

WIX1002 FUNDAMENTALS OF PROGRAMMING ASSIGNMENT

A. Problem specification

Your company has received several new projects this month. All these projects are equally important, with certain level of technical difficulties and challenges. You have been instructed by the CIO to lead a team of **4-5 members**, to handle one of the projects. You have been given **10 weeks** to complete the project and prepare a demo. Have fun and be creative! You will eventually sit down and be impressed by your own work! All the best!

B. Group formation

1. Form a team of 4-5 members (same group session) to do the project.
2. Every team member must contribute to the project, including certain amount of coding. The role played by and contribution of each member to the project must be included in the managerial report.

C. Submission

1. A technical report explaining the assigned project, the requirements of the project, the approach taken to solve the task, a detail description of your solution (includes the flow-chart, modules, etc.), sample snapshot of your program output.
2. A managerial report explaining the formation of the team, role and assigned work for each of the members, the project timeline, the problems faced in accomplishing this assignment and your solutions, and other issues arise.
3. The complete source code.

D. Important dates:

1. Submission: (Week 13) 12pm on (Tuesday, 12th January 2021)
2. Demo: (Week 13) during tutorial and lab session (Thursday, 14th January 2021)

E. Marking scheme: (15 marks)

1. Source code/Program. (12 marks)
 - a. Meeting the basic task requirements, i.e. identify the requirements of the project and produce a workable solution – **8 marks**
 - b. Extra features/functionalities – **4 marks**.
2. Report (**3 marks**)

Plagiarism is strictly PROHIBITED. You will be penalised if any of your codes is found online.

Project 1: Yuu-Tube



A. Introduction

Kamil is a video content creator, he has been filming videos and upload them on YouTube for more than 5 years. He has over 800k subscribers and a total of 10M views on YouTube. However, due to one careless mistake he made, his YouTube account has been permanently banned and he is not allowed to re-join the community.

Creating video is everything for Kamil. He spent his entire teenager life to learn video editing and has spent over RM 60k on his filming accessories. There is no going back because he loves video more than anything else!

Lucky for Kamil, a friend of him, namely Yuu, who is a professional software developer decided to help him out by creating an alternate online video sharing platform just like YouTube.

Although there's a dozen of existing YouTube alternative (eg. DailyMotion, Vimeo, Bitchute, etc.), but since Kamil already familiar with the User Interface(UI) and features of YouTube, so Yuu will make Yuu-Tube out of the original YouTube.

B. Problem Statement

The minimum features that **must** be included in the new Yuu-Tube are:

Features	Description	Hints / Suggestions
Login (Registration & Authentication)	<p>A new user is able to create a new account by providing some info such as email address, password and name to create a new Yuu-Tube account.</p> <p>Meanwhile, an existing user can login to the application by typing his credentials (email and password).</p> <p>Also, the user is able to log out once logged in.</p>	<p>You should be able to store the users' credential somewhere offline (or online if you able to do so). We don't want the user data to vanish after each reboot.</p>
Videos	<p>A video sharing platform must allow the user to upload new videos.</p> <p>Yuu-Tube should be able to support the following operations:</p> <ul style="list-style-type: none">- Upload video- Delete video- Play video	<p>Note that the video and action stated, should be REAL. Example of video format:</p> <ul style="list-style-type: none">- MP4- AVI- WMV- FLV <p>You can create a new folder to store all the videos uploaded by</p>

		a particular user.
Trending videos	Yuu-Tube homepage will show the top 5 trending videos from the entire application.	The trending videos will be sorted based on total views count.
Search	Yuu-Tube need to have the search feature. Which allow the user to search for either video or a channel. Besides, you will need to return other videos which have similar keywords / title. Sort them according to matching score.	You should have a list of all available videos title and user list to allow fast checking and searching.
User Operation	A registered Yuu-Tube user can perform the following operations: <ul style="list-style-type: none"> - Like & Dislike video - Comment on video - Subscribe to a user - Manage own account (change email, password, close account) 	Note that statistics for a video/user should be visible to the public. Example: Video: views, like dislike count, comments User: videos owned, subscriber count

Some of the attributes for your Video class and User class:

Video:

- title: String (Required)
- likeCount: Integer (Required)
- dislikeCount: Integer (Required)
- viewsCount: Integer (Required)
- comments: String[] (Required)
- trendingNow: Boolean (Optional)
- videoID: String (Optional)

User:

- name: String (Required)
- email: String (Required)
- password: String (Required)
- videosCount: Integer (Required)
- subscribersCount: Integer (Required)
- videos : Video[] (Required)
- totalViews: Integer (Optional)

You are free to add/modify attributes for your own need.

C. Sample Input & Output (CLI)

Homepage

```

----- Welcome to Yuu-Tube v1.0 ! -----

Trending on Yuu-Tube
#1 The Most Insane 900 IQ Among Us Outplay!
#2 Java Tutorial for Beginners [2020]
#3 TWICE "FANCY" M/V
#4 Blackpink - How you like that M/V
#5 BTS (방탄소년단) 'Dynamite' Official MV

```

User Login

What are you up to now? Please select [1-5][A-C]: A

Great!

Email: hello@gmail.com

Password: pass12345

----- Welcome back, user! -----

video Playing (Proof of Concept)

video poc.mkv (p/s: Double click to see the video, mobile is not supported)

D. Extra Features

Implement some of these features and watch your marks rocket high!

1. Implement GUI

Human loves graphical interface doesn't it? Try to implement a graphical user interface and design it using your own idea! There is a lot of tutorials and guideline all over the Internet, don't limit yourself to the lecture note.

2. Use Databases instead of text files.

Well, to be honest no one uses plain text files to store credentials, especially this is the new Yuu-Tube!

3. Login with real Gmail account.

Why don't we "steal" YouTube customer and bring them to here? Jk.

It is nice to have our application be able to support 3rd party account authentication.

4. Two Factor Authentication (2FA)

This is a really cool feature to have. Little tips: Randomly generate a number and send the number to the user's email (using SMTP).

