

Master 1 GIL - Gestion de projet
Document d'Architecture Logiciel du "6 qui prend"
Agora V3-1

Groupe 1 Agora V3-1 : Partie bibliothèque de jeux

14 février 2024

Version	3
Date	14 février 2024
Rédigé par	BINGINOT Etienne CAUCHOIS Niels DUCROQ Yohann KHABOURI Izana MAZUY Axelle MONTAGNE Erwann THIBERVILLE Malvina VAN LIEDEKERKE Florian

Mises à jour du document

Version	Date	Modification réalisée
1	29 novembre 2023	Création du document
2	17 décembre 2023	Automatisation de la pose des cartes
3	22 janvier 2024	Mise à jour après les retours de l'enseignant de gestion de projet
4	14 février 2024	Mise à jour de la BDD après le sprint 1

Table des matières

1	Objet du document	4
2	Documents de références	5
3	Terminologie	5
4	Architecture statique	6
4.1	Les entités	6
4.1.1	Structure	6
4.1.2	Description des composants	6
4.2	Les services	7
4.2.1	Structure	7
4.2.2	Description des composants	8
4.3	Les contrôleurs	10
4.3.1	Structure	10
4.3.2	Description des composants	10
5	Architecture dynamique	11
6	Maquette de vue	14

1 Objet du document

Ce document a pour but de décrire l'architecture logicielle d'une version digitale du jeu 6 qui prend. Nous allons utiliser la bibliothèque que l'on va créer pour l'implémentation de jeux.

Ce jeu constitue une base à analyser pour alimenter la bibliothèque de manière générale et regrouper avec les autres analyses de jeux les points communs.

Ce jeu sera rendu fonctionnel aux joueurs inscrits et implémenté avec l'utilisation du framework Symfony.

2 Documents de références

Nos références utilisées sont les DAL des précédentes années concernant le projet AGORA, la feuille de présentation du projet, les STB générale et du 6 qui prend réalisées ainsi que les compte-rendus des réunions avec le client et le référent technique.

3 Terminologie

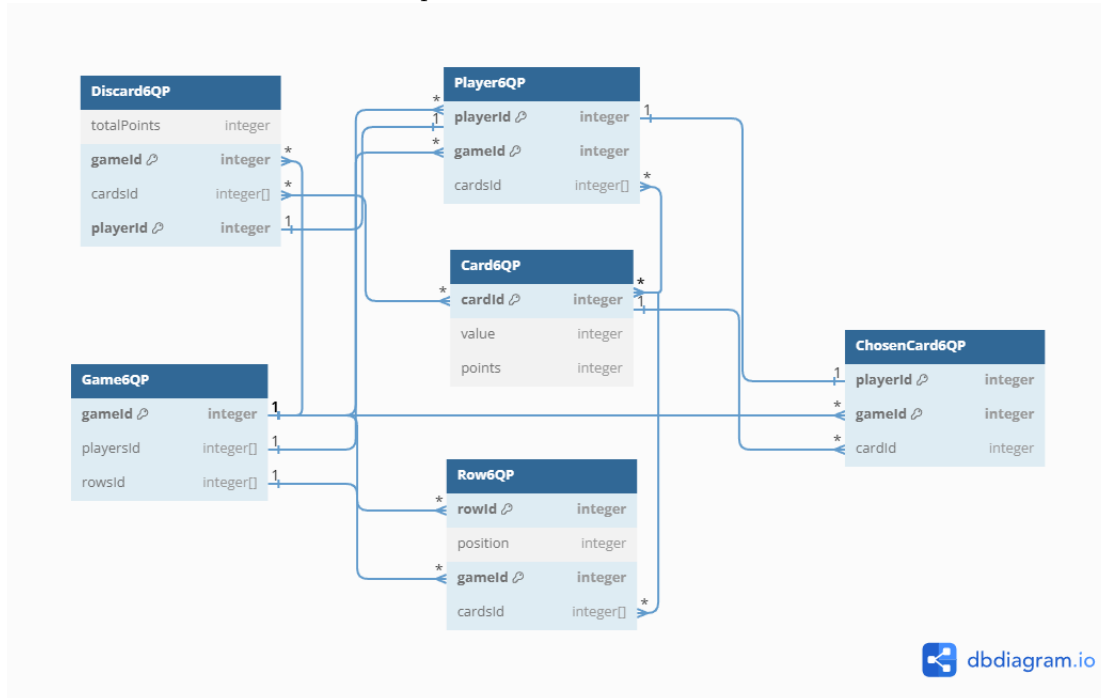
Tête de boeuf : dans le jeu "6 qui prend", un boeuf correspond à 1 point. Les points seront additionnés à la fin du jeu, le but est d'en avoir le moins possible. Une carte peut avoir de 1 à 7 têtes de boeufs.

