Jeu de plateau tour par tour

RÉALISÉ PAR KAPNIST NGANTAH

ENCADRÉ PAR NICOLAS RIVIN

Rapport de projet technologique N°6: Jeu de plateau tour par tour

Sommaire

1 Cadrage du projet.

1.2 Résumé.

Dans ce projet, il est question pour nous de créer un jeu en ligne en Javascript, dans lequel deux joueurs évolueront chacun leur tour pour s'affronter. A l'issu de ce jeu, il ne pourra en rester qu'un seul.
Sur une carte, deux joueurs, les armes(minimum 4), et les cailloux(impasses) seront générés de façons aléatoire.
Ces armes pourront être récoltés par les joueurs qui passent dessus.
L'arme par défaut qui équipe le joueur doit infliger 10 points de dégât.

1.2 Enjeux et objectifs.

L'objectif de cette application est de:
-Réalise un jeu multi joueurs en ligne
-Mettre en place une application Javascript et jQuery

1.3 les livrables.

Pour développer le jeu de plateau, Kapnist Ngantah fournira les éléments suivants.
-l'architecture du jeu réutilisable
-les diagrammes UML de cette application
- code source complet du projet

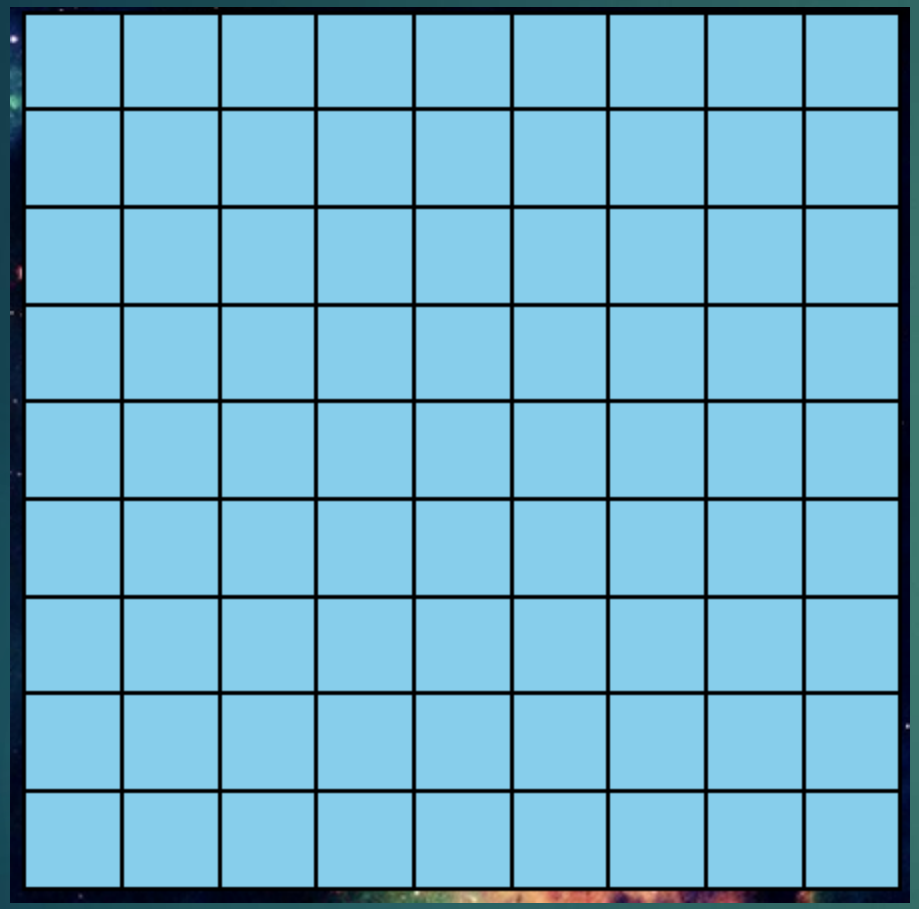
2 Présentation de mes objets

Afin d'assurer un fonctionnement optimal de notre jeu, il conviendrait de déterminer les différents éléments le constituant, ainsi nous pouvons noter:

-le plateau
-les blocks
- Kapnist(joueur 1)
-Genius(joueur 2)
les armes:

-le Sabre Egyptien
-Le Sabre magique
-la Hache de guerre
-Le Bouclier de Zelda

2 Présentation des objets



Plateau de jeu

Kapnist

Santé: excellente

Arme: Sabre Egyptien

Dégât arme: 10

Posture: attaque, défense

Point de vie: 100

Genius

Santé: excellente

Arme: Sabre Egyptien

Dégât arme: 10

Posture: attaque, défense

Point de vie: 100

Sabre Egyptien
(arme par default
de Kapnist)
dégat: 10points

Sabre Magique
(arme par defat
de Genius
degat: 10points

Bouclié
de Zelda
degat:
15points

Hâche de
guerre
degat:
20points

Nouvelle partie

Jeu de plateau tour par tour

Sante: Excellente

Arme: Sarbre Magique

Degat arme: 10

Posture Attaque Défense

Point de vie: 100

Sante: Excellente

Arme: Sarbre Egyptien

Degat arme: 10

Posture Attaque Défense

Point de vie: 100

Block ou impasse

3 Architecture du jeu

Génération de mes objets

Fonction principales:

function armes() : permet de générer mes arme

Function placePlayer() qui permet de générer mes joueurs

Function block() qui permet de générer mes blocks(impasses)

Fonctions utiles:

getRandomNumber() permet de générer un nombre aleatoire entre un min et un max

Function ajoutClassCasesvides() qui permet d'ajouter la classe isAcces au cases vides,

function ajoutClassImpasses() qui permet d'ajouter la classe impasse à nos block,

Gestion des mouvements

- Fonctions principales:
- eventPlayer1() pour assurer les déplacement du joueur1 et le ramassage des armes.
- eventPlayer2() pour assurer les déplacement du joueur1 et le ramassage des armes.

- Fonctions utiles:

celles qui gèrent les cases cliquables

- positionSpriteTop(posXJoueur1[0], posXJoueur1[1]);
- positionSpriteLeft(posXJoueur1[0], posXJoueur1[1]);
- positionSpriteRigth(posXJoueur1[0], posXJoueur1[1]);
- positionSpriteBottom(posXJoueur1[0], posXJoueur1[1]);
- Play() qui gre le bruitage

Gestion du combat

- Fonctions principales
- figth() qui déclanche le combat si les deux joueurs sont cote à cote
- combat() qui gère le combat
- Attaque1() et attaque2() qui permettent d'attaquer pour les joueurs1 et joueur2
- défense1() et défense2() qui permettent de défendre pour les joueurs1 et joueur2
- Fonctions utilses
- getRandomNumber() qui permet de savoir qui engage le combat
- Play() qui gère le bruitage