5. Bring it online.

Maybe some of you are interested in hosting this application somewhere in the cloud? Go ahead :), some of the examples are: GCP (free trial available), AWS, Azure or even Google Drive.

6. Anything that's worth trying.

Beside the suggestions listed up there, you can implement your own ideas and extra features. This is a good chance for you to learn new things that are not in the textbook, so good luck and wish you all the best.

E. Questions

If you are not clear about the project, you can contact LEE CHENG HUI (chlee9926@gmail.com)
Subject: WIX1002 (Yuu-Tube)

Project 2: The Donald

A. Introduction

In 2019, there was a group of university students who studied Computer Science who were too bored of studying. One day, they gathered together and pointed out this problem. At that moment, one of them had popped out with a game which is "The Donald". It was a fun card game and requires lots of brain cells to be sacrificed if you wanted to win the game. It sounds interesting, and succeeded to attract the attention of the others, but then sadly they found that they didn't have the important tool to play it which is the card. "Hmmm, we could code this out in java couldn't we?", one of them said. "Well we could score fresh blood to play with if we make the juniors do it.", another kinda bad guy said. "Hehe yes, big brother can teach u how to have fun :)", they said.

B. Problem Statement

You are required to develop a program that would be able to play the game. The program should be able to be played with multiplayer without the internet.

i) Basic Requirement

The program should support multiplayers. By default, there should be 4 players who would be forming two teams each with 2 people. The program should be able to calculate the score of each player and define the winning point correctly. The game should also follow the rules stated below.

ii) Game rules

There is a card deck with 52 cards, each card with a different Rank or Color. The value of the card is compared by the color then only by the rank value. However, there is one more rule for this, for example if the first player played card BLUE 4, then all the following players should play BLUE card also if they have a BLUE card on their hand. If they don't have BLUE card, they can play any other color card, but the other card would be considered as having lowest value unless it is a "donald" card which is called at the beginning **if and only if the player who called for the donald have played the card already**. To have a better understanding, can refer to the output in the output section. Then **after the player has won a round of game, the next round would be starting from the player who won the round**.

Default Card Priority

Rank value : $1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < A < B < C$

At the beginning, after the cards are assigned equally and randomly to the players, they should start to call for donald (number of donald + donald value) to decide the winning criteria. **Call for donald** would change the winning criteria. For example, if I call for 3 RED, then my team would need to win 9 (6 + 3) pairs of cards to win the game while the other team only needs 5 (8 - 3) pairs of cards to win the game. Besides, if 3 RED is called then it means the RED card would be bigger than any other color card without considering their rank **if and only if the RED card is played by the player who called for it**. Same goes to other donald values except for **NO DONALD** which means the game will follow the default card priority. When calling for donald, if someone has called for a certain value like 1 GREEN, then the following player should call for a certain value that is higher than 1 GREEN, for example 2 GREEN, 1 BLUE, 1 YELLOW, or NO DONALD if they want to bid for it. The value of the donald called would be compared by number then only followed by color.

Donald Priority

Donald number: $1 < 2 < 3 < 4 < 5 < 6$

Donald color : GREEN < BLUE < YELLOW < RED < NO DONALD

Winning Criteria

Team who called for donald: $6 + \text{number of donald}$

Another team: $8 - (\text{number of donald})$

C. Sample

```
Player 1, please enter your name: MIN
Player 2, please enter your name: ONG
Player 3, please enter your name: BOO
Player 4, please enter your name: RICH
Constructing the cards in the deck...
52 cards have been constructed
Shuffling the card deck
Completed!

Assigning the cards to the player...
Assigned!
```

```
MIN want to check your card?[YES/NO]YES
YELLOW 2 | RED 6 | GREEN 7 | RED 8 | GREEN 8 | GREEN A | YELLOW 9 | BLUE 9 | BLUE 6 | GREEN 9 | YELLOW A | RED 3 | GREEN C |

MIN call for donald?[YES/NO]YES

Enter the number of donald:[1-7]
7
Enter the donald:[RED,YELLOW,BLUE,GREEN,NO]
RED

ONG want to check your card?[YES/NO]YES
RED 9 | YELLOW 7 | GREEN 6 | BLUE 4 | GREEN 8 | RED 4 | YELLOW 10 | YELLOW 3 | RED 2 | GREEN 4 | GREEN 5 | YELLOW B | BLUE B |

ONG call for donald?[YES/NO]NO

BOO want to check your card?[YES/NO]YES
BLUE 10 | BLUE 1 | RED A | YELLOW 8 | BLUE 3 | BLUE 8 | GREEN 10 | BLUE A | RED 10 | BLUE 7 | YELLOW 4 | RED C | YELLOW C |

BOO call for donald?[YES/NO]NO

RICH want to check your card?[YES/NO]YES
RED 1 | RED 8 | GREEN 2 | YELLOW 1 | BLUE 5 | BLUE 2 | YELLOW 6 | YELLOW 5 | GREEN 1 | BLUE C | RED 7 | RED 5 | GREEN 3 |

RICH call for donald?[YES/NO]NO
MIN would be the donald
MIN choose your teammate
By entering "Color Rank" of someone's card
RED 8
```

```
Donald is played: false
Donald number is: 7
Donald color is: RED
Donald player is: MIN
MIN's turn:
Please enter your command:
1 : Arrange Card
2 : Play card
3 : Show score(Excluding teammate's score)
4 : Show hand card
2
Which card do you want to play?(Starting from 0)
0
MIN has played YELLOW 2
```

Example of the game interface

Situation

Player 1 called for 7 RED as donald.

Player 1 called for a player who has card RED 8 as the teammate. So in this case, player 1 would team with player 4 and they would require to win $(6+7)$ which is 13 rounds to win the game while player 2 and player 3 would be in a team and they would require to win $(8-7)$ which is 1 round to win the game.

When the first player who would play the card plays let's say BLUE 9, then the other player should play the BLUE card also if they have a BLUE card in their hand.

