Projet Logiciel Transversal

Thomas CARCAGNO – Fabien EYSSARTIER

Table des matières

1 Objectif

1.1 Présentation générale

Présenter ici une description générale du projet. On peut s'appuyer sur des schémas ou croquis pour illustrer cette présentation. Éventuellement, proposer des projets existants et/ou captures d'écrans permettant de rapidement comprendre la nature du projet.

Archétype : Le jeu proposé se base sur la bataille navale, dans une version légèrement modifiée.

Détails : Ce jeu se joue en un contre un. Dans la suite, nos deux joueurs seront appelés “Joueur A” pour celui qui contrôle les bateaux et “Joueur B” pour celui qui contrôle les canons.

- **Interface graphique**

L’interface graphique présente la carte de jeu avec une vue plongeante.
Joueur A voit ses bateaux sur l’eau ainsi que la côte floutée par de la brume. Il voit aussi un quadrillage en semi-transparence qui permet au joueur de savoir où il peut se déplacer.
Joueur B voit ses canons posés un peu en hauteur sur la côte rocheuse. La mer est visible au large, mais est légèrement dissimulée par de la brume qui lui empêche de voir les bateaux. Il voit également un quadrillage correspondant au cases sur lesquelles il peut tirer et qui délimitent les zones atteignables par chaque canon (par exemple, les limites peuvent être représentées en utilisant deux couleurs différentes pour les contours des cases limites des deux canons). S’il touche un bateau ennemi, la case ciblée restera en surbrillance pendant deux tours de jeu. De plus, il a une petite carte visible sur son écran qui lui indique s'il y a des bateaux dans la moitié gauche ou dans la moitié droite de la carte.

- **Menu principal**

Le menu principal présente différents choix possibles :

- Celui de lancer une partie en solo (contre une IA). Le joueur accède alors à un autre menu qui propose de lancer une partie rapide (règles standard, durée courte à moyenne, difficulté facile) ou bien de configurer les paramètres d'une partie (nombre de bateaux, difficulté, etc.).
- Celui de lancer une partie en multijoueur. De la même manière que pour le mode un joueur, une partie avec des règles standard est proposée, mais les joueurs ont aussi la possibilité de régler les paramètres de la partie. Un mode classé est également proposé mais dans ce cas, les règles sont fixées et uniformisées au préalable pour tous les joueurs. Le joueur entre alors dans une file d'attente. Il est placé contre un adversaire de niveau plus ou moins équivalent, et, dans le cas d'une partie classée, le gain de points est adapté en fonction de la différence de niveaux entre les deux joueurs.
- Celui de régler les options de jeu : les contrôles, avec la possibilité d'assigner des touches du clavier ou de jouer à la souris, mais aussi les options graphiques (qualité, dynamique, effets visuels) et sonores (volume de la musique, volume d'animation de chaque action, etc.). Pour les graphismes, les effets sont désactivables, et plusieurs niveaux de qualité sont proposés : faible, intermédiaire et fort. Pour le son, une musique en background est présente lors de chaque partie (un, voire plusieurs thèmes de piraterie ou de navigation standard, libres de droit), mais le joueur peut la désactiver. De même, les sons d'ambiance et d'effets sont désactivables.
- Celui de consulter le classement en ligne. La classement pourrait être fait soit à partir d'un Elo (équivalent au système existant pour les échecs), soit en fonction de ligues (comme on le retrouve dans la plupart des jeux en ligne). Le classement est lui-même subdivisé en trois catégories : Joueur A, Joueur B et global. Deux types de classement sont présentés au joueur : le classement des parties normales, qui présentent des informations générales, et celui des parties classées, qui permet en plus de déterminer le rang des joueurs.
- Celui de consulter ses succès. Les succès seraient des missions secondaires que le joueur aura eu l'occasion de réaliser en jouant. Par exemple, détruire un certain nombre de bateaux ou gagner dix parties à la suite.
- Celui de quitter le jeu.

1.2 Règles du jeu

Présenter ici une description des principales règles du jeu. Il doit y avoir suffisamment d'éléments pour pouvoir former entièrement le jeu, sans pour autant entrer dans les détails . Notez que c'est une description en « français » qui est demandé, il n'est pas question d'informatique et de programmation dans cette section.

