in side the project folder. After running the Main, the winner of the game will be write to the console.

The input file defines the number of fields, and then defines them. We know about all fields: the type. If a field is a service or lucky field, the cost of it is also defined. After the these parameters, the file tells the number of the players, and then enumerates the players with their names and strategies.

For testing purpose, one can also put the roll dices in each turn at the end of the input file (these numbers can only be 1, 2, 3, 4, 5, 6).

Chapter 3

UML Diagram

Chapter 4

Methods documentation

- **Monopoly's methods:**

  start Starts the game with list of players and list of fields and keep playing until only one player left, then output that player as the winner of the game.

  read Reads and stores the data from file.

- **Player's methods:**

  stillAlive Determines whether the player went bankrupt or not.

  updatePos Updates the current position of the player after rolling dice.

  pay Decreases player's balance by a given amount.

  earn Increases player's balance by a given amount.

  decideToBuy The decision of a player when stepped on an empty property or his owned property but haven not built a house yet. This is an abstract method and it will vary depending on the strategy.

  - **Careful** Buys if the price does not exceed half of his money.
  - **Greedy** Buys whenever he has a chance.
  - **Tactical** Skips each second chance when he could buy.

  buy The process of spending money for buying/building.

  lose Gives back all of his properties after losing.

- **Field's methods:**

  event The event when a player stepped on the field. This is an abstract method and it will vary depending on the type of field and the decision of the player.

  > **Lucky** The player earns some money.
  >
  > **Property** If the field is empty or the player who stepped on is the owner of the field, then he will decide to buy/upgrade the field. Else the player must pay the stepping fee to the owner.
  >
  > **Service** The player pays some money.

Chapter 5

Testing