

---

## Modélisation et Conception Objet

### TD 4 : Le Yams

---

Vous allez créer, en utilisant tout ce que vous avez vu jusque maintenant, un application permettant de jouer au *Yams* (les règles sont disponibles en section 3).

## 1 Tâches

Préparez un petit dossier de vos actions pour s'en rappeler les sessions suivantes et conserver un déroulé de votre projet.

### 1.1 Trouver les tâches

On commence par définir des tâches à réaliser

1. Lire les règles.
2. Réfléchir à une UI permettant de jouer au Yam's. Faites des scénario de tours de jeu possibles (début de partie, milieu de partie et fin de partie) et regardez comment utiliser votre interface.
3. Faire une première liste de choses à faire pour réaliser votre jeu
4. Extraire de la liste les éléments minimaux pour avoir une version pré-alpha jouable.

La liste minimale pour avoir un jeu est ce qu'il faut que vous ayez développé à la fin des 2 TPs. Il y aura forcément d'autres éléments à ajouter (des oublis ou encore des tâches trop grosses qu'il a fallu découper en sous-tâches) mais cette liste est votre guide dans la préparation de votre projet.

### 1.2 Complexité des tâches

Vous allez maintenant réaliser les différentes tâches. Pour permettre de travailler à plusieurs et favoriser les ajouts de fonctionnalités, tout code doit être testé :

- les classes et les différentes fonctions sont testés comme d'habitude,
- pour l'UI, on écrira des petits scénarios d'utilisation. Par exemple : *en cliquant sur le bouton "roll", les 5 dès doivent être lancés.*

### 1.3 Ordre de réalisation

L'ordre dans le quel les tâches sont codées va dépendre des dépendances entre tâches et de la complexité de réalisation. Pour se mettre d'accord sur la complexité d'une tâche à réaliser on pourra utiliser [https://fr.wikipedia.org/wiki/Planning\\_poker](https://fr.wikipedia.org/wiki/Planning_poker).

Ensuite, chaque développeur choisi la tâche qu'il souhaite réaliser.

## 2 En TD

Pendant le TD, vous aurez à préparer les différentes tâches et à déterminer une première liste. Donc vous aurez à :

- trouver une UI permettant de jouer,
- déterminer les classes et leurs relations au format UML (pour l'instant gros grain, les relation fines et différentes méthodes seront ajoutées au fur et à mesure),
- trouver une première complexité des différentes tâches.

Entre chaque partie, on fera un point entre les différents groupes. Il n'est pas nécessaire d'avoir la même solution, ni les mêmes complexités pour une même tâche : chaque équipe à sa propre perception des tâches à accomplir.

### 2.1 Les 2 prochains TPs

On effectuera toujours ce processus :

1. Chaque membre du projet choisi une tâche à réaliser.
  - Si aucune tâche ne semble réalisable, on coupe une tâche trop complexe en plusieurs sous-tâches ou on ajoute des tâches permettant de réaliser plus facilement une autre tâche
  - On évalue la complexité des nouvelles tâches à réaliser.
2. une fois la tâche et les tests réalisés on supprime la tâche de la liste à faire et on l'ajoute dans la liste des tâches faites.

Une façon sympathique (et utilisée de façon pro) de gérer les différentes tâches est d'utiliser un tableau, ou encore <https://trello.com> (un petit tuto est disponible là : <https://www.youtube.com/watch?v=DqmXYPNrHcw>)

### 3 Les règles

On utilisera ici les règles de : [https://fr.wikibooks.org/wiki/Boîte\\_à\\_jeux/Yams](https://fr.wikibooks.org/wiki/Boîte_à_jeux/Yams). Le but est de créer un jeu à 1 joueur.

**Matériel :** 5 dés à six faces

**Règles:** Les joueurs jettent les dés et doivent former des combinaisons. Ils ont le droit de lancer les dés 3 fois de suite, tout en sélectionnant les dés qu'ils souhaitent garder. Ils tentent une combinaison par session de jeu.

A la fin de la phase de lancer des dés, ils choisissent dans quelle combinaison ils inscrivent leurs résultat, le nombre de points obtenus varie avec le choix de la combinaison. S'il ne peuvent jouer la combinaison voulue, ils doivent se reporter sur un autre choix ou bien, si aucune autre combinaison n'est possible, inscrire 0 dans la case de la combinaison voulue.

Le tableau des scores est divisés en 2 sections (partie mineure et majeure). Ils peuvent inscrire un bonus en fonction du nombre de points obtenus dans la partie mineure.

#### Combinaisons :

- Partie mineure :
  - Maximum de 1: les joueurs doivent obtenir le plus grand nombre de dés avec la face 1 / ils marquent 1 fois le nombre de dés de valeur 1
  - Maximum de 2: les joueurs doivent obtenir le plus grand nombre de dés avec la face 2 / ils marquent 2 fois le nombre de dés de valeur 2
  - Maximum de 3: les joueurs doivent obtenir le plus grand nombre de dés avec la face 3 / ils marquent 3 fois le nombre de dés de valeur 3
  - Maximum de 4: les joueurs doivent obtenir le plus grand nombre de dés avec la face 4 / ils marquent 4 fois le nombre de dés de valeur 4
  - Maximum de 5: les joueurs doivent obtenir le plus grand nombre de dés avec la face 5 / ils marquent 5 fois le nombre de dés de valeur 5
  - Maximum de 6: les joueurs doivent obtenir le plus grand nombre de dés avec la face 6 / ils marquent 6 fois le nombre de dés de valeur 6
  - Ils obtiennent un bonus de 35 points si le total de la partie mineure est supérieure à 63 points.
- Partie majeure :
  - Breelan: les joueurs doivent obtenir 3 dés de même valeur / ils marquent 3 fois la valeur des dés identiques
  - Carré: les joueurs doivent obtenir 4 dés de même valeur / ils marquent 4 fois la valeur des dés identiques

- Full: les joueurs doivent obtenir 3 dés de même valeur et 2 dés de même valeur - (breelan + paire) / ils marquent 25 points
- Petite Suite: les joueurs doivent obtenir 4 dés qui se suivent (1-2-3-4 / 2-3-4-5 / 3-4-5-6) / ils marquent 30 points
- Grande Suite: les joueurs doivent obtenir 5 dés qui se suivent (1-2-3-4-5 / 2-3-4-5-6) / ils marquent 40 points
- Yams: les joueurs doivent obtenir 5 dés de même valeur / ils marquent 50 points
- Chance: les joueurs doivent obtenir le plus grand nombre de points / ils marquent la somme de la valeur des dés

En fonction du total (bonus inclus) de la partie mineure et de la partie majeure, le vainqueur est désigné.