4-Diagrammes UML

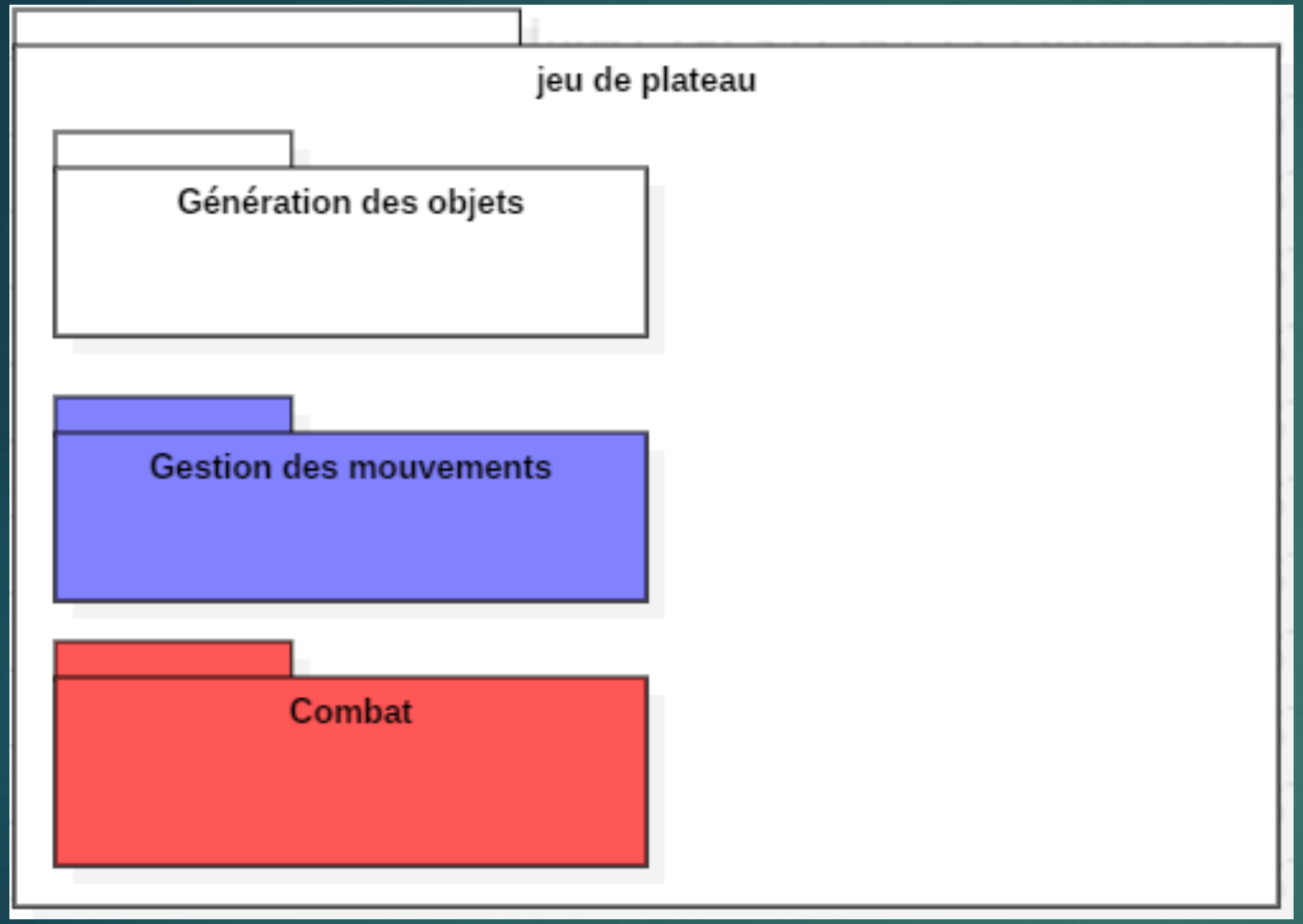


Diagramme de package

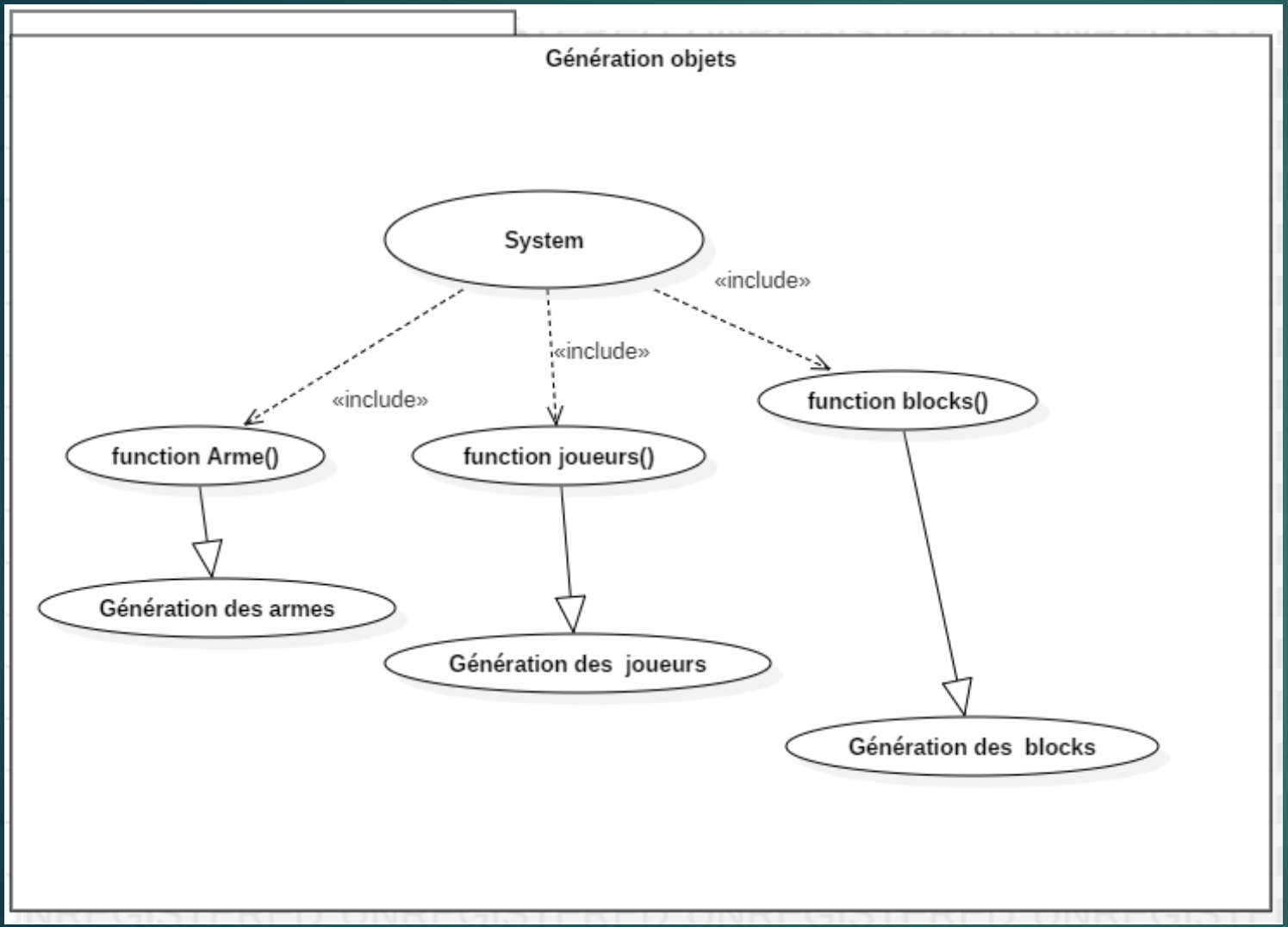