Scenario 1:

Player 1 plays BLUE 8

Player 2 has a BLUE card, so must play either one of the BLUE cards also. For example, player 2 played BLUE 9

Player 3 has a BLUE card also, so must play either one of them. For example, player 3 played BLUE 10

Player 4 got no BLUE card, then he/she could play any card he/she wants. But if the card played by Player 4 is not a donald card, then the card would be having lowest rank instead. For example, donald card is RED. player 4 played YELLOW A, it is not a donald card. So yellow A would have the lowest rank in this case.

So in scenario 1, player 3 would win the round. Then, the game ends, player 3 and 2 won the game.

Scenario 2:

Player 1 plays BLUE 8

Player 2 has a BLUE card, so must play either one of the BLUE card also. For example, player 2 played BLUE 9

Player 3 has a BLUE card also, so must play either one of them. For example, player 3 played BLUE 10

Player 4 got no BLUE card, then he/she could play any card he/she wants. For example, donald card is RED. player 4 played RED 8, it is donald card. However player 1 (the person who called for donald) never played any RED card before so meaning the donald is not "activated" yet so it is still considered as lowest rank card.

So in scenario 2, player 3 would win the round. Then, the game ends, player 3 and 2 won the game.

Scenario 3:

Player 1 plays BLUE 8

Player 2 has a BLUE card, so must play either one of the BLUE card also.

Player 3 has a BLUE card also, so must play either one of them

Player 4 got no BLUE card, then he/she could play any card he/she wants. For example, donald card is RED. player 4 played RED 8, it is donald card and player 1 (the person who called for donald) played RED card before so it is considered as "activated". The RED 8 would be considered as having the highest rank. So in this case, player 4 would win the game.

So in scenario 3, player 4 would win the round. Then, the next round of the game would be started from player 4.

Really sorry about the sudden changes, but it is the quite important rule of the game, so please forgive me to do so. If you have any questions, feel free to ask me

D. Additional Challenge

- 1) Make some different game modes such as player vs computer.
- 2) Deploy a GUI (Graphic User Interface) for it.
- 3) Feel free to add in any other card games into it.
- 4) Make use of the Exception class to handle the input of users. For example, catch the InputMismatchException for the field where the user should enter an integer but he/she gave it a float value.
- 5) Any other features that you could think of.

E. Question

If you are not clear about the project, you can contact ONG HENG KIAT (onghengkiat105@gmail.com) Subject: WIX1002 (The Donald).

Project 3: iHealth

A. Introduction

Due to the increasing case of Covid-19, the clinic management decided to decrease the density of people in clinic at a particular time by developing a system named iHealth. iHealth's objective is to allow people to make appointments from home, and only come to the clinic at their designated time.

B. Problem Statement

Basic Requirements of iHealth

iHealth is divided into two parts, which consists of:

Patient side

1. Accepts reservation from patients

Patients should be able to make reservation, at their chosen day and time. The system will then offer a time as close to the chosen time as possible for the patient and queue the patient if patient accept the offer. The system should consider the capacity of clinic, which means it should avoid the clinic being too crowded.

2. Registration

Patients should be able to register their profile if they are using the system for the first time. For example, their I/C, name, phone, and etc.

3. Check appointments

Patient should be able to check the date and time of their appointment by providing their details such as name and I/C or appointment number.

Clinic side

1. Manage appointments

Clinic should be able to manage the appointments by listing, viewing and editing them.

2. Manage opening/closing time

Clinic should be able to set the time/date available for patients to make appointments.

C. Sample Input/Output

```

~~~~~ iHealth v0.0.1 ~~~~~
How can I help you today:

1. Make Appointment
2. Check My Appointment

Please enter you choice: 1

Is this your first time making appointment ?

Please enter Yes/No: Yes

Please enter your full name: Mr. Roger

Please enter your I/C No. : 000102031234

```

```

~~~~~ iHealth v0.0.1 ~~~~~
How can I help you today:

1. Make Appointment
2. Check My Appointment

Please enter you choice: 2

Please enter your full name: Mr. Roger

Please enter your I/C No. : 000102031234

You have 1 appointment(s)

Date: 12/10/2020
Time: 1430

```

```

~~~~~ iHealth Clinic Admin v0.0.1 ~~~~~
How can I help you today:

