



ÉCOLE NATIONALE SUPÉRIEURE DE L'ÉLECTRONIQUE  
ET DE SES APPLICATIONS

PROJET DU 15 SEPTEMBRE 2018 AU 15 JANVIER 2019

---

## Projet IS conception d'un jeu-vidéo de stratégie

---



*Auteurs*

SÉBASTIEN DEJEAN-GEORGES  
TIM KRIEF

*Professeur encadrant*

David Picard

## Table des matières

<b>1</b>	<b>Présentation générale</b>	<b>2</b>
1.1	Archétype . . . . .	2
1.2	Règles du jeu . . . . .	2
1.2.1	Le champ de bataille . . . . .	3
1.2.2	Les ressources . . . . .	3
1.2.3	Les unités . . . . .	4
1.3	Ressources . . . . .	5
<b>2</b>	<b>Description et conception des états</b>	<b>7</b>
2.1	Description des états . . . . .	7
2.1.1	Eléments fixes . . . . .	7
2.1.2	Eléments mobiles . . . . .	8
2.1.3	Etat général . . . . .	8
<b>3</b>	<b>L’affichage graphique</b>	<b>15</b>
3.1	Premiers rendus . . . . .	15
3.2	Descripteur de Tile . . . . .	15
<b>4</b>	<b>Le moteur</b>	<b>17</b>
4.1	Les commandes de jeu . . . . .	17
4.2	Les commandes d’affichage . . . . .	17
<b>5</b>	<b>Les IA</b>	<b>19</b>
<b>6</b>	<b>Modularisation</b>	<b>19</b>
<b>7</b>	<b>Progression</b>	<b>19</b>
	<b>Annexes</b>	<b>20</b>
<b>A</b>	<b>Quelques éléments à propos</b>	<b>20</b>

# 1 Présentation générale

## 1.1 Archétype

Notre but est de réaliser un jeu stratégique au tour par tour semblable à Advance Wars



FIGURE 3 – Sprite des tuiles

## 1.2 Règles du jeu

Notre jeu, "Until The Last Barrel", est un jeu de stratégie au tour par tour dont le but est de vaincre ses adversaires en capturant son Quartier Général (QG) sur le champ de bataille. Le dernier joueur survivant est donc déclaré vainqueur. Pour se faire le joueur dispose de trois ressources : l'argent, le pétrole et des cartes.

### 1.2.1 Le champ de bataille

Il s'agit d'une carte de tuiles (tile map) de différents types qui limitent ou améliorent la vitesse des unités des joueurs. On peut notamment différencier les tuiles infranchissables comme les cours d'eau, et aussi les montagnes ou les forêts pour certaines unités, des plaines, chemins, routes et pont qui enjambent les cours d'eau. Sur le champ de bataille se trouveront les QGs des joueurs, leurs unités, des villes, des usines et des puits de pétrole.

**Les bâtiments** Les bâtiments peuvent être capturés par les joueurs afin d'obtenir plus de ressources et autres avantages. Les puits de pétroles et les villes génèrent à chaque début de tour, respectivement du pétrole et de l'argent. Lorsqu'il veut jouer une carte le joueur peut placer des infanteries sur les villes et des unités motorisés sur les usines.

### 1.2.2 Les ressources

**Les cartes** Chaque joueur dispose d'un deck de carte duquel il pioche une carte à chaque début de tour. Une carte dispose d'un prix, qu'il est nécessaire de payer pour la jouer sur le champ de bataille, d'un type, de la description de ses effets.

- **Les cartes unités** permettent pour un prix indiqué sur la carte d'invoquer sur le champ de bataille la dite unité. Dans le cadre des blindés, ceux-ci ne peuvent être invoqués qu'à proximité des usines.
- **Les cartes soutiens** se résolvent immédiatement sur n'importe quelle cible valide. Il peut s'agir d'un tir d'artillerie qui endommage des unités, à des soin ou encore de faire piocher le joueur.
- **Les cartes équipements** permettent d'améliorer les caractéristiques d'une unité alliée déjà présente sur le champ de bataille.
- **Les cartes installations** sont un type de carte dont la fonction est de modifier les tuiles du champ de bataille. Il s'agit alors de pièges, bunker, pont, etc, afin de modifier le champ de bataille à son avantage. Les installations sont déployées à partir de certaines unités ayant la capacité nécessaire pour déployer l'installation voulue.

