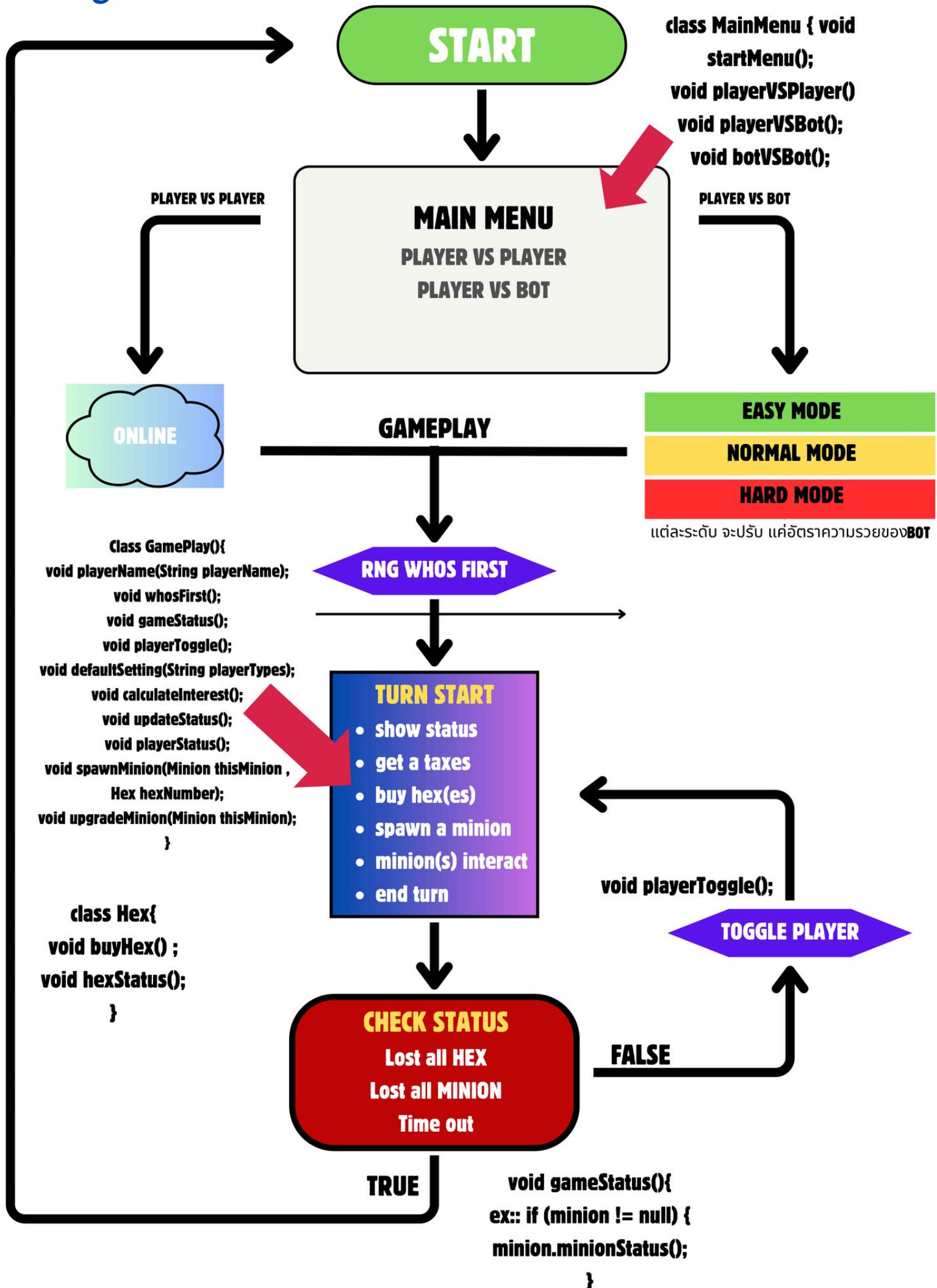


# Design Overview



# MINION

```
Class Minion(){  
public Minion(String name, int hp, int  
attackPower, String skill) {}  
  
void Minion(String Name);  
  
void minionStatus();  
void calculateStatus();  
void updateStatus();  
void minionStatus();  
void upgradeMinion();  
void attackTo(Minion enemy);  
void moveTo(Hex hexNumber);  
}
```