4 Architecture statique

4.1 Les entités

4.1.1 Structure

L'architecture statique se présente par les entités du jeu que l'on retrouve en base de données, permettant ainsi de stocker les informations d'une partie en cours.



4.1.2 Description des composants

Nom de la classe : Game6QP

Description : Cette classe permet d'identifier une partie du jeu "6 qui prend" en cours

Attributs	Méthodes
gameId : un entier identifiant une partie en cours	
playersId : un tableau d'identifiants de joueurs permettant de connaître les participants	
rowsId : un tableau d'identifiants de lignes constituant le jeu (le plateau principal comporte 4 lignes).	

Nom de la classe : Player6QP

Description : Cette classe permet d'identifier un joueur du jeu "6 qui prend" en cours de partie

Attributs	Méthodes
playerId : un entier identifiant un joueur d'une partie	
gameId : un entier identifiant la partie auquel le joueur est associé	
cardsId : un tableau d'entiers d'identifiants de cartes appartenant au joueur	

Nom de la classe : Row6QP

Description : Cette classe permet d'enregistrer l'état d'une des lignes du jeu 6 qui prend pour une partie en cours

Attributs	Méthodes
rowId : un entier identifiant une ligne pour une partie	
position : un entier donnant la position de la ligne sur le plateau	
gameId : un entier identifiant la partie en cours à laquelle est associée la ligne	
cardsId : un tableau d'entiers ordonné identifiant les cartes positionnées sur la ligne	

Nom de la classe : Card6QP

Description : Cette classe permet d'enregistrer ce qui correspond à une carte de 6 qui prend

Attributs	Méthodes
cardId : un entier identifiant une carte	
value : un entier correspondant à la valeur indiquée sur la carte (de 1 à 104)	
points : un entier correspondant aux points rapportés par la carte (les têtes de boeuf)	

Nom de la classe : ChosenCard

Description : Cette classe permet d'identifier les cartes choisies par les joueurs en début de tour.

Attributs	Méthodes
playerId : un entier identifiant le joueur ayant choisi la carte	
gameId : un entier identifiant la partie auquel la carte choisie est associée	
cardId : un entier identifiant la carte du jeu choisie	

Nom de la classe : Discard6QP

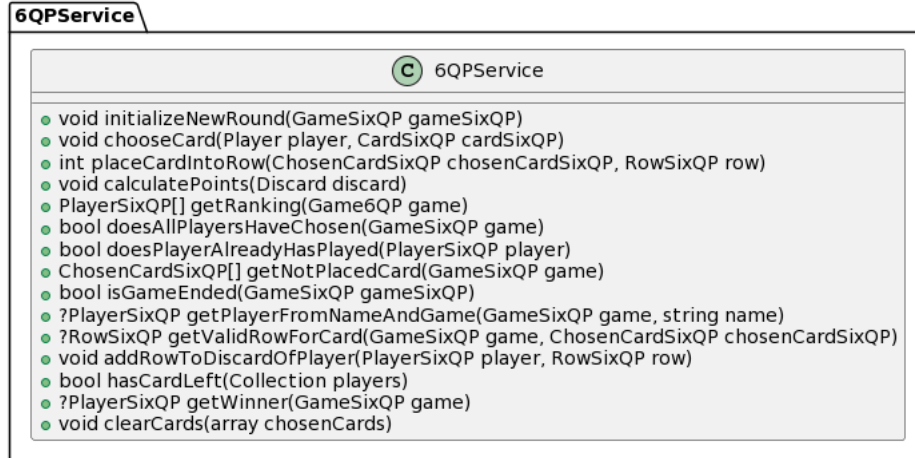
Description : Cette classe permet d'identifier les cartes récupérées par les joueurs.