**L'argent** Au début de chaque tour, un joueur reçoit de l'argent en fonction de chaque ville qu'il a conquis, il peut alors dépenser cet argent lors de son tour afin de jouer des cartes. L'argent peut aussi être dépensé pour piocher des cartes.

**Le pétrole** Au début de chaque tour, un joueur reçoit du pétrole en fonction de chaque puit de pétrole qu'il possède, il peut alors utiliser ce pétrole pour déplacer des unités motorisées sur le champ de bataille. Si un joueur ne dispose pas de pétrole ces unités motorisées sont alors immobilisées voir détruites s'il s'agit d'unités aériennes.

### 1.2.3 Les unités

Il existe différentes unités jouables depuis sa main vers le champ de batailles avec différentes caractéristiques. De les unités de base ont :

- Des points de vie. L'unité est détruite si ses points de vie tombent à zéro.
- Des points de déplacement.
- Un niveau (conditionne les dégâts d'attaque par exemple).
- De la portée. Combien de tuiles peuvent être entre l'unité attaquante et l'unité attaquée.
- Des avantages face à certaines unités
- Des capacités spéciales
- Des équipements (ajoutés par d'autres cartes d'équipements)

### 1.3 Ressources

Afin de pouvoir représenter les différentes unités, tuiles et autres éléments du jeu, nous avons créé par nous-même des sprites. Nous utilisons une palette de 32 couleurs.



FIGURE 4 – Sprite des tuiles



FIGURE 5 – Sprite des unités

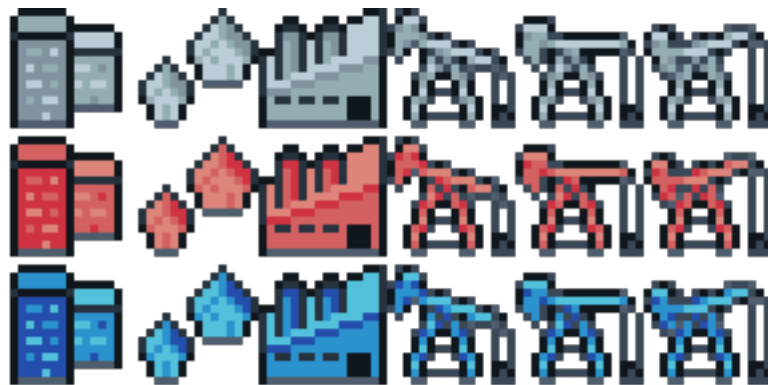


FIGURE 6 – Sprite des bâtiments

## 2 Description et conception des états

### 2.1 Description des états

Un état du jeu est décrit par des éléments fixes (le champs de bataille), par un ensemble d'éléments mobiles (unités) et par les ressources des joueurs. Il existe aussi un compteur de tour afin de déterminer quel joueur peut jouer. On a donc :

- Des coordonnées (x,y) pour les tuiles et les unités sur le champ de bataille
- Des classes différenciant les tuiles et les unités
- Des objets pour les joueurs déterminant les cartes, l'argent et le pétrole qu'ils possèdent.
- Un compteur pour les tours

#### 2.1.1 Eléments fixes

Le champs de bataille est formé par une grille d'éléments nommé tuiles ou cases. La taille de la grille est fixée au démarrage de la partie. Les types de cases sont :

- Les cases d'eau, de montagne et de forêt. Infranchissable par n'importe quelle unité.
- Les cases d'herbes qui permettent un déplacement standard.
- Les cases chemins qui permettent un déplacement plus rapide que celles d'herbe.
- Les cases routes qui permettent un déplacement plus rapide que celles d'herbe.
- Les ponts permettent de passer au dessus des cases d'eau et possèdent les mêmes caractéristiques que les chemins ou routes. Il existe.
- Les bâtiments se trouvent sur la carte. Ils peuvent soit être neutre, soit appartenir à un joueur. Parmi eux il existe : le QG, les usines, les villes et villages, et les puits de pétrole.
- Les installations sont des modifications du champs de bataille par les joueurs. Il peut s'agir de construire un pont, installer une barricade ou d'autres lignes de défense.



