Riley O'Byrne

Anand Odbayar

Md Mahmud
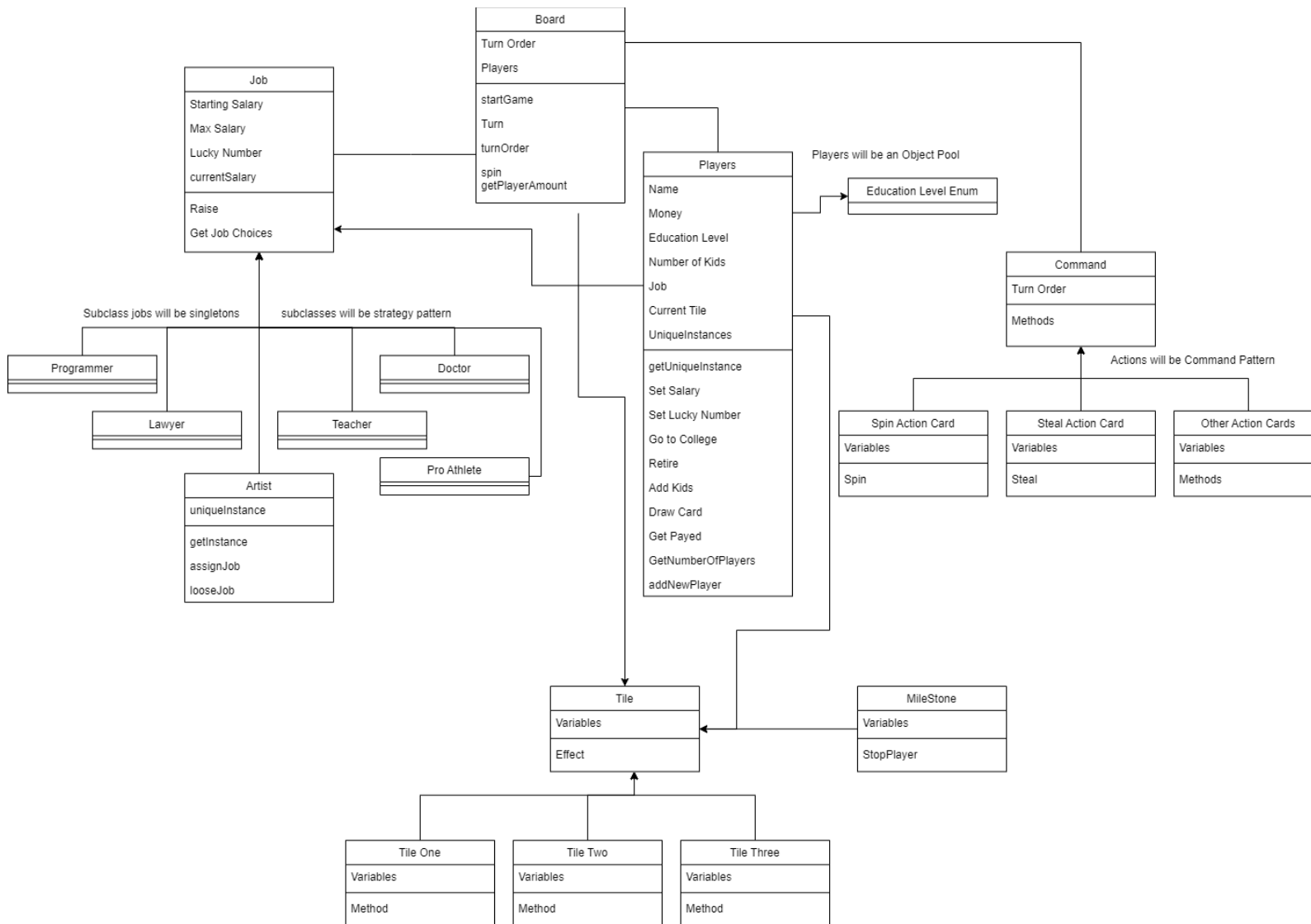
CSCI 4448 OOAD Project 6

Status Summary

- Please include the names of all team members and the title of the project

    - Team Members: Riley O'Byrne, Anand Odbayar, Md Mahmud

    - Project Name: The Game of Life

- Work Done

    - Job Class implemented (Riley) - Work done on creating an abstract job class. Classes that inherit from Job (ex Artist) will have all of the methods they need using a strategy pattern. Jobs are able to be assigned to a Player now.

    - Artist Class implemented (Riley) - Work done on creating the Artist Class using a singleton pattern, ensuring that only one player can have that job at a time.

    - Player Class created (Riley) - Created the Player class and implemented an object pool ensuring that only 4 players can exist.

    - Action Cards Class created (Md) - Work has been completed for action cards. Getters and setters have been implemented. The command pattern structure has been established for later use.

    - Board Class created (Anand) - The board class has the start game and turn methods

    - Player Class expanded (Anand) - Added variable pertaining to board class

- Changes or Issues Encountered

- ○ A small issue we encountered was that we were planning on having the player class be responsible for assigning jobs (i.e. player.assignJob(job)). However, we found it was easier with the singleton patterns of the individual jobs to do it differently (Artist.assignJob(p)).
- ○ We also Realized that we needed to track if a unique instance of a Job has been assigned to a Player. So we included that.
- Patterns
  - ○ We are using a Strategy Pattern for the Jobs so they can inherit methods that they need to function properly.
  - ○ We are using a Singleton Pattern for the individual jobs as well to ensure that only one player can have any given job at a time.
  - ○ We are using an Object Pool to ensure that only 4 players can exist at any given time.
  - ○ We are using a Command Pattern for all of the different Action cards.

Class Diagram

Board
Turn Order
Players

startGame
Turn
turnOrder
spin
getPlayerAmount

Job
Starting Salary
Max Salary
Lucky Number
currentSalary

Raise
Get Job Choices

Players
Name
Money
Education Level
Number of Kids
Job
Current Tile
UniqueInstances

getUniqueInstance
Set Salary
Set Lucky Number
Go to College
Retire
Add Kids
Draw Card
Get Payed
GetNumberOfPlayers
addNewPlayer

Players will be an Object Pool

Education Level Enum

Command
Turn Order

Methods

Actions will be Command Pattern

Subclass jobs will be singletons

subclasses will be strategy pattern

Programmer

Lawyer

Doctor

Teacher

Pro Athlete

Artist
uniqueInstance

getInstance
assignJob
looseJob

Spin Action Card
Variables

Spin

Steal Action Card
Variables

Steal

Other Action Cards
Variables

Methods

Tile
Variables

Effect

MileStone
Variables

StopPlayer

Tile One
Variables

Method

Tile Two
Variables

Method

Tile Three
Variables

Method

Can also be viewed here:

https://drive.google.com/file/d/1PBwTL1uUSLqxFot-93Yf1XOFgtZmD7s6/view

Plan for Next Iteration

- We need to finish up the small amount of work that remains for the Job and Player classes.

  - The other job classes will be almost identical to Artist, but we wanted to make sure we had that one right before working on the others.

- Finish the Board.

- Work on the GameOfLife Class, putting all of the pieces together.

    - Work on user interface (terminal prompts)

    - Put all of the pieces together.

- Setup action cards with a wide variety of cards available in the physical game. Allow players to have a list of cards they earn throughout the game, which they can later cash in at retirement.