Attributs	Méthodes
playerId : un entier identifiant le joueur ayant récupéré les cartes	
gameId : un entier identifiant la partie auquel les cartes sont associées	
cardsId : un tableau d'entiers contenant les identifiants des cartes du jeu récupérées	
totalPoints : un entier contenant la somme des points du joueur	

4.2 Les services

4.2.1 Structure

L'architecture statique se présente par les services du jeu, permettant ainsi de manipuler les informations d'une partie en cours.



4.2.2 Description des composants

Nom de la classe : 6QPService


Description : Cette classe permet de manipuler les entités afin d'en changer le comportement.

Méthode	Type de retour	Description	Référence
initializeNewRound (GameSixQP game)	void	initialise une nouvelle manche d'une partie (game)	STB-GES-6QP-04
chooseCard (PlayerSixQP player, CardSixQP card)	void	à partir de la carte (card), passe la carte du joueur (player) en état choisi.	STB-GES-6QP-01
placeCardIntoRow (ChosenCardSixQP card, RowSixQP row)	int	place une carte card sur la ligne désignée par row	STB-GES-6QP-02
calculatePoints (DiscardSixQP discard)	void	calcule les points cumulés par un joueur dans son tas discard et stocke cette valeur dans totalPoints.	STB-GES-6QP-08
getRanking (GameSixQP game)	array	renvoie un tableau d'entier ordonnés du joueur ayant le moins de points à celui qui en a le plus.	STB-GES-6QP-10
doesAllPlayersHaveChosen (GameSixQP game)	bool	indique si tous les joueurs ont choisi leur carte	
doesAllPlayersAlreadyHavePlayed (GameSixQP game)	bool	indique si tous les joueurs ont joué	
getNotPlacedCard (GameSixQP game)	array	retourne un tableau de cartes choisies encore non placées	
isGameEnded (GameSixQP game)	bool	indique si la partie est finie	
getPlayerFromNameAndGame (GameSixQP game, string name)	?PlayerSixQP	cherche le joueur à partir d'un nom et d'un jeu	
getValidRowForCard GameSixQP game, ChosenCardSixQP chosenCardSixQP)	?RowSixQP	calcule la ligne où poser la carte	
addRowToDiscardOffPlayer (PlayerSixQP player, RowixQP row)	void	ajoute toutes les cartes de la ligne dans la défausse du joueur	
hasCardLeft (Collection players)	bool	indique si un joueur a encore une carte	
getWinner (GameSixQP game)	?PlayerSixQP	renvoie le gagnant de la partie	
clearCards (array chosenCards)	void	supprime les cartes choisies du tableau	

4.3 Les contrôleurs

4.3.1 Structure

Le contrôleur du six qui prend permettra aux joueurs de réaliser les différentes actions sur le jeu.