### 2.1.2 Eléments mobiles

Les éléments mobiles autrement dit les unités possèdent une direction, un status fatigué, des points de mouvements, une position, des équipements, des PVs, un niveau, une spécialité. Les cartes sont elles aussi des éléments mobiles. En effet un joueur jouant une carte la fait passer de sa main au champs de bataille. Il existe aussi des états intermédiaires quand le joueur décide de déplacer une unité ou jouer une carte.

- Les points de mouvements autorise les déplacement selon le type de terrain. A zéro, l'unité est immobile.
- Le status "fatigué" (exhausted) pour savoir si l'unité peut entreprendre des actions ou non.
- Les PVs détermine si l'unité est en vie. A zéro l'unité est détruite et retirée du champs de bataille.
- Le niveau est un état fixe qui détermine la puissance d'attaque de l'unité.
- Les équipements sont susceptibles de modifier les statistiques de l'unité équipé.

### 2.1.3 Etat général

Il existe quelques caractéristiques générales :

- Un compteur de tour pour déterminer le joueur actif
- Un status "victoire" qui affiche au vainqueur et au vaincu la fin du jeu ainsi que son résultat.
- Une liste de joueurs ainsi qu'une liste des objets sur le champs de bataille.

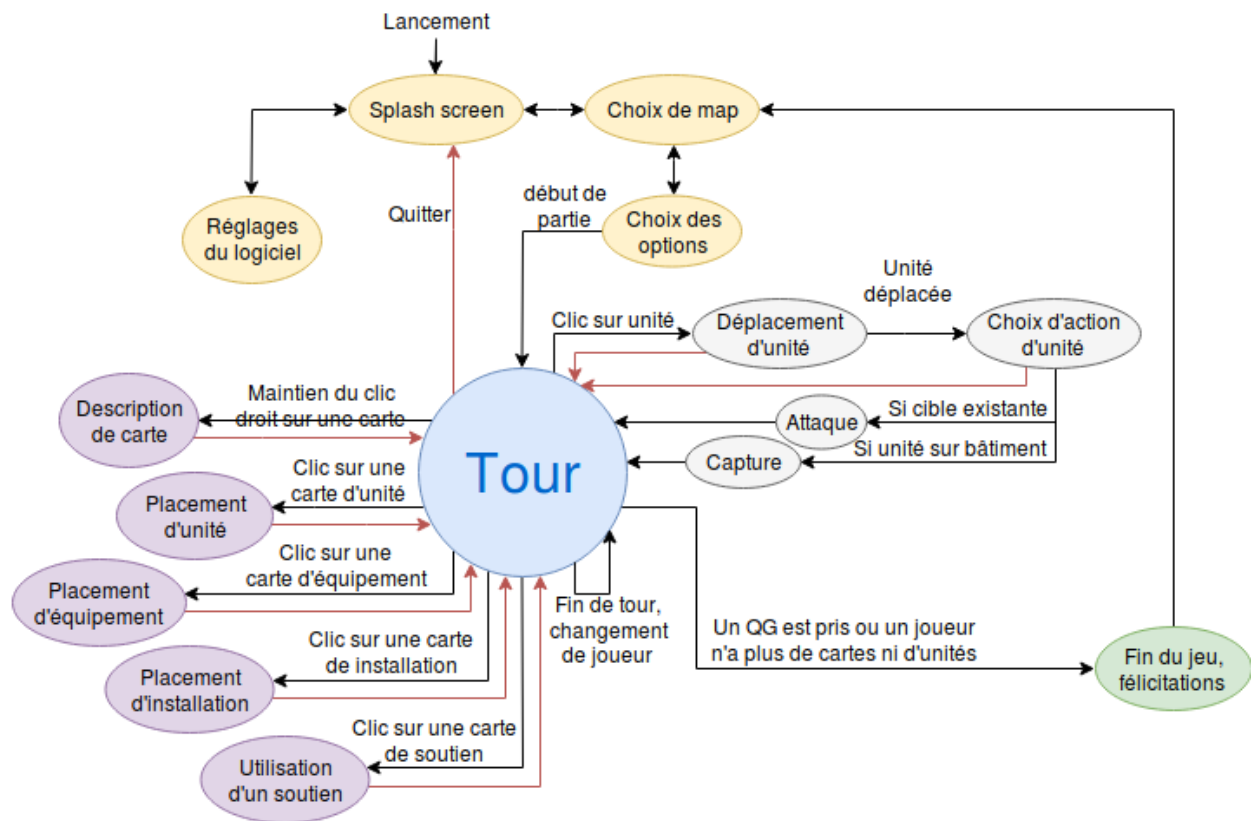


FIGURE 7 – Diagramme du tour

Description de l'état	Description de la transition
<b>Splash screen</b> Ecran d'accueil	On peut accéder au <b>Réglages du logiciel</b> ou au <b>Choix de map</b> grâce à un clic sur des icônes correspondantes
<b>Réglages logiciel</b> Cela permettra de régler l'affichage et d'autres caractéristiques de la fenêtre	On peut retourner au <b>Splash screen</b> via un clic sur un bouton retour.
<b>Choix de map</b> C'est la liste des différentes maps concues que le joueur peut sélectionner.	On peut retourner au <b>Splash screen</b> via un bouton retour ou arriver dans le <b>Choix des options</b> en confirmant son choix de map.
<b>Choix des options</b> C'est ici qu'on peut régler différentes options comme la difficulté de l'IA etc...	On peut retourner au <b>Choix de map</b> via un bouton retour, ou bien lancer la partie en commençant le premier <b>Tour</b>
<b>Tour</b> Le <b>Tour</b> permet à un joueur d'effectuer toutes les actions du jeu	Via la souris il peut : -maintenir le clic droit pour avoir la <b>Description de carte</b> de l'une de celle qu'il a en main. Relâcher le clic fera revenir à l'état <b>Tour</b> précédent le clic. -avec le clic gauche sélectionner une de ses <b>Placement</b> (unité, équipement, installation) ou <b>Utilisation de soutien</b> , cette action peut être annulée par un clic droit. -avec le clic gauche sélectionner une unité pour effectuer un <b>Déplacement d'unité</b> -si le joueur adverse n'a plus de QG, la <b>Fin du jeu</b> est déclenchée. -la fin du tour qui voit un nouveau <b>Tour</b> se lancer avec un autre joueur aux commandes.