- **Règles de base**

Joueur A possède trois bateaux qu'il va pouvoir déplacer sur la carte de jeu. Au premier tour, il doit placer ses trois bateaux sur un côté de la carte et à chaque tour, il a le droit de réaliser deux mouvements ; il peut choisir d'avancer ou de tourner l'un de ses bateaux. Son objectif est de traverser la carte et atteindre l'autre côté sans perdre ses bateaux.
Joueur B possède deux canons fixés dès le début de la partie à deux positions différentes et il peut les utiliser une fois chacun à chaque tour pour tirer un boulet dans une zone limitée (plus de la moitié de la carte est atteignable par chaque canon, mais, par exemple, le canon de droite ne peut pas tirer dans l'angle gauche le plus éloigné).

- **Déroulement d'une partie**

Au cours de chaque manche, les joueurs obtiennent des points. Les points permettent de départager le vainqueur. Les points sont attribués en fonction du nombre de fois que les bateaux sont touchés ou que les boulets de canon sont esquivés selon le rôle tenu par le joueur. Des points bonus peuvent être distribués pour récompenser un joueur. Par exemple, il peut recevoir des points s'il réussit à toucher un bateau plusieurs fois de suite ou à esquiver les boulets de canon sur plusieurs tours consécutifs.

- **Compétences spéciales**

Des compétences peuvent être implémentées sous la forme de bonus pour chaque camp :
- Joueur B peut recevoir un bonus si, à l'approche de la fin de la partie, il n'a toujours pas réussi à détruire les bateaux. Par exemple, il pourrait avoir la possibilité de tirer plein de boulets de canon sur toute une ligne/colonne ou bien de pouvoir augmenter la cadence de ses tirs.

- Joueur A peut également recevoir un bonus s'il n'est pas parvenu à protéger ses bateaux. Par exemple, lorsqu'il ne lui reste plus qu'un bateau et que celui-ci a déjà été touché, il peut avoir le droit d'effectuer plus de déplacements par tour ou bien, si il perd un bateau dans la première moitié de la carte, il pourra prédire où atterriront les boulets de canon pendant quelques tours.

- **Abandon et déconnexion**

En jeu, une option de reddition est disponible pour permettre à un joueur d'abandonner la partie, ce qui met instantanément fin à la partie sur une victoire de l'adversaire, en plus de lui donner des points bonus.

Pour les déconnexions au serveur, un délai est offert au joueur pour lui permettre de se reconnecter (30 secondes à 1 minute), afin de ne pas pénaliser les personnes qui jouent dans de moins bonnes conditions que les autres. Au terme de ce délai, le joueur déconnecté est systématiquement déclaré perdant, comme s'il avait abandonné la partie. Le score du joueur déconnecté ne sera pas pris en compte dans son score global.

- **"Loin du clavier" (AFK)**

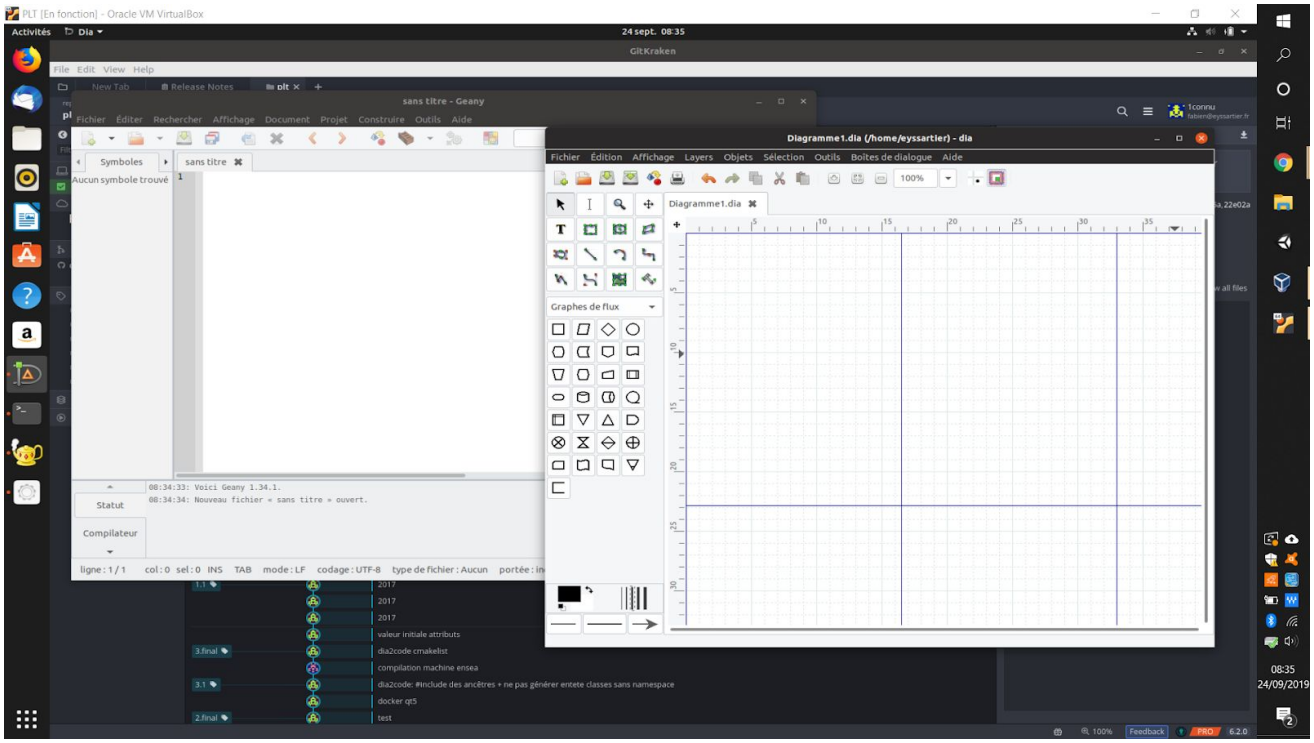
Un tour de jeu est limité (30s en partie classique, 1 minute en partie classée). Une fois le temps de jeu écoulé, si le joueur n'a toujours pas fait son coup, il est considéré que le joueur a passé son tour et si cela se reproduit 3 tours de suite, le joueur est considéré comme ayant abandonné la partie en cas de partie classique, et le temps passe de 1 minute à 30 secondes par tour de jeu en partie classée (si cela se reproduit encore 3 fois, idem qu'en partie classique).

