Compte rendu de la semaine 21

Cette semaine, j’ai effectué le codage de la classe Player. C’est la classe qui traite du joueur. Elle contient son nom, son equipe et son nombre de points. Elle contient des getter et setter pour les arguments de la classe. Elle possède une méthode pour ajouter des points une méthode pour verifier la condition de victoire (12 points). Elle permet d’initialiser les pions du joueur sur la grille.
La semaine prochaine je pense faire la classe Game et aider mon binome dans son codage d’autres classes. Je n’ai pas fini de coder la méthode initTabPion() car je souhaite en discuter avec Florian d’abord pour comparer nos idées sur celle-ci.

Veuillez trouver ci-dessous les classes faites cette semaine.

CLASSE PLAYER :

```
package arcanor;

/**
 * Public class Player
 */
public abstract class Player{
  protected String name;
  protected PionColor team;
  protected int point;

  /**
   * Contructor of the Player class. Initializes all the attributes of the class
   * @param name (String) player's name
   * @param team (PionColor) player's team color
   * @param point (int) player's amount of point
   */
  public Player(String name, PionColor team, int point){

  }
  /**
   * Used to have the name of the player
   * @return (String) player's name
   */
  public String getNom(){
    return this.nom;
  }

  /**
   * Used to set the name of the player
   * @param nom (String) new player's name
   */
  public void setNom(String nom){
    if (nom != null) {
      this.nom = nom;
    } else{
      System.out.println("Player - setNom - Parameter Error ");
    }
  }

  /**
   * Used to have the color of the player's team color
   * @return (PionColor) player's team color
   */
  public PionColor getTeam(){
    return this.team;
  }

  /**
```

```
 * Used to set the color of the player's team
 * @param team (PionColor)
 */
public void setTeam(String team){
  if (team !=null) {
    this.team = team;
  } else{
    System.out.println("Player - setTeam - Parameter Error");
  }
}

/**
 * Check the amount of point to declare a winner
 * @return (boolean) true if winner false otherwise
 */
public boolean checkWin(){
  boolean check = false;
  if (this.point >= 12) {
    check = true;
  }
}

/**
 * Add amount of point to the player
 * @param amount (int) the amount of point to add at the attribute point of the player
 */
public void addPoint(int amount){
  if (amount >= 1) {
    this.point = this.point + amount;
  } else {
    System.out.println("Player - addPoint - Parameter error");
  }
}

/**
 * Give the amount of point of the player.
 * @return (int) amount of point
 */
public int getPoint(){
  return this.point;
}

/**
 * Initializes the pawn of the player on the board
 */
private void initTabPion(){

}

/**
 * Allows the player to play and move his pawn, add point or declare a winner
 */
```

```
  public abstract void play();
}
```

ENUMERATION PionColor

```
package arcanor;

/**
 * Possible color of the players
 */
public enum PionColor {
  WHITE,
  BLACK,
  NONE;
}
```