1. List Appointments
2. List Patients
3. Set opening/closing time

Please enter you choice: 1

List of appointments

Name           Date           Time
Mr. Roger      12/10/2020     1430

```

D. Extra features

1. Implement GUI for iHealth
2. Notification for patient so that they won't forget their appointment
3. Allow multiple instances of application to run at the same time.
4. Allow the application to run on multiple computers (all instances should have the same data)
5. Can you solve the race problem which occurs when multiple instances are making appointments at the same time, which may cause the number of appointments to exceeds the actual available slots?
6. Any reasonable feature to show off your programming skills.

E. Questions

If you are not clear about the project, you can contact JOHN LAW SIE GIONG (johnlsg9909@gmail.com) Subject: WIX1002 (iHealth)

Project 4: Monopoly X RPG



A. Introduction

" Monopoly is a board game currently published by Hasbro. In the game, players roll two six-sided dice to move around the game board, buying and trading properties, and developing them with houses and hotels. Players collect rent from their opponents, with the goal being to drive them into bankruptcy."

- [Monopoly \(game\) \(Wikipedia\)](#)

" A role-playing game (sometimes spelled roleplaying game; abbreviated RPG) is a game in which players assume the roles of characters in a fictional setting. Players take responsibility for acting out these roles within a narrative, either through literal acting, or through a process of structured decision-making regarding character development."

- [Role-playing game \(Wikipedia\)](#)

Emma and Ryan are boy-girl twins, both of them are your best friends. As their best friend, you know that Emma is a board game enthusiast while Ryan loves to play role-playing video games. Their birthday is on 7th January and you have decided to present a gift to them. However, purchasing a board game and a video game as gifts will cost you an arm and leg. While you are worrying about the gift, an idea comes to your mind — create a whole new game by fusing the element of board game and RPG. Therefore, you would like to develop a game by using the programming skill you have learnt, but keep in mind that there are only a few months away from their next year birthday. Good luck!

B. Problem Statement

You are required to create a brand new Board Game mixed with the element of RPG system in CLI using Java programming language.

Layout:

A 9 x 9 grid with total 32 tiles surrounded.

CHEST	SIN-M	DUO-M	TRI-M		SIN-M	DUO-M	SIN-M	SHOP
TRI-M	MESSAGE: [Player 1] moved 4 steps. Now is [Player 2]'s turn, please roll the dice.							SIN-M
SIN-M								DUO-M
SIN-M								DUO-M
DUO-M								TRI-M
SIN-M	COMMAND: 1 Roll dice							SIN-M
DUO-M	2 Check stats							SIN-M
	3 Quit							
SHOP	SIN-M	SIN-M	DUO-M	1	SIN-M	TRI-M	DUO-M	<- START 2 3 4
Input:								

(Sample output)

Tiles:

Type of Tiles	Number of Tiles	Description
START	1	Starting tile of every player. When player passes over this tile (finish a circle), his Level will increase by 1.
SHOP	2	When player lands on this tile, he can buy items, sell items and upgrade his weapon using Gold.
CHEST	1	When player lands on this tile, he can open a treasure chests and obtain amount of Gold or item.
<EMPTY>	4	When player lands on this tile, he can heal himself using items or do nothing.
SIN-M	12	When player lands on this tile, he will encounter one monster and trigger a battle.
DUO-M	8	When player lands on this tile, he will encounter two monsters and trigger a battle.
TRI-M	4	When player lands on this tile, he will encounter three monsters and trigger a battle.

Player's basic statistic

Stats	Initial Value (suggested)	Description
Level	1	Represents how powerful a player is, higher the level, higher the other stats such as HP, Strength, Defence and Agility.
HP	25	Represents the amount of damage a player can take, a player lose the game when his HP reaches 0.
Strength	5	Determines the amount of damage inflicts on the enemy.
Defence	5	Determines the amount of damage receives from the enemy.
Agility	5	Represents how agile a player is, determine the chance of fleeing from battle and the chance of evading an attack.

EXP	0	A measurement required to increase the Level by 1 when reaching certain limit.
Gold	200	Represents in-game money use to buy items or upgrade weapon from shop.

Game rules/mechanics:

1. Number of players: 2 - 4
2. An unbiased six-sided dice is used.
3. Every player begins with same stats.
4. Starting by every player rolls the dice, highest number go first while lowest number go last.
5. Number of steps taken by player is determined by the dice, one roll per player's turn, no special move or 'roll the dice twice'.
6. Battle (read **Battle system** section for more details):
 - a. Battle against monsters occurs when a player lands on SIN-M, DUO-M or TRI-M tile **immediately**.
 - b. Battle against player occurs when 2 and only 2 players land on same tile. If another player lands at that same tile again (3 players on same tile), no battle is triggered.
 - c. No battle occurs at START, CHEST and SHOP tiles.
7. End game conditions:
 - a. Only one player left in the game
 - b. All players decide to quit the game, the player with highest Level followed by amount of Gold will be the winner.

C. Basic requirement

1. Game interface
 - Game board
 - An area to show the entire game board, including tiles and players' position
 - Prompt message
 - An output text area to display instructions, action taken by players, enemies state etc.
 - Command
 - An output text area to display available option for player to perform an action
 - Current player statistic
 - An output text area or option to display current player's stats

(The sample output is just for reference, you can design your game interface)

2. OOP concept
 - Create a Java abstract class named "Role" with relevant data fields and abstract methods declared. The relevant data fields and abstract methods are related to stats mentioned in **Player's basic statistic** and **Battle system** sections. This abstract class must be extended by 2 or more Java classes. (Figure it yourself!)
3. Battle system
 - Uses turn-based system, e.g. Player 1 -> Monster 1 -> Monster 2 -> Player 1 -> ...
 - One action per battle turn.
 - Basic action options in battle:
 1. Attack - Deals damage to enemy

2. Item - Let player uses his items
 3. Flee - Escapes from battle
- Monsters have their own stats such as Level, HP, Strength, Defence and Agility.
 - Formulae:
 - Damage formula - To calculate amount of damage inflicts based on player's Strength and target's Defence

E.g. $\text{Damage} = \text{Strength} * (X / (Y + \text{Defence}))$ where X and Y are constant values

 - Other formulae to make the battle unpredictable, read **Probabilities** section.
 - Battle against monsters
 - Player always takes the first turn in this battle.
 - Battle ends when:
 - all encountered monsters are defeated
 - player flee from battle
 - player's HP reach 0
 - Rewards for the player succeeded in this battle are Gold, EXP and drop item.
 - Battle against player
 - Higher priority than Battle against monsters - When Player A lands on a SIN-M tile, Player A will have battle against monsters first. On the next turn (after Player A finish the battle), Player B rolls and lands on the same tile as Player A, battle against monsters will not trigger, instead battle against player will start immediately. Player B will take the first turn in this battle.
 - Both players must have reached at least minimum Level 5 in order to trigger this battle.
 - The only way a player can flee from this battle is using "Smoke bomb" item (no flee action option)
 - Battle ends when one of the players:
 - defeated (HP reaches 0)
 - flee from battle (no reward for both of the players)
 - Rewards for the winner are Gold and EXP.

4. Level-up system

- EXP threshold increases for each subsequence Level.

E.g:

Level	Required EXP
2	100
3	150
4	280

- When player levelling up, some of his stats values must have some increment.

5. Various items, weapons and monsters

- Sets some purchasable items such as "Potion", "Hi-Potion" (healing item), "Smoke bomb" etc. Furthermore, set some rare items which can only obtain from treasure chest.
- Other than upgrading default weapon which is Sword, player can buy different types of weapons for instance Hammer, Spear, War Axe, etc. which give extra value to player's stats (with trade-offs) or special ability.
- Players are able to deal with different type of monsters (some monsters will have higher HP, while others may have higher Strength or Agility).
- You can search for yourself or design your own items, weapons and monsters.

6. Probabilities

- Accuracy/Evasion
 - Create a formula to determine the accuracy or evasion of an attack, use player's stats and enemies' stats as variables for this formula.
- Flee
 - The probability of escaping a battle against monsters depends on player's Agility value and monsters' Agility value.
 - Using "Smoke Bomb" item will guarantee 100% of chance for escaping any battle.
- Item drop system
 - Integrate an algorithm to calculate the probabilities of getting an item after defeating the monsters.

7. Apply and implement given game rules

- Build your own codes which satisfy all the game rules stated above. Try to avoid repetitive code segments.

D. Additional Challenges

Extra Features

1. GUI

Make the game more attractive, try to implement graphical user interface to provide visual impact. You can put some animations and sound effects into your game. (Tips: JavaFX)

2. Online multiplayer

Cloud base service? Socket programming? Explore it yourself!

3. Bot

If cannot make the game into online, try implement a game bot so you can play alone. 🤖

4. Save & Load system

Player can save the game and resume it later. Uses Java IO knowledge you have learnt in lectures to accomplish this features. File encryption is another extra point, you can implement it to protect your saved files.

5. Random arrangement

The arrangement of tiles in game board is generated randomly for every new game.

6. Packaging your game

Learn how to deploy your product, make your game into a runnable application. (Tips: jar, jlink, JDK 14 jpackage, Launch4j)

7. Any other extra features that can impress us or make the game more thrilling.

E. Questions

If you are not clear about the project, you can contact TING WEI JING (tingweijingting2000@gmail.com) Subject: WIX1002 (Monopoly)

Project 5: Attack on Titan – Parallel World



Several hundred years ago, humans were nearly exterminated by giants. Giants are typically several stories tall, seem to have no intelligence, devour human beings and, worst of all, seem to do it for the pleasure rather than as a food source. A small percentage of humanity survived by walling themselves in a city protected by extremely high walls. Fortunately, the weapons were invented in recent years found effective to kill the titans.

Problem statement

As a programmer, you need to build a tower defense game in **JAVA** which allows the player to protect the residents within the wall from the titans.

Basic Requirements

Game Interface

