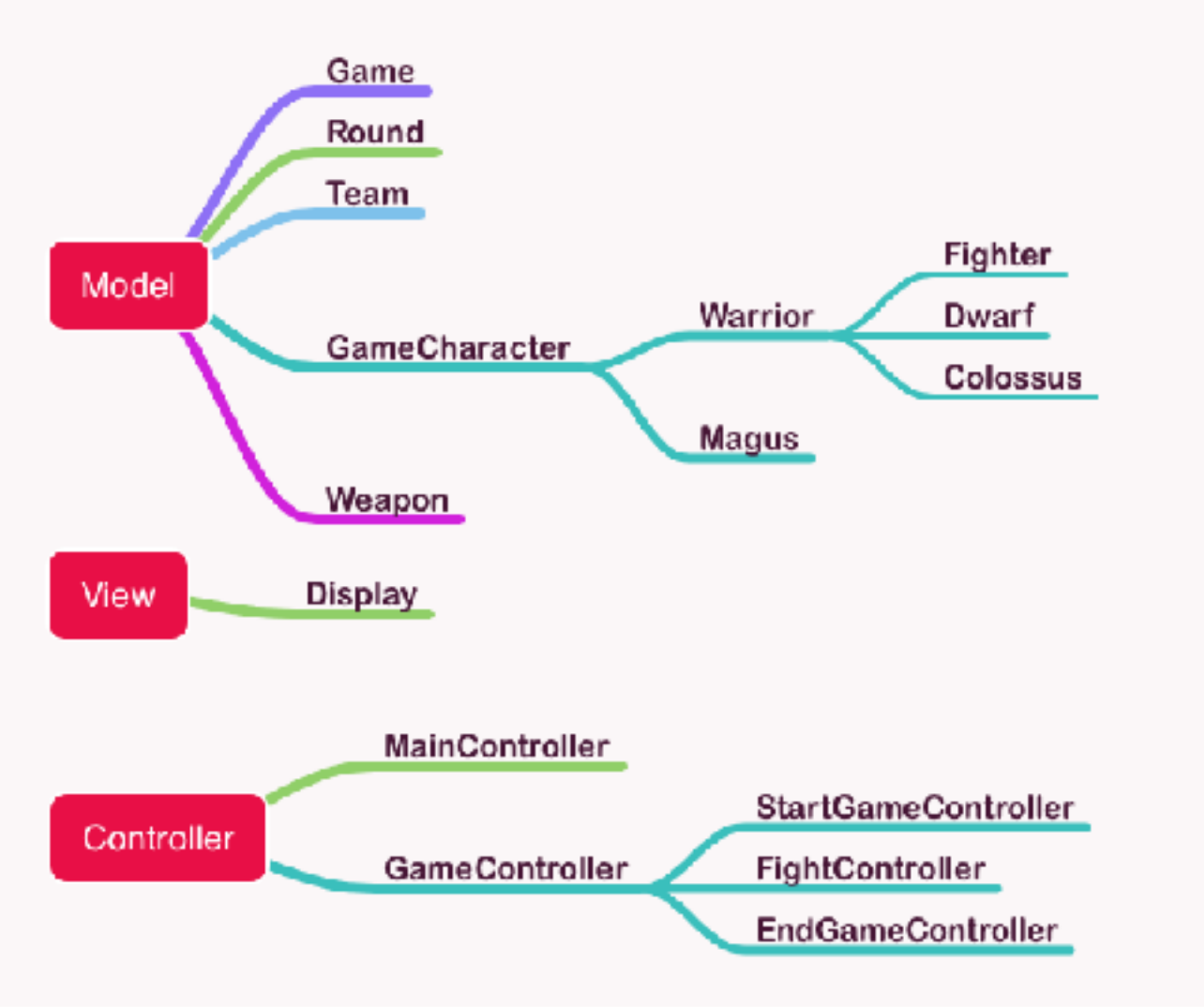
Le choc des brutes

Les bonus

Bonus 1 : MVC

Pourquoi ?
Après avoir développé les classes qui correspondent au modèle, je n'avais pas envie de mélanger des fonctions d'affichage et de données dans la même classe. Grâce aux conseils d'Ambroise, j'ai choisi d'utiliser le modèle MVC.

Comment ?
J'ai commencé par créer la classe Display pour répondre à mes besoins d'affichage formaté et de contrôle de saisie.
Puis j'ai développé les classes du contrôleur. Une pour chaque étape du jeu (Création des équipes, les combats, le résumé de fin de combat).



Bonus 2 : Auto-select

Pourquoi ?
Pendant la phase de test, je me suis rendu compte que lorsqu'il ne restait qu'un seul personnage en vie dans une équipe, il serait plus intuitif que ce personnage soit automatiquement sélectionné pour jouer le tour.
De même, lorsqu'il ne reste qu'un seul personnage cible, il pourrait lui aussi être automatiquement sélectionné.

Comment ?
J'ai ajouté à la fonction du choix du personnage actif une condition « **si** l'équipe du joueur qui joue le tour n'a qu'un seul personnage en vie **alors** c'est ce personnage qui va jouer **sinon** demande au joueur quel personnage jouer. »
De même pour la fonction du choix de la cible.

```
private func askActiveCharacter(activeTeam: Team) -> GameCharacter {
    if activeTeam.countCharacterAlive() == 1 {
        let activeCharacter: GameCharacter = activeTeam.getCharacter(name: activeTeam.getCharactersAliveNames()[0])
        display.gmSpeak(text: "Tu n'as que \(activeCharacter.name) en vie.", mood: Display.gmMood.normal)
        display.littleBreak()
        return activeCharacter
    } else {
        display.gmSpeak(text: "\(activeTeam.player), choisis avec quel personnage tu vas jouer ce tour :", mood: Display.gmMood.normal)
        return activeTeam.getCharacter(name: display.readStringBetweenWords(in: activeTeam.getCharactersAliveNames()))
    }
}
```

```
private func askTargetCharacter(activeTeam: Team, activeCharacter: GameCharacter) -> GameCharacter {
    if activeCharacter is Pogue {
        if activeTeam.countCharacterAlive() == 1 {
            display.gmSpeak(text: "\(activeCharacter.name) est bien seul. Il soigne ses blessures.", mood: Display.gmMood.normal)
            display.littleBreak()
            return activeCharacter
        } else {
            display.gmSpeak(text: "Choisis la cible de \(activeCharacter.name) :", mood: Display.gmMood.normal)
            return activeTeam.getCharacter(name: display.readStringBetweenWords(in: activeTeam.getCharactersAliveNames()))
        }
    } else {
        let targetTeam: Team = game.getInactiveTeam()
        if targetTeam.countCharacterAlive() == 1 {
            let targetCharacter: GameCharacter = targetTeam.getCharacter(name: targetTeam.getCharactersAliveNames()[0])
            display.gmSpeak(text: "Il ne reste que \(targetCharacter.name) à abattre.", mood: Display.gmMood.normal)
            display.littleBreak()
            return targetCharacter
        } else {
            display.gmSpeak(text: "Choisis la cible de \(activeCharacter.name) :", mood: Display.gmMood.normal)
            return targetTeam.getCharacter(name: display.readStringBetweenWords(in: targetTeam.getCharactersAliveNames()))
        }
    }
}
```