Diagramme du cas d'utilisation génération des objets

Diagramme du cas d'utilisation gestion des mouvements

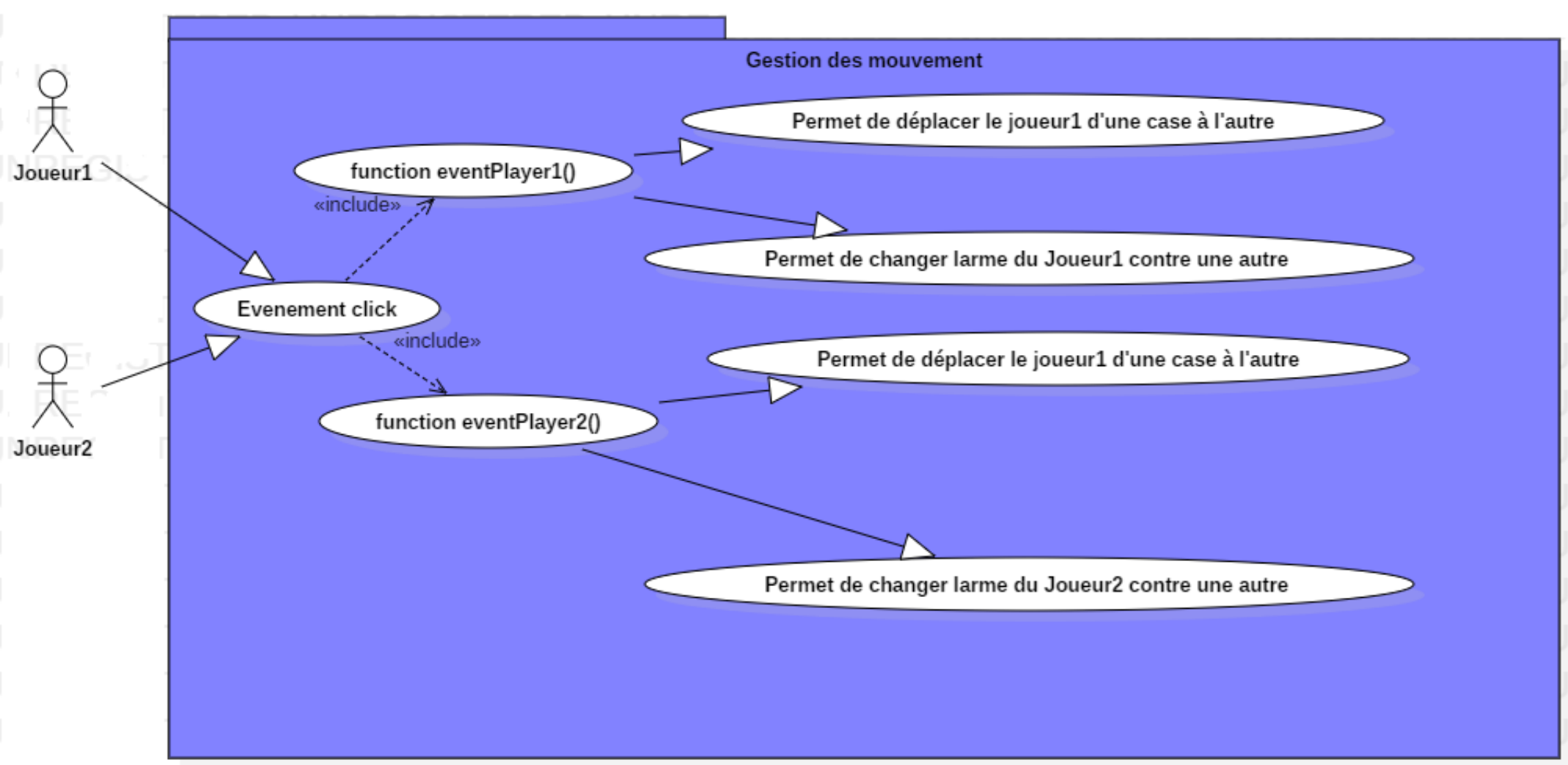
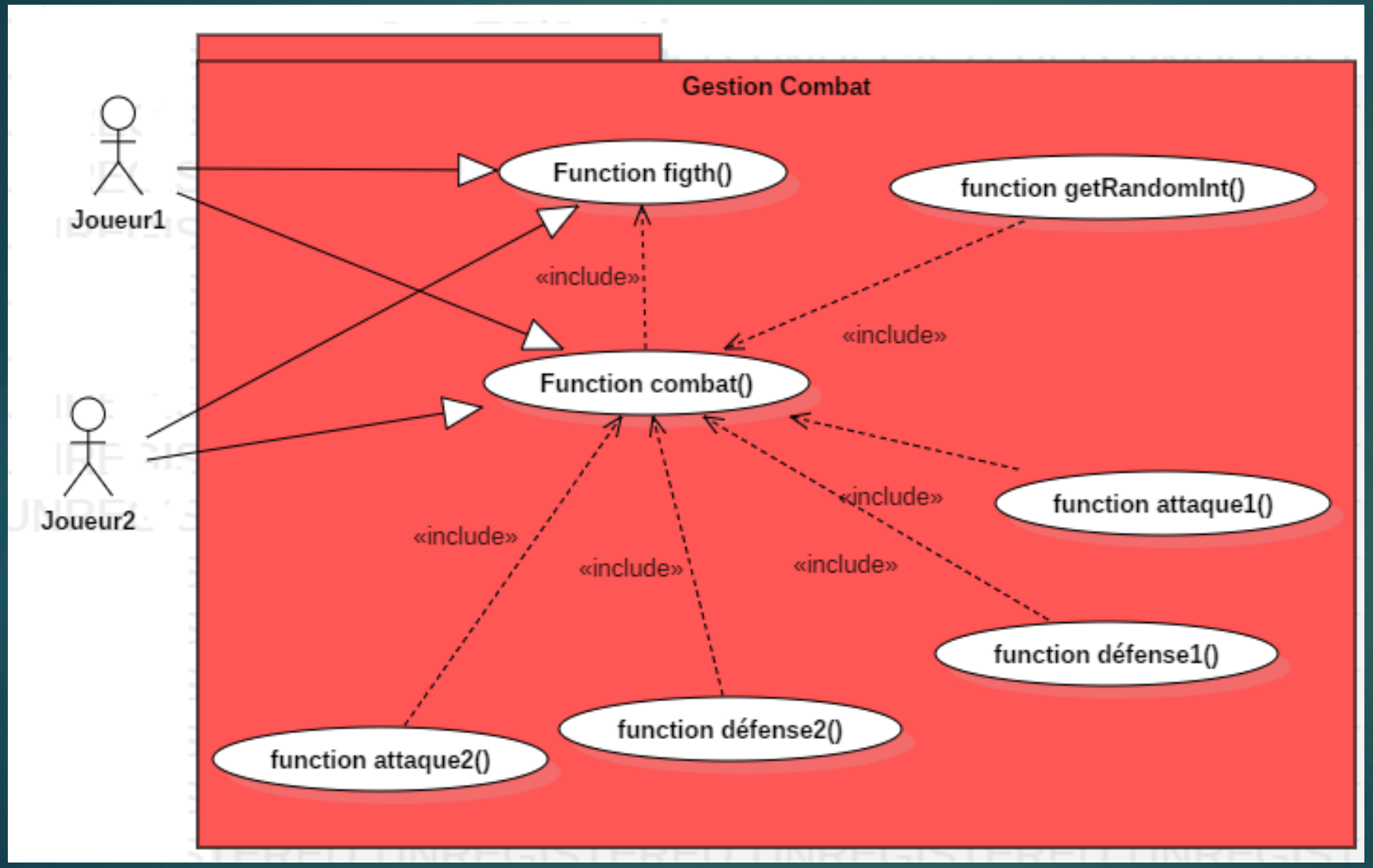