```
The Game Started

No enemy yet

On The Ground
Row
0
1
2
3
4
5
6
7
8
9
    HOUR 0
    Coin: 40

-- -- -- -- -- The Wall
  0 1 2 3 4 5 6 7 8 9 Index
50 50 50 50 50 50 50 50 50 50 HP
-- -- -- -- --

Player's turn
Choose the weapon(s) you would like to upgrade (Type a string of integer or hit Enter to skip)
|
```

Wall

- 10 unit walls, each unit has 50 HP(Hit points/Health)
- Whenever any wall unit has it's HP equal to 0 or lower, the player lose

Coin

- The only resource in the game
- Initial number = 50 coins
- add 5 coins every hour

Weapon

- There's weapons on the walls
- Each wall unit has 1 weapon
- Weapon has 3 levels, different levels come with different attack points
- Level 1 = 2 attack points, Level 2 = 5 attack points, Level 3 = 10 attack points
- To upgrade the weapon to level 1, the player has to pay respective number of coin based the damage of the next level
- To upgrade the weapon into level 1(attack = 2 points), 2 coins required
- To upgrade to level 2(attack = 5 points) 5 coins required
- To upgrade to level 3(attack = 10 points) 10 coins required
- Visualize the weapons on walls

Ground

- 10 units X 10 units
- In front of the walls
- Titans only can position on the ground

Titans

- There are 2 types of titans
- Colossus Titan and Armoured Titan
- They take damage when there's weapon on the lane the titan position at
- Overlapping Titans printed as "AC"

Colossus Titan

- Only show up on row 9
- Show up after 5 hours
- HP = 50, Attack = 10 points
- Move left or right randomly

Armoured Titan

- Only show up on row 0
- Show up after 5 hours
- HP = 100, Attack = 5 points
- Move closer to walls after every hour (randomly left or right)
- When he reaches row 9(in front of the wall), Armoured Titan destroys the weapon on his lane
- When there's no weapon on the wall where Armoured Titan position at, he moves to closer wall with weapon by 1 unit
- When he's moving, he won't damage the wall
- When there's no weapon on the wall, Armoured Titan starts to damage one wall unit, not moving left and right

Game Rules

- The Game started on Hour 0
- Print the wall and ground
- For each hour, the enemy and the player takes turns
- After enemies' turn, the player will be given options
- First question: Choose the weapon(s) you would like to upgrade (Type a string of integer or hit Enter to skip)
- Second question: Do you want to upgrade all walls? (press 1 if yes, press Enter if no) Current coin number: [current coin number placeholder]
- Third question: How many HP do you want to add up to the wall(s)? Current coin number: [current coin number placeholder]

- If the player chooses walls in the third question(e.g. the player type 1234 indicating that he/she want to add the HP of wall units 1, 2, 3 and 4.), show the question: How many HP do you want to add up to the wall(s)? Current coin number: [current coin number placeholder]
- If the coins are not enough to upgrade all weapons selected, the system will try to upgrade as many weapons as possible

Example

```
Choose the weapon(s) you would like to upgrade (Type a string of integer or hit Enter to skip)
1110022229
Not enough money!
Not enough money!
On The Ground
Row
0
1
2
3
4
5
6
7
8
9
    **
    ** ** **
    ** ** **
    -- -- -- -- -- -- -- -- --
    0  1  2  3  4  5  6  7  8  9
    50 50 50 50 50 50 50 50 50 50
    -- -- -- -- -- -- -- -- --
    The Wall
    Index
    HP
```

```
Player's turn
Do you want to upgrade all walls? (press 1 if yes, press Enter if no) Current coin number: 7
|
```