## DEFAULT STATS

NAME  
LEVEL  
HP  
DEF  
ATK  
ATK FORM  
MOVE FORM  
SP SKILLS  
COST

BUFFER

RANGER

TANKER

FIGHTER

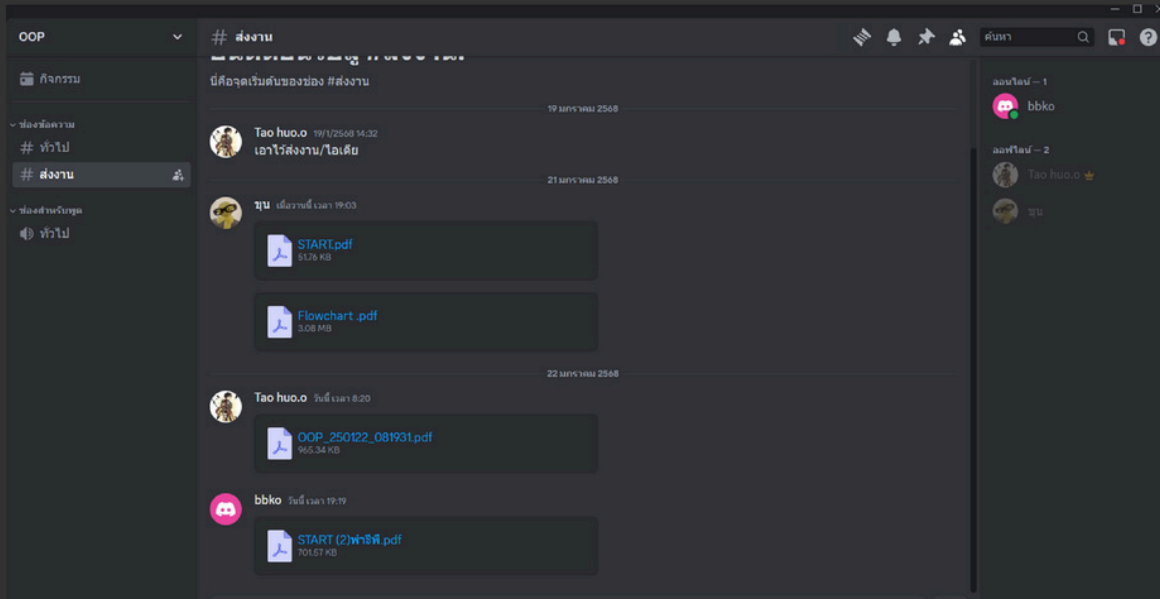
ASSASIN

SPECIFIC NAME  
RANDOM STATS  
FIXED SP SKILLS / COST /  
MOVESET

## Parsing

- รับข้อมูลผู้เล่น**ex::**เลือกโหมดเกมส์ เลือกตัวละคร**5**แบบ การซื้อช่อง
- การกดคำสั่ง โจมตีหรือ**moveminion**
- การประมวลการเลือกกระทำของผู้เล่นในแต่ละเทิร์น

# Toos Github, Java (Spring Framework), React + Vite Discord



## Testing: describe your testing palsn

-ตั้งเป้าหมายในแต่ละฟังก์ชัน และตรวจสอบอินพุตในกรณีที่ถูกต้องและไม่ถูกต้อง

-ทดสอบทุกๆmethod

-สร้างtest case ให้ครอบคลุมโดยอาจจะไล่ไปที่ละmodeในส่วนที่มีในตัวเกม

ex if-esle

function

move and by

โดยมีการทำงานร่วมกันเป็นทีมพร้อมๆกัน

## Work Plan

-หาความถนัดของแต่ละคนในการทำงานหรือหากไม่สามารถทำงานคนเดียวในส่วน

นั้นได้ก็ต้องมีการช่วยกันและศึกษาเพิ่มเติมไปพร้อมกันในการทำงานในทีม

หลักๆพวกเราจะออกแบบให้เห็นภาพเป้าหมายในจุดนั้นๆแล้วค่อยลงมือทำไปใน

แต่ละส่วนทั้ง**Frontend/Backend**

+มีการประชุมและอัปเดตงานที่ต้องจะทำได้ทุกอาทิตย์ละสองครั้งทางช่อง

ทางออนไลน์และออนไซต์

# CODE DESIGN

จะมีการใน **array** ในรูปแบบต่างๆ และ การใช้ **Hash** ในหลายๆ บทบาท หากจำเป็น โดยเน้นไปที่ประสิทธิภาพ และ ตามมาด้วยความเรียบง่ายในการเรียกใช้งาน เพื่อลดความซับซ้อนของระบบ ซึ่งก็จะนำมาซึ่ง ประสิทธิภาพที่จะต้องต่ำลง แต่ก็จะทำให้การเขียนโค้ดนั้นไหลลื่น และมีบัคที่น้อยที่สุดเท่าที่เป็นได้