Diagramme du cas d'utilisation combat



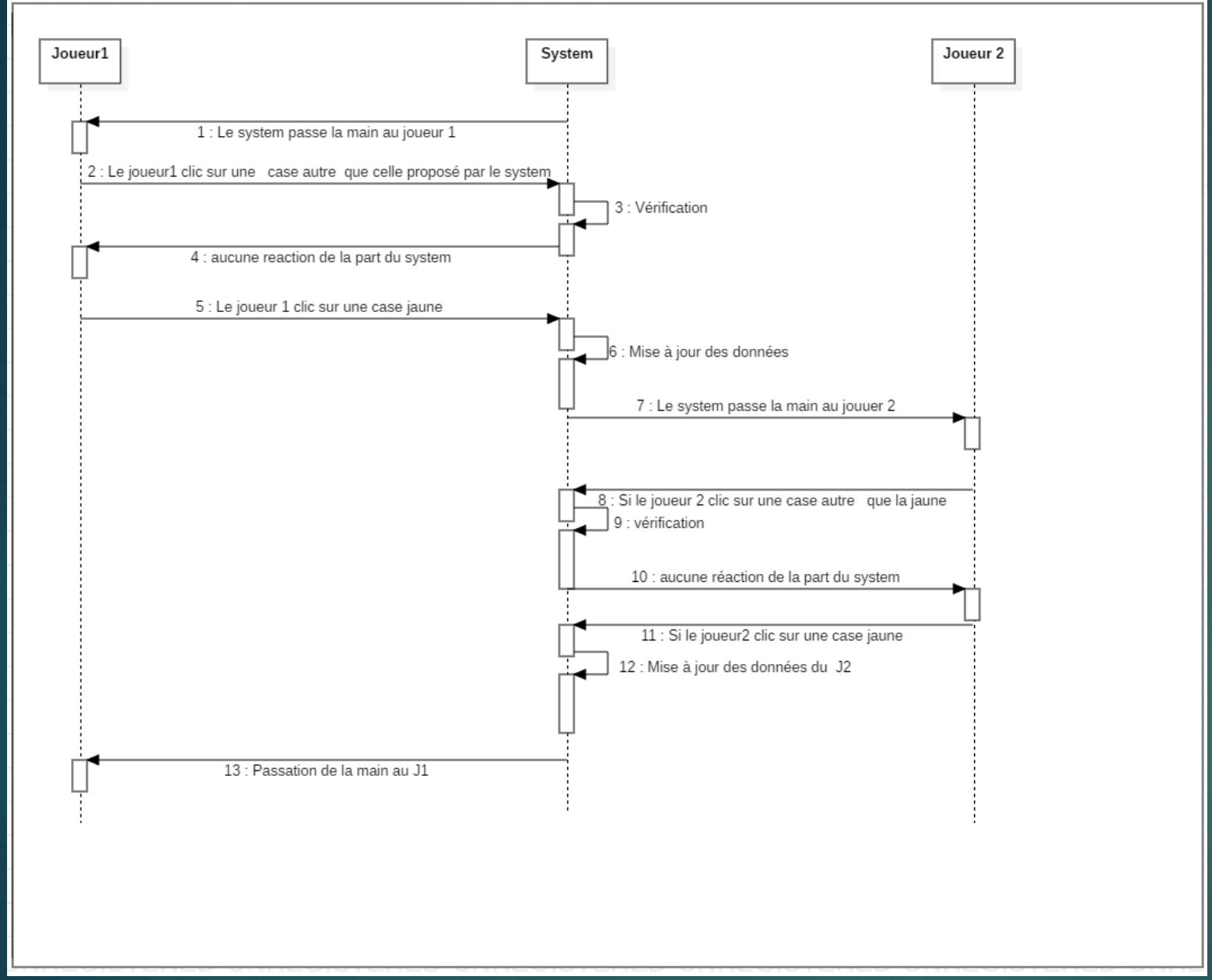


Diagramme de séquence Mouvement