<b>Placement d'unité</b> Suite à la sélection d'une carte d'unité, le joueur peut décider de la faire jouer sur une case autorisée.	Un clic gauche sur une case autorisé fera apparaître l'unité de la carte sur le champs de bataille, et la carte sera alors défaussée. On retourne alors à <b>Tour</b> .
<b>Placement d'équipement</b> Suite à la sélection d'une carte d'équipement, le joueur peut décider de la faire jouer sur une unité autorisée.	Un clic gauche sur une unité autorisé fera apparaître l'équipement de la carte dans les caractéristiques de l'unité, et la carte sera alors défaussée. On retourne alors à <b>Tour</b> .
<b>Placement d'installation</b> Suite à la sélection d'une carte d'installation, le joueur peut décider de la faire jouer sur une case autorisée.	Un clic gauche sur une case autorisé fera apparaître l'installation de la carte sur le champs de bataille, et la carte sera alors défaussée. On retourne alors à <b>Tour</b> .

<b>Utilisation de soutien</b> Suite à la sélection d'une carte de soutien, le joueur peut décider de la faire jouer sur une case ou une unité autorisée.	Un clic gauche sur l'entité autorisée déclenchera l'effet de la carte sur le champ de bataille, et la carte sera alors défaussée. On retourne alors à <b>Tour</b> .
<b>Déplacement d'unité</b> On a sélectionné une unité dans le but de la déplacer. Les cases où elle pourra se déplacer en fonction de ses points de mouvements	Un clic gauche permettra à l'unité de se diriger vers la case sélectionnée. Une fois le déplacement effectué, l'unité pourra effectuer une action dans <b>Choix d'action d'unité</b>
<b>Choix d'action d'unité</b> On a déplacé l'unité on souhaite maintenant qu'elle effectue une action dans le tour.	Un clic droit fera revenir à <b>Tour</b> et l'unité n'effectuera aucune action. Un clic gauche sur un bâtiment lancera la <b>Capture</b> si l'unité est sur un bâtiment ennemi. Un clic gauche sur une unité adverse à portée déclenchera une <b>Attaque</b> .
<b>Attaque</b> Un combat est lancé entre l'unité attaquante et l'unité attaquée	Une fois la résolution des dégâts terminée, on reprends le <b>Tour</b>
<b>Capture</b> Une unité a commencé à capturer un bâtiment adverse.	Une fois le siège mis en place, le <b>Tour</b> reprendra automatiquement.
<b>Fin du jeu</b>	Une fois Celle-ci déclenchée, on revient au <b>Choix de map</b>

