



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

1	Présentation générale	2
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
2	Description et conception des états	7
2.1	Description des états	7
2.1.1	Éléments fixes	7
2.1.2	Éléments mobiles	8
2.1.3	État général	8
3	L’affichage graphique	15
3.1	Premiers rendus	15
3.2	Descripteur de Tile	15
4	Le moteur	17
4.1	Les commandes de jeu	17
4.2	Les commandes d’affichage	17
5	Les IA	19
6	Modularisation et network	21
7	Progression	21
	Annexes	22
A	Quelques éléments à propos	22

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

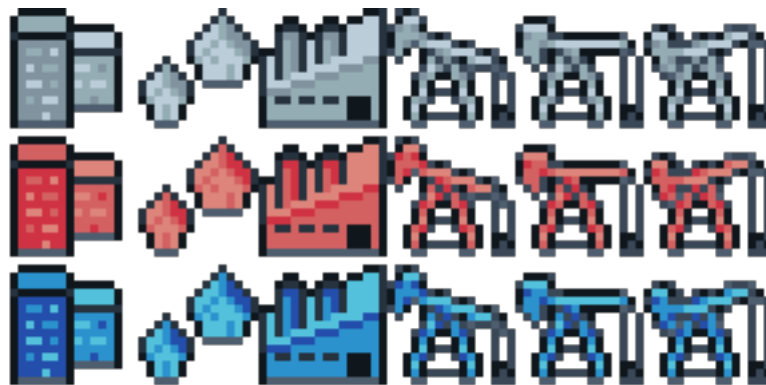


FIGURE 6 – Sprite des bâtiments

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.