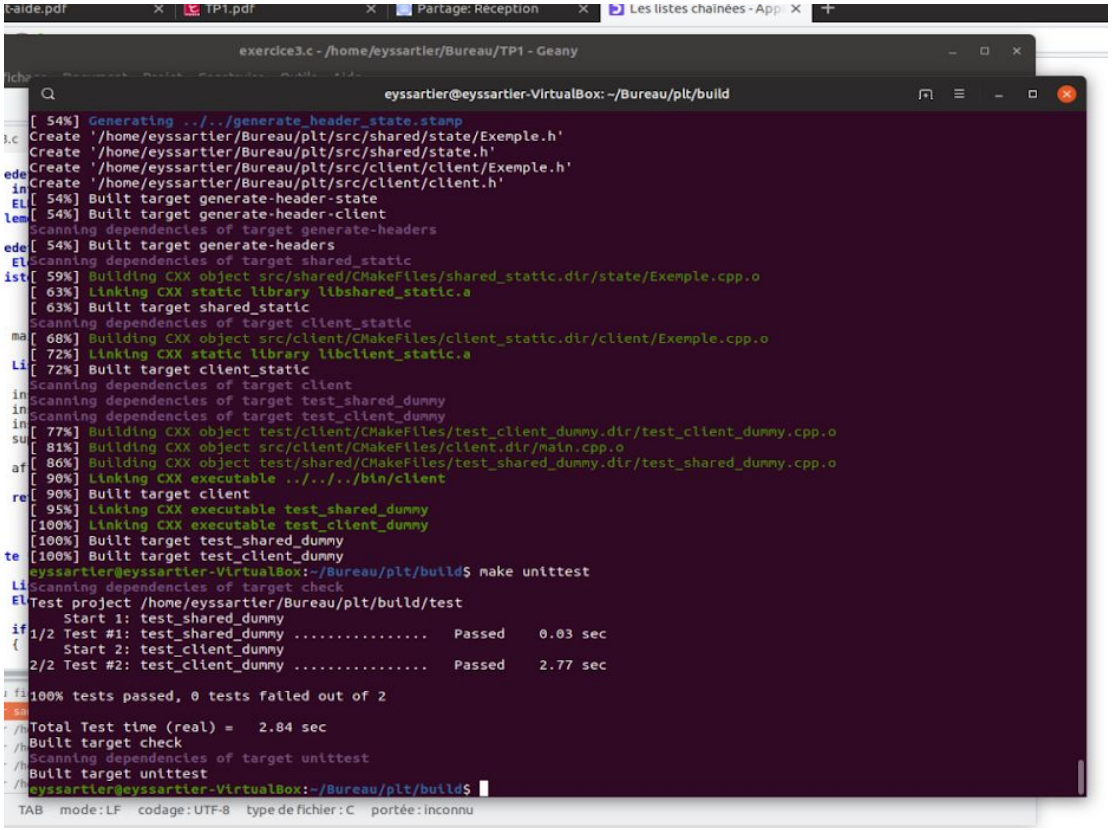
- **Passer son tour**

Le joueur a la possibilité de passer son tour s'il le souhaite. Cependant, cette action est signalée par un message à l'adversaire et si un joueur passe son tour trois tours de suite, il est considéré comme ayant abandonné la partie.