<b>GameState</b> A pour but de décrire l'état du jeu	Attributs : map, players, existingCards, turn Méthodes : GameState, getMap, addPlayer, getPlayer, endTurn, getTurn, GenerateRessource, getPlayerTurnId
<b>FieldObject</b> C'est la classe qui décrit tout les objets du champs de bataille : leur emplacement, l'image associée, le propriétaire et le type	Attributs : position, sprite, owner, type Méthodes : FieldObject, setOwner, getPosition, setPosition, setObjectType
<b>Building</b> Un bâtiment peut être capturé par des unités afin qu'un joueur en prenne le contrôle	Attribut : health Méthode : Building, captured
<b>ObjectType</b> C'est une classe dénumération des types possibles que peut prendre un FieldObject	Attributs : HEADQUARTER, TOWN, FACTORY, OILWELL, INSTALLATION, UNIT
<b>Headquarter</b>	Attribut : moneyProduction Méthodes : Headquarter, setMoneyProduction, captured, setMoneyProduction, getMoneyProduction
<b>Town</b>	Attribut : moneyProduction Méthode :Town, Town (un deuxième constructeur), setMoneyProduction, setMoneyProduction, getMoneyProduction
<b>Factory</b>	
<b>OilWell</b>	Attribut :oilProduction Méthodes : OilWell, setOilProduction
<b>Installation</b>	Attribut : sourceCard
<b>Unit</b>	Attributs : equipments, headcount, sourceCard, direction, lastUpdate Méthodes :getHeadCount, move, attack, defence, capture, getDirection
<b>Direction</b> C'est une classe d'énumération qui permettra de changer les sprites en fonction des mouvements des unités	Attributs : UP, RIGHT, DOWN, LEFT

<b>Map</b>	Attributs : tileset, size, data Méthodes : Map, getSize, getdata, getTile, getTileset
<b>Tileset</b>	Attributs : tiles, resourcePath, columns, tileSize, resourceSize Méthodes : Tileset, getTile, getResourcePath, getColumns, getTileheight, getTileSize, getTilePosition, getTileIntRect
<b>Tile</b>	Attributs : speed Méthodes : Tile, getSpeed
<b>Player</b>	Attributs : deck, hand, money, oil, name, ownedFieldObjects, id, count Méthodes : Player, InitPlayer, getMoney, setMoney, getOil, setOil, getId, getName, addOwnedFieldObject, removeOwnedFieldObject, getOwnedFieldObject, getCount
<b>Deck</b>	Attributs : cards Méthodes : Deck, Deck (deuxième constructeurs), pull, remove
<b>Cards</b>	Attributs : cards Méthodes : Cards, getCard
<b>Card</b>	Attributs : id, title, cost, description, type Méthodes : Card, Card (un deuxième constructeur), playCard
<b>CardType</b> Classe d'énumération pour différencier les types de cartes	Attributs : UNIT, EQUIPMENT, INSTALLATION, SUPPORT
<b>EquipmentCard</b>	Attributs : level, strengths, headCount Méthodes : EquipmentCard, playCard
<b>UnitCard</b>	Attributs : level, unitType, strengths, movesByTurn, oilCost Méthodes : UnitCard, playCard
<b>InstallationCard</b>	Attributs : stopOpponent, damage, iterations Méthodes : InstallationCard, playCard
<b>SupportCard</b>	Attributs : damage, heal, moneyLoss, moneyGain, oilLoss, oilGain Méthodes : SupportCard, playCard



### 3 L'affichage graphique

#### 3.1 Premiers rendus

L'affichage de la carte fonctionne grâce à l'implémentation d'un render. La carte est donc définie sous la forme de tileset dans le state, et importée dans le render afin de l'afficher. Ainsi on constate qu'on a l'affichage dans une fenêtre de la carte ainsi que de fieldobject.

#### 3.2 Descripteur de Tile

Un affichage en bas à droite permet en lisant la position de la souris de donner les informations sur une case comme sa texture, son coût en déplacement et sa défense.