## Known Problems

- 1 ) Incomplete design** - การออกแบบ **Class** และ **Methods** ทั้งหมด ยังไม่สมบูรณ์ ซึ่ง จะมีการเพิ่มเติมในอนาคตตามความจำเป็น แต่ ภาพรวมแล้ว การ ดีไซน์ค่อนข้างจะมีความสมบูรณ์แล้ว
- 2 ) Online & BOT** - ด้วยความที่ทุกคนในทีมไม่เคยทำเรื่องที่เกี่ยวข้องกับระบบนี้มาก่อน จึงอาจจะเกิดปัญหาในด้านนี้ได้
- 3 ) Graphic design** - ในทีมยังต้องหาวิธีแก้ปัญหาเรื่องการวาดตัวละคร หรือ การ ออกแบบเกมให้น่าสนใจต่อผู้เล่น

# COMMENTS

## 1 ) How much time did you spend on the design?

= 4 วัน เวลาส่วนใหญ่ จะถูกใช้ไปกับการเข้าใจรูปแบบการเล่นของเกมที่มาจาก **Project** ตั้งต้น

## 2 ) What advice should we have given you before you started?

= การสร้าง **BOT** / การเขียน **UI** ให้สามารถใช้ร่วมกับภาษา **Java** ได้

## 3 ) What was surprising about the design?

= การได้สร้างเกี่ยวที่สามารถเล่นกับ **BOT** หรือ **Online** ได้

## 4 ) What was hard about the design?

= การได้ออกแบบสร้างเกมในแบบที่ทางรายวิชาต้องการ

## 5 ) What did you like or dislike about the project?

= ไม่ชอบที่ การอธิบายเกม **KOMBAT** ค่อนข้างเข้าใจยาก ควรมีภาพหรือวิดีโอ ประกอบความเข้าใจ เรื่องระบบที่ต้องการ

## 6 ) What would you change about the project?

= ระบบที่เกี่ยวข้องกับ **Interest rate** โดย จะขึ้นอยู่กับ ปริมาณ **Hex** ที่ถือครอง / ระบบที่ **Minions** สามารถอัพเลเวลได้ จากการใช้เงิน

## Project timeline (subject to change)

- Wed 15 Jan:
  - = Project team agreement due
    - This is a formal document outlining the expectations, roles, and responsibilities of your team, covering all aspects of the team's work. It should specify the team's goals, how team members will communicate with each other, and what happens if someone violates the agreement.
    - This includes guidelines or principles that team members agree upon, in order to facilitate collaboration, communication, and productivity within your team.
- Wed 22 Jan:
  - = Design overview document for game state and minion-strategy evaluator due
    - This includes the outline of class hierarchy for your game state, parser, and evaluator.
    - Your design will be given feedback, so you can address flaws before you submit your code.
- Wed 5 Feb:
  - = Implementation of game state and minion-strategy evaluator due
    - At this point, your project should be able to parse and execute minion strategies, given a mock game state required for the execution.
  - = Design overview document for user interface due
    - This includes the outline of the display (view and controller).
    - Your design will be given feedback, so you can address flaws before you submit your code.
- Wed 19 Feb:
  - = Implementation of user interface due
    - At this point, your evaluator should be integrated with the game state.
    - At this point, your user interface should be displayable and can mock some strategy commands
  - = Design overview document for server-client implementation due
    - This includes the outline of the interaction between game state and user interface.
    - Your design will be given feedback, so you can address flaws before you submit your code.
- Wed 5 Mar:
  - = Server-client implementation due
    - At this point, your game state should be able to interact with your user interface, and vice versa.
  - = Project report due
    - Architectural overview of your system
    - Report on what was done according to plan and outside the plan
    - Retrospect on what was done well, and what could have been better
  - = Possible extension: after the final exam period (no later than Tue 25 Mar)