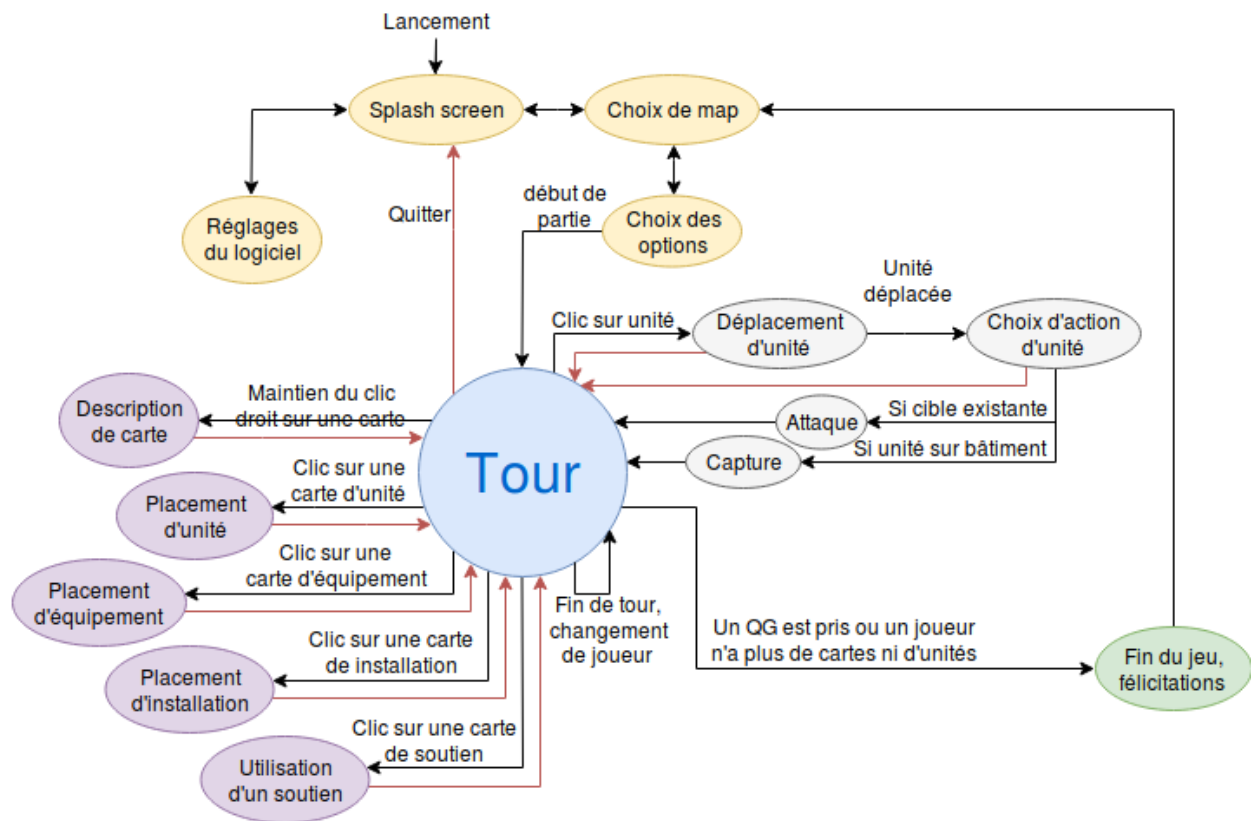


FIGURE 7 – Diagramme du tour

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

Utilisation de soutien 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 Tour .
Déplacement d'unité 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 Choix d'action d'unité
Choix d'action d'unité On a déplacé l'unité on souhaite maintenant qu'elle effectue une action dans le tour.	Un clic droit fera revenir à Tour et l'unité n'effectuera aucune action. Un clic gauche sur un bâtiment lancera la Capture si l'unité est sur un bâtiment ennemi. Un clic gauche sur une unité adverse à portée déclenchera une Attaque .
Attaque Un combat est lancé entre l'unité attaquante et l'unité attaquée	Une fois la résolution des dégâts terminée, on reprends le Tour
Capture Une unité a commencé à capturer un bâtiment adverse.	Une fois le siège mis en place, le Tour reprendra automatiquement.
Fin du jeu	Une fois Celle-ci déclenchée, on revient au Choix de map

GameState A pour but de décrire l'état du jeu	Attributs : map, players, existingCards, turn Méthodes : GameState, getMap, addPlayer, getPlayer, endTurn, getTurn, GenerateRessource, getPlayerTurnId, getDistance
FieldObject 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, getDirection, setDirection, getOwner, getMovePoints, setMovePoints, resetMovePoints, getNormalMovePoints
Building Un bâtiment peut être capturé par des unités afin qu'un joueur en prenne le contrôle	Attribut : health Méthode : Building, captured
ObjectType C'est une classe énumération des types possibles que peut prendre un FieldObject	Attributs : HEADQUARTER, TOWN, FACTORY, OILWELL, INSTALLATION, UNIT
Headquarter	Attribut : moneyProduction Méthodes : Headquarter, setMoneyProduction, captured, setMoneyProduction, getMoneyProduction
Town	Attribut : moneyProduction Méthode :Town, Town (un deuxième constructeur), setMoneyProduction, setMoneyProduction, getMoneyProduction
Factory	
OilWell	Attribut :oilProduction Méthodes : OilWell, setOilProduction
Installation	Attribut : sourceCard
Unit	Attributs : equipments, headcount, sourceCard, direction, lastUpdate Méthodes :getHeadCount, move, attack, defence, capture, getDirection
Direction C'est une classe d'énumération qui permettra de changer les sprites en fonction des mouvements des unités	Attributs : UP, RIGHT, DOWN, LEFT

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

Gstate A pour but de décrire l'état du jeu pour l'affichage	Attributs : gmap, gplayers, state Méthodes : GState, getGmap, getGPlayers, getTurn, GePlayerCount,
GRender 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
GPlayer Permet de déterminer quel objets de quels joueurs afficher	Attribut : Méthode : GPlayer, draw
GMAP C'est une classe ayant pour but d'afficher la carte	Attributs : map Méthodes : GMap, draw, getTile, getTileDescription
TextureManager Permet de gérer les textures	Attribut : textures, order Méthode : TextureManager (constructeur et destructeur), getLength, GetTexture, loadTexture

GameEngine 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
Command 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
CommandEndTurn Permet de finir le tour en cours	Attribut : Méthode : CommandEndTurn, execute, cancel, save
CommandMove C'est une classe ayant pour but de commander les mouvement	Attributs : direction Méthodes : CommandMove, execute, cancel, save

5 Les IA

Plusieurs types d'IA ont été créés afin de rendre simuler des joueurs.

