

# Tu te mets combien?

Giorgio Caculli LA196672, Guillaume Lambert LA198116, Tanguy Taminiau LA199566  
Groupe B05

21 février 2021

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Analyse de l'existant</b>	<b>4</b>
<b>2 Description général de l'application</b>	<b>4</b>
2.1 Fichier . . . . .	4
2.2 Spécification technique . . . . .	4
2.3 L'interface graphique utilisateur (GUI) . . . . .	4
<b>3 Product backlog</b>	<b>5</b>
<b>4 Diagramme de classe</b>	<b>5</b>
<b>5 Plan de sortie</b>	<b>5</b>

## Introduction

Dans le cadre du cours de “Projets” de l’UE 210, nous allons devoir créer un jeu similaire à “Tu te mets combien?” (TTMC). Il s’agit d’un jeu comportant des cartes comportant des questions. Il existe quatre thèmes repérable par leurs couleurs respectives, la couleur :

- mauve est attribué au carte ayant pour thème “improbable”
- orange pour “plaisir”
- bleu pour “informatique”
- vert pour “scolaire”

Chaque cartes possède un thème, un sujet en rapport avec le thème, et quatre questions.

Les question sont numéroté de un à quatre et trié par ordre croissant de difficulté.

Le jeu commence avec un joueur qui tire une carte, il pose la question "tu te mets combien en (sujet de la carte)?" à l’autre joueur, l’autre joueur lui répond un chiffre entre un et quatre, le joueur ayant tiré la carte pose alors la question correspondante au nombre donné par l’autre joueur. En cas de bonne réponse, le joueur ayant répondu gagne un nombre de points équivalent au numéro de la question. En cas de mauvaise réponse, le joueur ne gagne aucun point. C’est ensuite au joueurs ayant répondu de tirer une carte et d’interroger l’autre.

# 1 Analyse de l'existant

Différents jeux existent sur le style de “Tu te mets combien?”. Plusieurs nous sont parvenu à la tête :

- Trivial Pursuit
- Qui veut gagner des millions ?
- Questions pour un champion

Étant donné que c'est un jeu de table, il ne possède pas de version digitale.

Comme pour les exemples cités précédemment, ce jeu implique deux ou plusieurs joueurs. Les participants devront piocher une carte, carte qui sera caractérisé par une couleur, couleur qui caractérisa aussi le thème de la question :

- le mauve pour “improbable”
- l'orange pour “plaisir”
- le bleu pour “informatique”
- le vert pour “scolaire”

Lorsque la carte sera pioché, si la réponse est correcte, les points correspondants à la difficulté de la carte lui seront attribué, si la réponse est erronée, alors le jouer ne gagnera aucun point.

## 2 Description général de l'application

### 2.1 Fichier

Le jeu utilisera comme fichier de sauvegarde pour les cartes ainsi que pour le deck le format JSON. Format très populaire et qui facilité grandement l'échange, le chargement et l'écriture des données. Le jeu intègre donc une bibliothèque de création et lecture du format JSON.

### 2.2 Spécification technique

Le programme est développé entièrement dans le langage de programmation Java. Plus précisément, la version 11 du JDK (Java Development Kit).

Les librairies utilisées seront :

- JavaFX 11
- GSON 2
- JUnit 5

La plupart des fonctionnalités seront basés sur la plupart des librairies natives que Java propose, fonctionnalités comme :

- Logger
- Net
- Thread

### 2.3 L'interface graphique utilisateur (GUI)

L'utilisateur aura comme principale interface graphique une fenêtre avec laquelle il pourra accéder au différentes parties du programme. Tout affichage sur la fenêtre dépendra des interactions que l'utilisateurs aura avec les différents menus qui lui seront proposés. De manière générale, l'interface démarrera sur un menu principale qui mènera l'utilisateur vers les différentes partie du logiciel :

- Le menu pour démarrer la partie
- Le menu pour paramétrer le programme
- Le menu pour manipuler les composants du jeu

Un variété de Panes lui seront proposé, et chaque Pane aura sa propre utilité et importance.