 6QPController
<ul style="list-style-type: none">● Response selectCard(GameSixQP game, CardSixQP card)● Response placeCardOnRow(GameSixQP game, CardSixQP card, RowSixQP row)

4.3.2 Description des composants

Méthode	Type de retour	Description	Référence
selectCard(GameSixQP game, CardSixQP card)	HTMLResponse	Sélectionne la carte card pour le joueur exécutant la commande, dans la partie game	STB-GES-6QP-01Q
placeCardOnRow(GameSixQP game, CardSixQP card, RowSixQP row)	HTMLResponse	Place la carte card sur la ligne row dans la partie game si cela est valide	STB-GES-6QP-03 et 05 et 07

5 Architecture dynamique

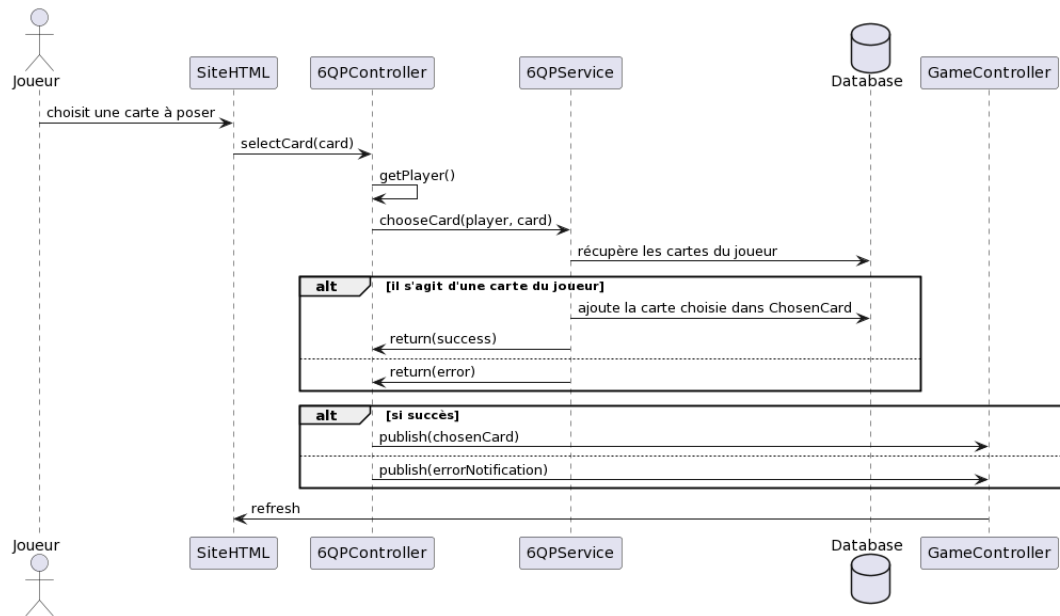


FIGURE 1 – Diagramme de séquence avec l'action "Le joueur choisit une carte"