1.3 Conception Logiciel

Présenter ici les packages de votre solution, ainsi que leurs dépendances.





2 Description et conception des états

L'objectif de cette section est une description très fine des états dans le projet. Plusieurs niveaux de descriptions sont attendus. Le premier doit être général, afin que le lecteur puisse comprendre les éléments et principes en jeux. Le niveau suivant est celui de la conception logiciel. Pour ce faire, on présente à la fois un diagramme des classes, ainsi qu'un commentaire détaillé de ce diagramme. Indiquer l'utilisation de patron de conception sera très apprécié. Notez bien que les règles de changement d'état ne sont pas attendues dans cette section, même s'il n'est pas interdit d'illustrer de temps à autre des états par leur possibles changements.

2.1 Description des états

2.2 Conception logiciel

2.3 Conception logiciel : extension pour le rendu

2.4 Conception logiciel : extension pour le moteur de jeu

2.5 Ressources

Illustration 1: Diagramme des classes d'état

3 Rendu : Stratégie et Conception

Présentez ici la stratégie générale que vous comptez suivre pour rendre un état. Cela doit tenir compte des problématiques de synchronisation entre les changements d'états et la vitesse d'affichage à l'écran. Puis, lorsque vous serez rendu à la partie client/serveur, expliquez comment vous aller gérer les problèmes liés à la latence. Après cette description, présentez la conception logicielle. Pour celle-ci, il est fortement recommandé de former une première partie indépendante de toute librairie graphique, puis de présenter d'autres parties qui l'implémente pour une librairie particulière. Enfin, toutes les classes de la première partie doivent avoir pour unique dépendance les classes d'état de la section précédente.

3.1 Stratégie de rendu d'un état

3.2 Conception logiciel

3.3 Conception logiciel : extension pour les animations

3.4 Ressources

3.5 Exemple de rendu

Illustration 2: Diagramme de classes pour le rendu

4 Règles de changement d'états et moteur de jeu

Dans cette section, il faut présenter les événements qui peuvent faire passer d'un état à un autre. Il faut également décrire les aspects lié au temps, comme la chronologie des événements et les aspects de synchronisation. Une fois ceci présenté, on propose une conception logiciel pour pouvoir mettre en œuvre ces règles, autrement dit le moteur de jeu.

4.1 Horloge globale

4.2 Changements extérieurs

4.3 Changements autonomes

4.4 Conception logiciel

4.5 Conception logiciel : extension pour l'IA

4.6 Conception logiciel : extension pour la parallélisation

Illustration 3: Diagrammes des classes pour le moteur de jeu

5 Intelligence Artificielle

Cette section est dédiée aux stratégies et outils développés pour créer un joueur artificiel. Ce robot doit utiliser les mêmes commandes qu'un joueur humain, ie utiliser les mêmes actions/ordres que ceux produit par le clavier ou la souris. Le robot ne doit pas avoir accès à plus information qu'un joueur humain. Comme pour les autres sections, commencez par présenter la stratégie, puis la conception logicielle.

5.1 Stratégies

5.1.1 Intelligence minimale

5.1.2 Intelligence basée sur des heuristiques

5.1.3 Intelligence basée sur les arbres de recherche

5.2 Conception logiciel

5.3 Conception logiciel : extension pour l'IA composée

5.4 Conception logiciel : extension pour IA avancée

5.5 Conception logiciel : extension pour la parallélisation

6 Modularisation

Cette section se concentre sur la répartition des différents modules du jeu dans différents processus. Deux niveaux doivent être considérés. Le premier est la répartition des modules sur différents threads. Notons bien que ce qui est attendu est un parallélisation maximale des traitements: il faut bien démontrer que l'intersection des processus communs ou bloquant est minimale. Le deuxième niveau est la répartition des modules sur différentes machines, via une interface réseau. Dans tous les cas, motivez vos choix, et indiquez également les latences qui en résulte.

6.1 Organisation des modules

6.1.1 Répartition sur différents threads

6.1.2 Répartition sur différentes machines

6.2 Conception logiciel

6.3 Conception logiciel : extension réseau

6.4 Conception logiciel : client Android

Illustration 4: Diagramme de classes pour la modularisation