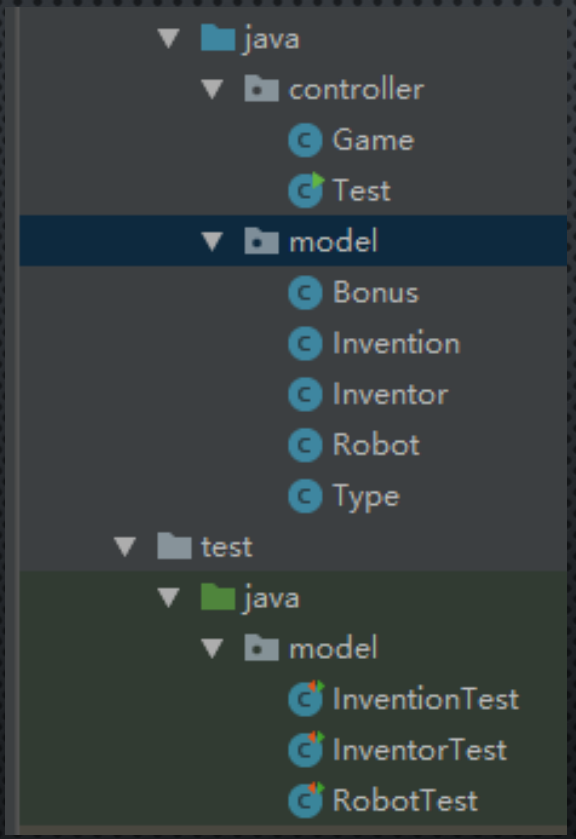
L'INVENTEUR

MUNDUS 3 GROUPE B

YU TENG, WANG HANTING, ZHANG YUXIN

LES POINTS FORTS:



La structure : MVC (Model View Controller) qui est facile pour gérer. Donc les codes se sont bien séparés.

Le robot AI : On a réalisé un robot AI. Il peut choisir l'invention disponible avec le plus point.

LES POINTS FAIBLE:

L’intellegence d’AI est en faible niveau. Donc il y a beaucoup de place pour l’améliorer.

Les Bonus. Maintenant il y a deux variable: point et abilité. On peut l’enrichit pour le rendre plus interesant.

DES INDICATIONS:

```
public class Game {
    Robot robot1;
    Invention[] Stage1inventions;
    Invention[] Stage2inventions;
    Invention[] Stage3inventions;
    Inventor[] inventors1;
    Robot robot2;
    Inventor[] inventors2;

    //Game Control
    void init() {...}
    void chooseOneTime(int stage) {...}
    Invention[] chooseStageInventions(int stage) {...}
    int getNextStage(int stage) {...}

    //init data
    void initInventors() {...}
    void initInventions(int stage) {...}

    //show result
    int compareWhoIsWinner() {...}
    void showStage(int stage) {...}
    void showInventions(Invention[] inventions) {...}
    void showResult() {...}
}
```

`void chooseOneTime(int stage){}`
Ce méthode reçoit le numéro de stage, chaque robot choisit une fois d'invention avec ses inventeurs.

`int getNextStage(int stage){}`
Pour changer le stage.

DES INDICATIONS:

```
public class Robot {
    String name;
    int point;
    Inventor[] inventors;
    public Robot(String name, Inventor[] inventors) {...}
    //Choose Method
    public Inventor chooseInventor() {...}
    public boolean chooseInvention(Inventor inventor,Invention[] inventions) {...}
    public boolean chooseInventionWithAI(Inventor inventor,Invention[] inventions) {...}
    //bonus
    private Bonus selectBonus(Invention invention) {...}
    private void addBonus(Inventor inventor, Bonus bonus) {...}

    public void reDisponibleInventors() {...}
    //all inventor finish work
    public boolean checkAllIndisponible() {...}
    public void show() {...}

    public void addPoint(int point) { this.point = this.point + point; }
    public int getPoint() { return point; }
}
```

```
public Inventor chooseInventor(){…}
```

```
public boolean chooseInvention
(Inventor inventor,Invention[] inventions){}
```

```
public boolean chooseInventionWithAI(Inventor
inventor,Invention[] inventions){}
```

```
private Bonus selectBonus(Invention invention){}
```

DES INDICATIONS:

```java
public class Invention {
    String name;
    boolean state;
    int point;
    int[] require;
    Bonus[] bonuses;
    public Invention(String name, int[] require, int point, Bonus[] bonuses) {...}

    void requireMinAbility(Inventor inventor) {...}
    public boolean isFinish(){...}
    public void show() {...}
```

```java
public class Inventor {
    String name;
    boolean state;
    int[] ability;
    public Inventor(String name) {...}
    public Inventor(String name,int[] ability) {...}
    public void addAbility(int value) {...}
    public void show() {...}

    public void setDisponible() { state = true; }
    public void setIndisponsable() { state = false; }
    public boolean getDisponible() { return state; }
}
```