```
Player's turn
Choose the wall(s) that you would like to upgrade (Type a string of integer or hit Enter to skip)
123
On The Ground
Row
0
1
2
3
4
5
6
7
8
9
    **
    ** ** **
    ** ** **
    -- -- -- -- -- -- -- -- --
    0  1  2  3  4  5  6  7  8  9
    50 50 50 50 50 50 50 50 50 50
    -- -- -- -- -- -- -- -- --
    The Wall
    Index
    HP
```

```
Player's turn
How many HP do you want to add up to the wall(s)? Current coin number: 17
|
```

Player's turn
 How many HP do you want to add up to the wall(s)? Current coin number: 17
 7 6 5
 Your money is not enough
 On The Ground

```

Row
0
1
2
3
4
5
6
7
8
9

    **
  ** ** **
  ** ** **          **
-- -- -- -- -- -- -- -- --
  0  1  2  3  4  5  6  7  8  9
50 57 56 50 50 50 50 50 50 50
-- -- -- -- -- -- -- -- --

The Wall
Index
HP
  
```

The titan takes 5 damage
 The titan remains HP: 45
 The titan takes 2 damage
 The titan remains HP: 98
 On The Ground

```

Row
0
1
2
3
4
5
6
7
8
9

    AA          CC
    **
  ** ** **    **
  ** ** **    **          ** **
-- -- -- -- -- -- -- -- --
  0  1  2  3  4  5  6  7  8  9
50 57 56 50 40 50 50 50 50 50
-- -- -- -- -- -- -- -- --

The Wall
Index
HP
  
```

Coin + 5
 Enemies' turn(Enter to proceed)
 |

Challenges

- GUI Implementation
- Hard Game Mode
 - o Let the player chooses the game difficulties. The above game mode is Easy, make a Hard Game Mode. Multiple Armoured Titan in Hard Game Mode!
 - o No overlapping titans in Hard mode(no more combining Armoured Titan, AA and Colossus Titan, CC into AC)
 - o Provide faster coin generation speed(Level 0,1 and 2), coin generator in the above game considered level 0(default), to level 1(generate 10 coins per hour), player need to spend a certain number of coin, to level 2(generate 15

coins), need a certain number of coins too (more than coins required in level 1).

- Add obstacles on the ground for titans to hide
- Make Armoured Titan tend to move to walls without weapon before arriving the wall
- Different types of weapons
- Different types of titan with different behaviors
- Let the player to be on the site of titans, player decide the numbers and show up timing of titans while the computer try to build the weapon to defeat the player
- Any innovations to make the game fun and impress us!

Questions

If you are not clear about the project, you can contact CHOOI HE LIN (helinchooi@gmail.com) Subject: WIX1002 (Attack on Titan)

Project 6: Donald Duck's Digital Data Ducts

A. Introduction

=====
From: Mickey Mouse (CEO)
To: All Employees
Cc: All Company Executives
Date: 10:15 a.m., October 22
Subject: Dissolving Mickey Mouse Clubhouse

To fellow employees of Mickey Mouse Clubhouse,

We regret to inform you that we will be closing the Mickey Mouse Clubhouse as of (date). The general business decline has made it impossible to justify keeping the company open to the relevant stakeholders. To give you some context, as we all know during the COVID situation, the closing of our amusement parks has largely reduced our cash flow, combining with upcoming competition from alternative streaming platforms such as YouTube and TikTok videos which has captured a significant portion of our market share.

We thank you and wish you the best of success in your new positions, wherever it may be.

Mickey Mouse,
CEO, Mickey Mouse Clubhouse

=====
And that is how the once lovable pantsless entertainer, became jobless. Desperate to reclaim his former glory in the social ladder, he decides he's going to take up the coolest job of the decade, Data Scientist (debatable, chicken rice sellers are pretty much elite class in the UM hierarchy). But he is new to programming so he can't do it from scratch. He needs a library. And you'll be the ones to build it for him.

B. Problem Statement

You are required to develop a library to help Donald on his Data Science journey. Inputs and outputs will be in CSV format. CSV stands for comma separated values, meaning the values are separated by a comma as delimiter. Each line is a row and each value in the row is a column value. Sound familiar? It is basically Excel. In fact, opening CSV files on a windows computer will open it in excel. The first row of the CSV file usually contains the column labels.

Example CSV:

```
Name,Department,CurrentCGPA,Expected Graduation salary,Actual graduation salary
Meow,Artificial Intelligence,3.7,1000000,1000
Woof,Software Engineering,2.0,4200,4200
LWY,Information Systems,4.3,1000,1000000
```

Excel equivalent:

	A	B	C	D	E
1	Name	Department	CurrentCGPA	Expected Graduation salary	Actual graduation salary
2	Meow	Artificial Intelligence	3.7	1000000	1000
3	Woof	Software Engineering	3	4200	4200
4	LWY	Information Systems	4.3	1000	1000000
5					

The actual CSV specification is a bit more complicated than this as it is possible for a value to be a string containing commas which would ruin the strategy of simply splitting by comma. **However, to keep this question fun, you can assume that splitting by commas is enough to tokenize the**

line in the context of this assignment. In the spirit of having fun, you are also only required to parse integers, floats and strings.

The library needs to contain:

1. DataFrame Object

A DataFrame object to interface with the CSV file parsed.

a. Method to save DataFrame to CSV file

b. Method to construct DataFrame from CSV file

2. Manipulation methods

a. Method to concatenate DataFrames.

Column concatenation means stacking columns.

Fruit	Quantity		Color		Fruit	Quantity	Color
Potato	420	+	Potato Color	→	Potato	420	Potato Color
Tomato	404		Red		Tomato	404	Red
Plato	1		Platato Color		Plato	1	Platato Color

Row concatenation means stacking rows.

Fruit	Quantity	Color		Fruit	Quantity	Color
Potato	420	Potato Color	+	Potato	420	Potato Color
Tomato	404	Red		Tomato	404	Red
				Plato	1	Platato Color

If the combining axis doesn't match, the program should provide an error to tell Donald he made a mistake.

Fruit	Quantity	Color		
Potato	420	Potato Color	+	✗ Unequal number of columns
Tomato	404	Red		

Fruit	Quantity	Color	Dead
Plato	1	Platato Color	Yes

b. Method to obtain a subset of DataFrame with range of row or column. Rows are 0 indexed and the range is inclusive of the first element but exclusive of the last.

DataInFirstExample.rowRange(1,3)

Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Woof	Software Engineering	2	4200	4200
LWY	Information Systems	4.3	1000	1000000

DataInFirstExample.colRange(new String[] {"Department","Actual graduation salary"})

Department	Actual graduation salary
Software Engineering	4200
Artificial Intelligence	1000
Information Systems	1000000

c. Method to sort the rows by a column in the DataFrame.

DataInFirstExample.sort("CurrentCGPA")

Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Woof	Software Engineering	2	4200	4200
Meow	Artificial Intelligence	3.7	1000000	1000
LWY	Information Systems	4.3	1000	1000000

d. Method to remove duplicate rows based on subset of columns. There should be a parameter to choose whether to keep the first, last, specific number or no occurrence.

Data.dropDuplicates(new String[] {"Name"}, "first")

Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Emily	Software Engineering	2	4200	4200
Emily	Artificial Intelligence	3.7	1000000	1000
Boomily	Information Systems	4.3	1000	1000000

Keep = "first"



Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Emily	Software Engineering	2	4200	4200
Boomily	Information Systems	4.3	1000	1000000

e. Method to remove rows containing missing data in subset of columns

Data.dropNull(new String[] {"Department", "Actual graduation salary"})

Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Woof		2	4200	
Meow		3.7	1000000	1000
LWY	Information Systems		1000	1000000



Name	Department	CurrentCGPA	Expected Graduation	Actual graduation salary
Meow		3.7	1000000	1000
LWY	Information Systems		1000	1000000

3. Statistics package and imputers

a. Method compute variance, standard deviation, min, max, mean, median, mode and range of a column (Non numeric columns will only have mode)

b. Method to fill in missing values of specified columns with a specified value.

4. Scalers

Example values = {3.48, 2.30, 3.61, 3.16, 3.56, 2.9, 3.99, 4.87, 3.91, 6.28}

a. Method to perform Standard Scaling.

Standard scaling is subtracting the mean from all values in the column and dividing by the standard deviation.

Output if done on example = {-0.31070398, -1.43533803, -0.18680362, -0.61568949, -0.23445761,

-0.86349021, 0.17536667, 1.0140768, 0.09912029, 2.35791918}

b. Method to perform Min Max Scaling

Min max scaling is subtracting the min from all values in the column and dividing by the range

Output if done on example = {0.29648241, 0.000, 0.32914573, 0.2160804, 0.31658291, 0.15075377, 0.42462312, 0.64572864, 0.40452261, 1.000}

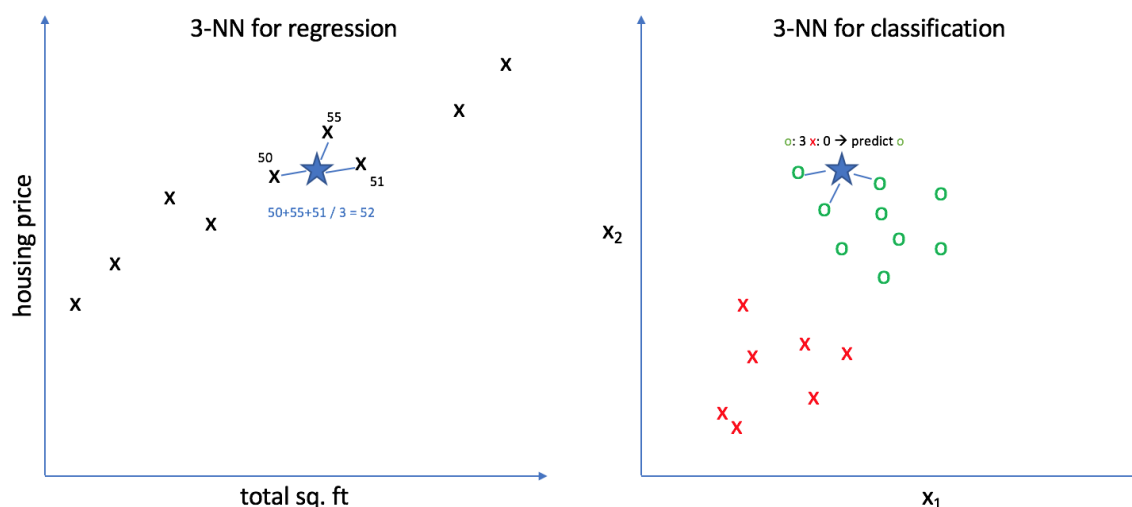
The proceeding requirements can assume the DataFrame is made up entirely of float columns.

5. K-Nearest Neighbors (k-nn)

K-nearest neighbors is a simple prediction algorithm that uses k-nearest known instances to try and predict unknown instances. For this question, do k-nn based on euclidean distance. Euclidean distance between 2 points is the root mean squared of differences between 2 points.

Regressor - output the mean of the instances

Classification - output the mode of the instances



a. Method to impute values for a column using **k-nn regressor** using subset of other columns.

b. Method to impute values for a column using **k-nn classifier** using subset of other columns.

6. Error metric

Being able to predict is no use if you don't know how often your predictions are correct.

Implement at least **2 error metrics for the regressor** and **2 error metrics for the classifier**.

Donald doesn't like errors so he isn't going to be particular about what you choose, he wants you to explore them on your own and implement what you think he should have.

Donald is also very particular about a few things:

1. Package management

Donald is not a fan of one file spaghetti code. You are required to practice proper **subpackaging and file management** in your library. (At least split each numbered requirement into its own file. though it is recommended you group them into subpackages to more easily conceptualize your library, especially as you start implementing additional challenges)

2. Proper commenting and documentation

Donald is a little dumb and is bad at logic. Do leave some **comments** so he doesn't get lost! (You don't need to put comments everywhere, just enough to describe what you intended with the method/subroutine.)

You should also document all the classes and methods in your library in one **"user manual"** to help Donald navigate and operate your library.

Use of **external libraries is prohibited** for implementations of the basic requirements of this question. They are allowed for implementations of the additional challenges as to not limit your creativity with what you can do, please use them **as a tool** in your implementation and not the implementation itself. External libraries here is defined as libraries outside of the java standard library.

C. Additional Challenge

1. Drag and drop with saving and loading pipelines