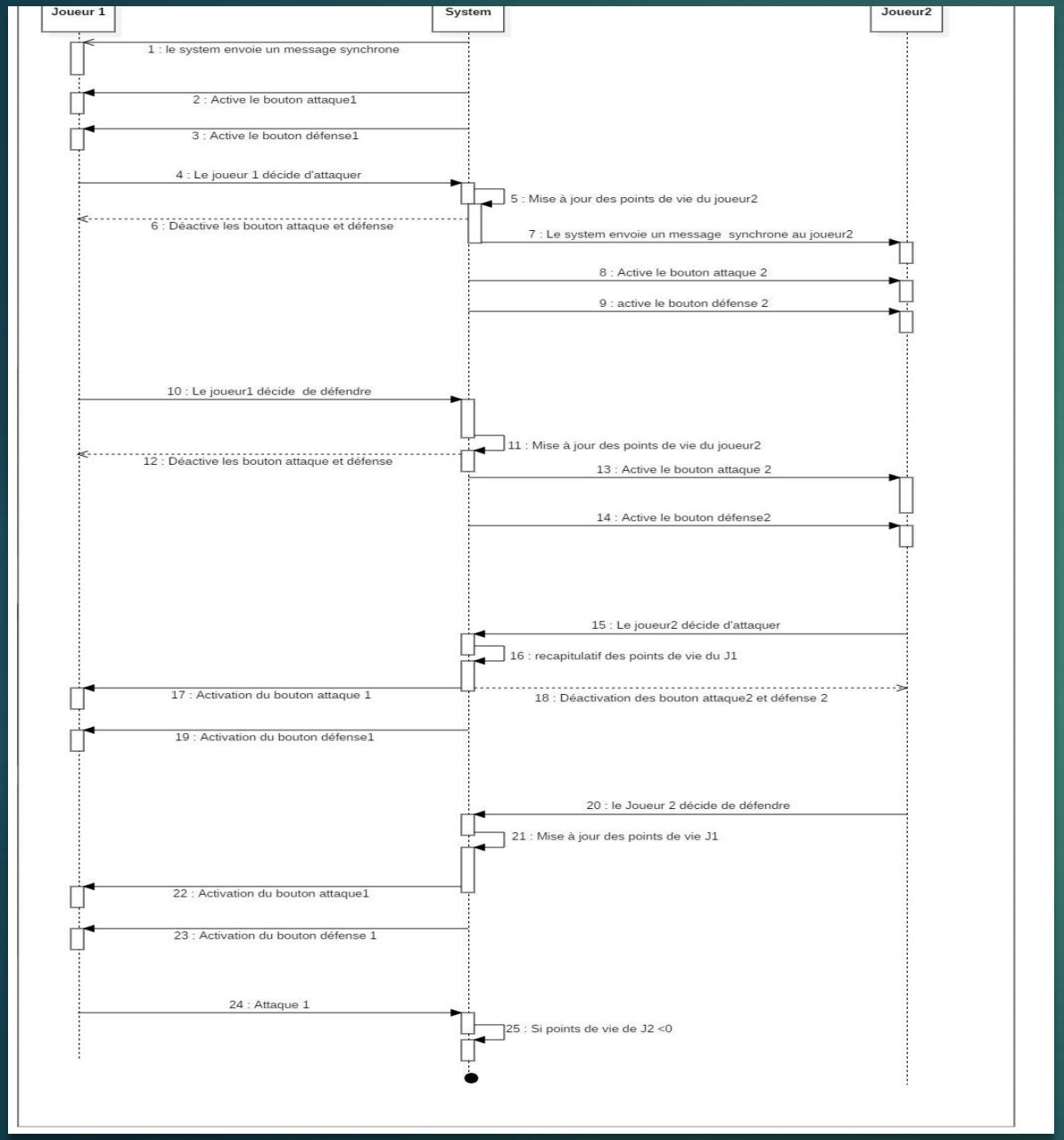


Diagramme de séquence combat

Présentation du code source et démonstration

Fin