La première est aléatoire et exécute donc une commande parmi celles auxquelles elle a accès.

La deuxième euristicque diminue la distance de Manhattan avec les unités adverses sans tenir compte des obstacles.

La troisième essaiera que chacune de ses unités attaque les unités adverses les plus proches d'elles.

La quatrième suit l'algorithme Astar afin de déterminer le plus court chemin dans un graphe de noeuds. Elle est capable de gérer chacune des unités de chaque joueur en prenant en compte l'existence des autres unités et de ce fait savoir qu'elles sont bloquantes.



FIGURE 11 – Rendu final

Voici le rendu tel qu'il est au lancement. On constate par ailleurs une ATH qui permet de voir les informations d'une case lorsqu'on passe le curseur dessus.

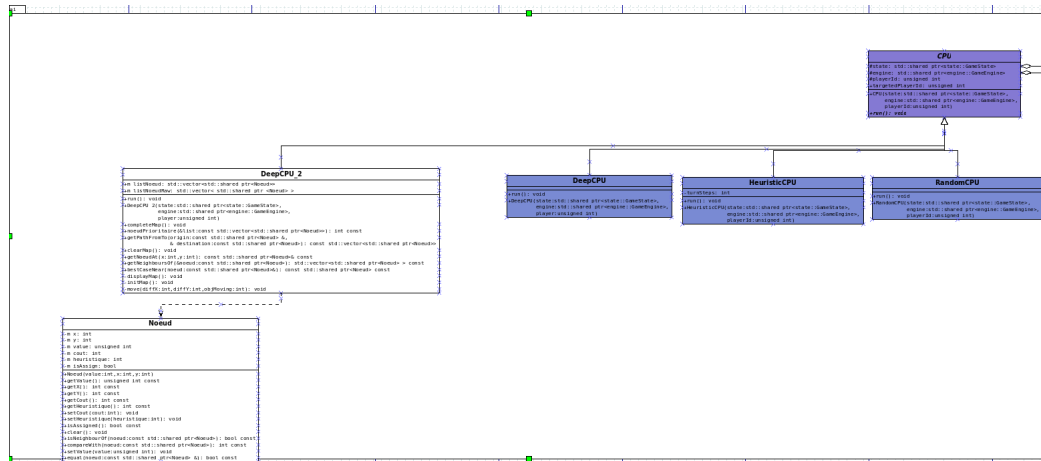


FIGURE 12 – Rendu final

CPU A pour but de gérer les IA pour générer les mouvements des joueurs non-humains	Attributs : state, engine, playerId, targetedPlayerId Méthodes : CPU, run
RandomCPU C'est une IA dont les déplacements aléatoires	Attributs : playerTriggeringId, Méthodes : run, RandomCPU
HeuristicCPU IA qui cherche à diminuer la distance de manhattan avec sa cible	Attribut : turnSteps Méthode : run, HeuristicCPU
DeepCPU C'est une IA plus intelligente qui est capable de trouver un chemin optimisé pour atteindre les unités adverses	Méthodes : run, DeepCPU
DeepCPU_2 C'est une IA qui suit l'algorithme A* pour trouver un chemin optimisé	Attributs : m_listNoeud, m_listNoeudRaw Méthodes : run, DeepCPU_2, completeMap, noeudPrioritaire, getPathFromTo, clearMap, getNoeudAt, getNeighbourOf, bestCaseNear, displayMap, displayMap, initMap, move
Noeud C'est une IA qui suit l'algorithme A* pour trouver un chemin optimisé	Attributs : m_x, m_y, m_value, m_cout, m_heuristique, m_isAssign Méthodes : Noeud, getValue, getX, getY, get Cout, getHeuristique, set Cout, setHeuristique, isAssigned, clear, isNeighbourOf, compareWith, setValue, equal

6 Modularisation et network

La mise en place de thread permet au moteur, à l'IA ainsi qu'au render de fonctionner parallèlement. On a donc la possibilité de faire fonctionner les animations, les changements d'affichage en même temps sans que cela n'impacte la fluidité du jeu. Le fichier replay.txt demandé pour le livrable de modules a été renommé pour devenir savefile.json. La possibilité de jouer en ligne sur un serveur.

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.