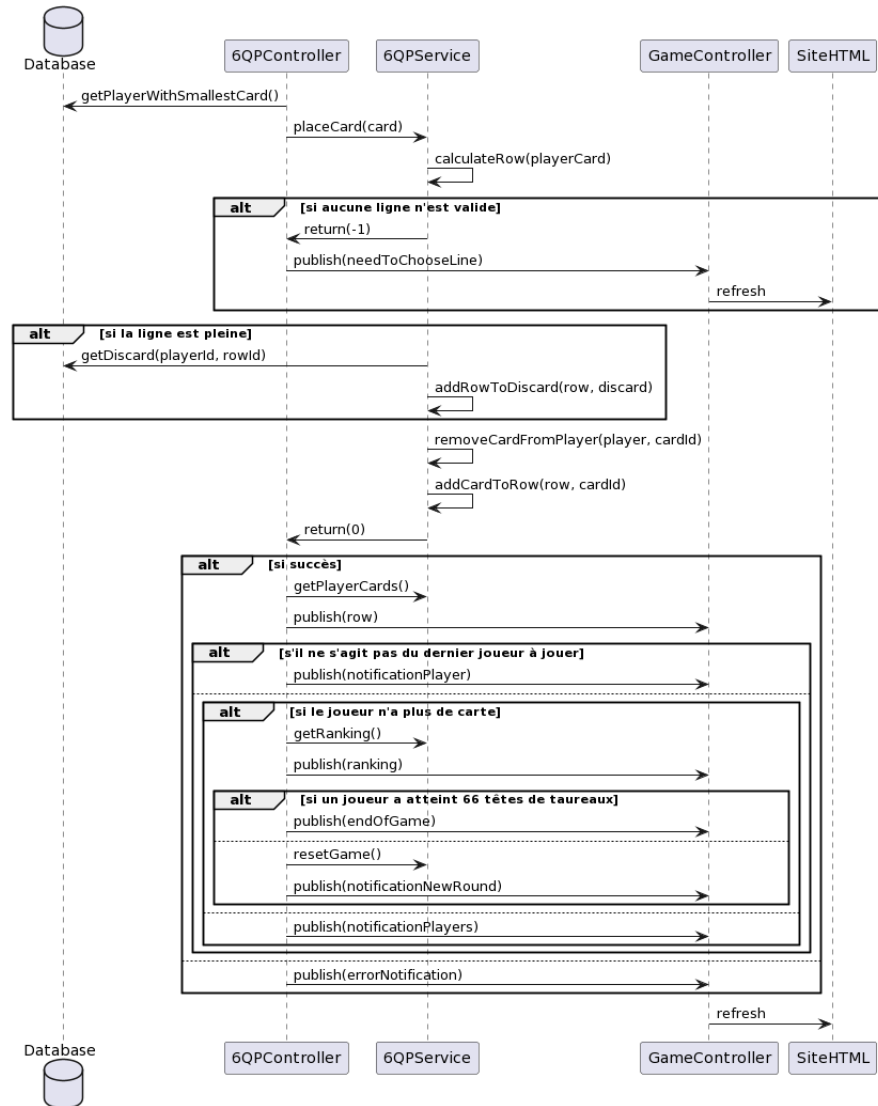
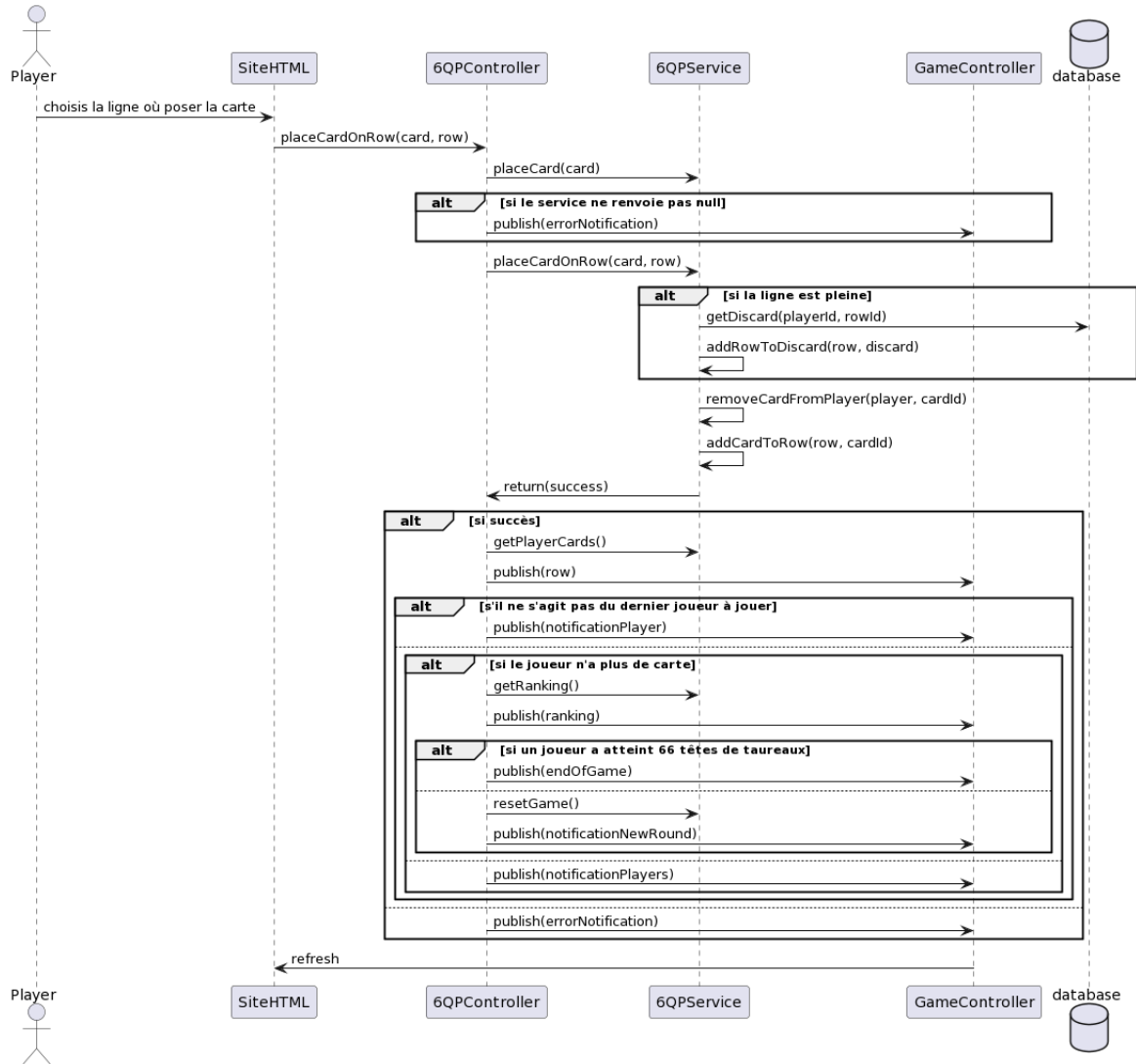