FIGURE 9 – Démonstration du render



<b>Gstate</b> A pour but de décrire l'état du jeu pour l'affichage	Attributs : gmap, gplayers, state Méthodes : GState, getGmap, getGPlayers, getTurn, GePlayerCount,
<b>GRender</b> C'est la classe qui décrit la fenêtre	Attributs : window, cursor, cameraPosition, cameraZoom, armyFont, vcrFont Méthodes : GRender, initTextures, getWindow, setCursor, display, getCameraPosition, updateCameraPostion, getCameraZoom, updateCameraZoom
<b>GPlayer</b> Permet de déterminer quel objets de quels joueurs afficher	Attribut : Méthode : GPlayer, draw
<b>GMAP</b> C'est une classe ayant pour but d'afficher la carte	Attributs : map Méthodes : GMap, draw, getTile, getTileDescription
<b>TextureManager</b> Permet de gérer les textures	Attribut : textures, order Méthode : TextureManager (constructeur et destructeur), getLength, GetTexture, loadTexture

## 4 Le moteur

Les mouvements ont été implémentés via un moteur. L'appui permet de modifier l'état en cours vers un nouvel état. En faisant ainsi on sépare l'affichage qui ne fait que lire, du moteur qui permet à la fois de lire l'état et de le modifier.

### 4.1 Les commandes de jeu

Il existe deux commandes afin d'agir sur le jeu :

- les commandes de direction qui permettent le mouvement des unités avec les touches directionnelles.
- Ctrl+Z permet d'annuler les déplacements du tour en cours.
- la commande de fin de tour avec la touche entrée.

### 4.2 Les commandes d'affichage

Pour modifier l'affichage de la carte ainsi que déplacer la vision du joueur, on peut utiliser le pad ou la souris.

- L'aggrandissement de la carte est modifiable en étirant la fenêtre.
- shift+molette(scroll sur pad) permet de faire défiler la carte horizontalement.
- ctrl+molette (zoom sur pad) permet de zoomer ou dézoomer.

<b>GameEngine</b> A pour but de gérer les commandes du joueur afin de modifier l'état du jeu	Attributs : currentState, commands, executedCommands Méthodes : GameEngine(constructeur et desctrueteur), getState, update, cancel, CancelFromPlayer, cancelTurn, addCommand, saveHistory, loadHistory
<b>Command</b> C'est la classe qui décrit les commandes disponibles	Attributs : playerTriggeringId, commandType (qui peut prendre deux valeurs 0 pour fin de tour et 1 pour les mouvements) Méthodes : Command, execute, cancel, getPlayerTriggeringId, getCommandType, save
<b>CommandEndTurn</b> Permet de finir le tour en cours	Attribut : Méthode : CommandEndTurn, execute, cancel, save
<b>CommandMove</b> C'est une classe ayant pour but de commander les mouvement	Attributs : direction Méthodes : CommandMove, execute, cancel, save

## 5 Les IA

Deux types d'IA ont été créés. La première euristique diminue la distance de Manhattan avec les unités adverses sans tenir compte des obstacles. La seconde essaiera que chacune de ses unités attaque les unités adverses les plus proches d'elles.

<b>CPU</b> A pour but de gérer les IA pour générer les mouvements des joueurs non-humains	Attributs : state, engine, playerId, targetedPlayerId Méthodes : CPU, run
<b>RandomCPU</b> C'est une IA dont les déplacements aléatoires	Attributs : playerTriggeringId, Méthodes : run, RandomCPU
<b>HeuristicCPU</b> IA qui cherche à diminuer la distance de manhattan avec sa cible	Attribut : turnSteps Méthode : run, HeuristicsCPU
<b>DeepCPU</b> C'est une IA plus intelligente qui est capable de trouver un chemin optimisé pour atteindre les unités adverses	Méthodes : run, DeepCPU

## 6 Modularisation

La mise en place de thread permet au moteur de fonctionner en parallèle du reste du jeu, notamment de l'affichage. Le fichier replay.txt a été renommé pour devenir savefile.json.

## 7 Progression

Suite au développement nous avons décidé de ne pas implémenter la mécanique de carte pour le rendu final mais les fichiers nécessaires à sa création sont déjà présents.

# Annexes

## A Quelques éléments à propos

Il a fallu modifier `dia2code` à un moment du développement afin de prendre en compte les `enum class`. Nous avons aussi importé la bibliothèque `yaml-cpp` dans le projet.