### 3 Product backlog

1	Développement du modèle, classe : Question, Deck, BasicCard.
2	Mise en place des tests unitaires pour vérifier l'efficacité des méthodes définies dans les différentes classes du modèle.
3	Initialisation des différentes exceptions qui peuvent être rencontrées lors de l'utilisation du programme.
4	Création d'un menu client afin de vérifier les différents inputs faits par l'utilisateur.
5	Ajout de règles dans les différentes classes du modèle au cas où l'utilisateur arriverait à faire une mauvaise manipulation.
6	Initialisation des différents "Panels" que l'utilisateur va interagir avec.
7	Mise en place des différents "Panels" dans une fenêtre.
8	Exécution des tests entre les interactions faites par l'utilisateur avec l'interface graphique et les classes modèle.
9	Ajout de règles supplémentaires en cas d'exécution d'une mauvaise manipulation.
10	Ajout d'effets graphiques ou sonores dans l'application afin de rendre l'utilisation plus agréable et intuitive.
11	Finalisation du produit

### 4 Diagramme de classe

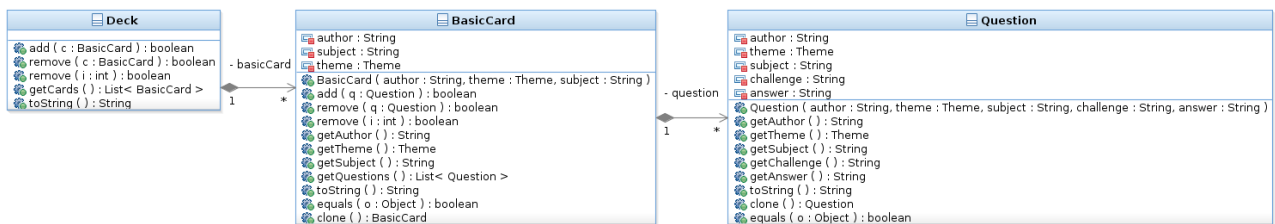


FIGURE 1 – Diagramme de classe du modèle

### 5 Plan de sortie

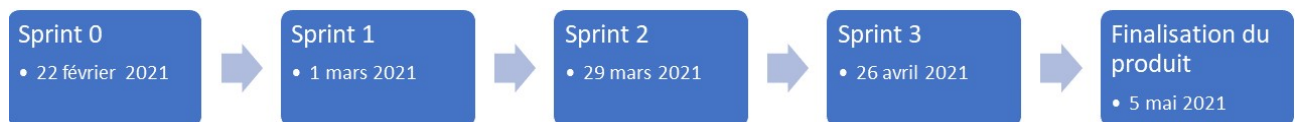


FIGURE 2 – Release plan

- Sprint 0 - Analyse et recherche.  
Cette première étape nous permettra de mettre en place le modèle du projet sur base de ce qui a été trouvé sur internet. Une fois compris les règles, il sera possible d'établir les fonctionnalités de base pour les composants principaux du programme ainsi que la mise en place des règles qui caractérisent chaque composant.
- Sprint 1 - Première démonstration en présence du Product Owner.  
Lors de la première démonstration, les différentes fonctionnalités seront testées. Les tests unitaires ne renverront aucun erreur et les différentes exceptions qui caractérisent chaque composant seront gérées.
- Sprint 2 - Deuxième démonstration du produit.  
Lors de la deuxième démonstration, un programme avec interface interactive sera mis en place. Ce prototype permettra d'interagir avec les différentes composantes et les manipuler librement.
- Sprint 3 - Troisième et dernière démonstration.  
Lors de la troisième démonstration, le programme aura une interface graphique basée sur JavaFX et toutes les interactions seront finalisées.
- Finalisation du produit - Présentation du produit final au client.  
Tous les bugs seront corrigés et toutes les interactions prévues seront fonctionnelles.

## Glossaire

**JDK** Java Development Kit. 4

**JSON** JavaScript Oriented Notion. 4

**TTMC** Tu te mets combien?. 3, 4