To be frank, Donald isn't very good at this whole programming thing. If you had a drag and drop interface for constructing pipelines, Donald would get much more done quicker. Donald would also like to be able to save these pipelines and load them to continue where he left off.

2. Proof of concept for distributed/parallel computing

Datasets in production these days can have millions of columns and billions of rows. For example, genomics data, high resolution image datasets and large arrays of IoT sensors. Analysis on these datasets usually require multiple computers churning through the dataset in parallel and sharing intermediate computations with each other. Donald would love to be able to use these datasets in his analysis as well. Maybe you could think of how you could convert the methods in the library you have created to use distributed computing.

3. Buzzword models

k-nearest neighbors is definitely a powerful algorithm. However, Donald's friends think he's lame because he doesn't use any "advanced" models with fancy names. He wants to be cool. So cool. Can you help him out?

4. Different scalers

Donald likes the scalers you've made so far. However, he's upset as the scalers seemed to make some of the datasets run worse! Looking into it, you realize that the scalers perform badly on skewed datasets as they amplify outliers. Are there other scalers out there to address this issue?

5. Read and write to database

Reading CSV files is nice. But a lot of data Donald is looking for is in databases waiting to be analyzed. Could you make the library support read and write from some of these databases? (SQL is enough. Feel free to explore non-SQL options too though.)

6. Web Scraping

Donald is getting a little bored of analyzing the few CSV files he has, there are only so many available free interesting CSVs. But not all datasets come in nicely packaged CSV files. Many websites on the magical internet provide data for free but don't have nice APIs to query or charge a fee for API access. Could you help Donald out by creating a web scraper in your library to get him some more interesting CSVs? (Donald will be interested in whatever data you are interested in. As long as your eyes glow when talking about your web scraped dataset, Donald will pay you (with marks).)

7. Take off the training wheels!

Constraints keep the project fun and allow you to focus on the fun parts. But Donald isn't exactly happy with your library limiting the datasets he can explore. Are you ready to face the quotes and commas nightmare? What about dates and timezones? Hashtables? yikes.

8. Anything else you find interesting

There's a lot of other statistics transforms, encoders and visualisations you can do to help Donald on his Data Science journey. Feel free to bring your fresh innovative ideas to Donald too!

D. Tips and comments

1. If there is anything you are having trouble solving, try and think if sorting makes your problem simpler.
2. Code is read more than it is written. Don't be too smart. You may think it is cool to be able to code super fast with your 3 character variables or that you're uber cool because u wrote a one liner with multiple for loops and clauses, but your teammates and you from 1 week in the future will appreciate the extra readability of slowing down to name your methods and variables properly, practice proper spacing and write meaningful comments. Always code as if you are going to explain it one week later, because you are going to have to, even to yourself.
3. You might find GitHub useful for collaborating as it allows your team to sync code instead of sending files to each other on whatsapp. This assignment is pretty modular, so it should not be too hard to moderate the repo and will be a nice learning experience for using GitHub.
4. Methods can have similar names as long as they have different signatures. This may be useful for implementing methods for different types or with different numbers of parameters without confusing yourself by choosing a convoluted method name.
5. Dealing with floating point is something new programmers find very annoying, especially people coming from strong math backgrounds who firmly believe in real number precision. The fact that a binary computation system cannot deal with real numbers is one of the proofs that the computational model cannot solve every problem, as the computation model can only output integers, which is a much smaller set. However, as you become more attuned to computer science, you will start accepting this limitation and avoid fighting it when it's not crucial (for better or for worse). As a result, machine learning algorithms that deal with real numbers usually use float instead of double as the extra precision isn't worth the increased computation and memory. You may use double in your implementations for this assignment if you wish, but know that we won't penalize your precious marks over floating point errors.
6. You might find coding a matrix library with operation broadcasting helpful. Dealing with matrix operations is much easier to conceptualize than writing a pyramid of for loops all the time. This is one of the reasons why linear algebra has become so important in computer science; for loops are just hard for human intuition to deal with.
7. The constraint of only allowing columns with floats beyond requirement 4 is simply because you have not learnt about maps. In python they are called dictionaries which makes much more sense to the common person. Simply map non float columns to float with some sort of schema and you will be able to use the non float columns in your models.

8. The distributed computing additional feature requires good networking fundamentals and it is not recommended you try it unless you're prepared for that. For those who do, it is pretty simple in abstract. You basically need to have a master node that delegates tasks to its workers. The schema you choose to specify tasks and manage the worker-master relationship is up to your creativity. Do notify the question author if you want to attempt it as he would love for you to go through your idea with him. However, again, don't suffer over it please.
9. You can implement an unsupervised learning algorithm for the buzzword additional feature if you wish. However, be warned that unsupervised learning algorithms are not as straightforward as supervised ones as your goal is not always clear. But if you understand it quite well and think you can explain it, go ahead!
10. If you try a reinforcement learning algorithm, please notify the problem author. He would love to hear you explain it.
11. There's an amazing website called [kaggle.com](https://www.kaggle.com) with a lot of data science competitions. If you found this assignment interesting. Feel free to continue learning about data science using kaggle.

E. Questions

If you are not clear about the project, you can contact ONG JACK MIN (ongjackm@gmail.com)

Subject: WIX1002 (Data Ducts)