FIGURE 2 – Diagramme de séquence avec l'action "Le serveur place la carte automatiquement"

Diagramme de séquence avec l'action "Le joueur place la carte"



6 Maquette de vue

Nous avons également réalisé une première version d'une maquette de vue, notamment pour la représentation générale du plateau. Celle-ci sert notamment à avoir une idée de la disposition des différents éléments, et est amenée à être affinée et améliorée au cours de l'avancée du projet. Il s'agit donc ici d'une version minimaliste :



FIGURE 3 – Première version de la maquette de la vue du plateau principal du jeu 6 qui prend

Les carrés en transparence correspondent à des cases sans carte (pour la disposition des cartes sur le plateau principal). Le classement n'est pas trié mais le sera par ordre croissant (STB-GES-6QP-10) tel que le premier est celui qui a le moins de têtes de boeufs (STB-GES-6QP-11). De même, la partie à gauche représente la zone de dépôt des cartes et sera amenée à être modifiée pour être plus visible de manière détachée du plateau principal et personnel.

La main du joueur (ses cartes dans son plateau personnel) sont amenées à être présentée en arc de cercle dans une prochaine version.

La palette de couleurs sera également modifiée pour coller à celle du jeu initial (STB-INT-04), les couleurs choisies ici servent uniquement à identifier les zones.

Un joueur clique sur une des cartes de sa main et celle-ci se positionnera dans la zone de choix des cartes (face cachée tant que les autres joueurs n'auront pas choisi leur carte). Cette carte est unique, un joueur ne peut pas cliquer sur deux cartes différentes sinon aucune action ne se produira (STB-GES-6QP-01).

Au début de la partie, la main du joueur sera constitué de 10 cartes (STB-GES-6QP-04).

Enfin, le positionnement de la souris sur une carte permet d'agrandir son rendu. Une aide sera fournie au lancement du jeu récapitulant les règles (STB-INT-05). On pourra également retrouver cette aide dans un menu déroulant qui figurera dans une prochaine version.