Projet libre 2e année MIC :

<nom du jeu>

## Objectifs pédagogiques

La réalisation de ce projet comporte plusieurs objectifs pédagogiques :

* Conception et réalisation d’une intelligence artificielle permettant de mettre en œuvre des algorithmes de décision.
* Conception objet et réalisation d’un projet complet en équipe, en mettant en place des outils pour le travail en groupe : Git/SVN et en considérant les problématiques de code liées au travail de groupe (protection du code (private etc …), mise en place de conventions, structuration et découpage du projet, création d’interfaces, mise en place de tests unitaires).

Ce projet est aussi en lien avec nos objectifs liés à notre orientation professionnelle, nous permettant d’acquérir une (maigre) expérience dans le monde du jeu vidéo que nous envisageons d’intégrer à l’issue de notre parcours.

## Technologies utilisées

Deux choix principaux sont envisagés pour les technologies utilisées

* C++/SFML : Utilisation d’un langage très prisé de l’industrie (en particulier vidéo ludique) et confrontation à des problèmes de gestion de la mémoire. Temps de développement plus long.
* C#/XNA  : Plus grande vitesse de développement, permet de maximiser le temps passé sur les problèmes algorithmiques purs (en particulier l’IA) et nous permettraient d’aller plus loin. Possibilité d’interfaçage avec un langage fonctionnel dérivé d’OCAML (F#).

## Déroulement du jeu

Deux joueurs jouent tour à tour et contrôlent des unités ainsi qu’une fabrique d’unités.

Le jeu se déroule sur une carte en forme de grille, où sont placées les unités, l’environnement, ainsi que divers « bonus ».



Figure 1. Exemple de source d'inspiration pour le fonctionnement du jeu : Advance Wars sur Game Boy Advance.

L’objectif principal est la récolte d’un maximum de points en une période donnée au travers de plusieurs objectifs (destructions d’unités, obtention d’objectifs). La réalisation de ces objectifs rapporte également de l’argent permettant de construire des unités.

Le jeu s’effectue en tour par tour durant lesquels les joueurs peuvent contrôler leurs unités individuellement dans un temps imparti de l’ordre de la minute, ainsi que créer de nouvelles unités.

Les joueurs pourront être des humains ou des intelligences artificielles.

## Environnement

Plusieurs éléments seront présents dans l’environnement :

* Mer (non traversable par les unités de base, mais potentiellement traversables par les unités marines).
* Mines (permettant l’obtention d’argent grâce à des unités de minage).
* Fabriques (permettant la réalisation d’unités moyennant un coût en argent).
* Bonus (permettant des augmentations de statistiques des unités).
* Murs (empêchant les mouvements de toutes les unités).
* Zones ralentissantes (ralentissant la plupart des unités).
* Zones infligeant des dégâts.

## Unités

Plusieurs types d’unités pourront exister et seront influencées différemment par l’environnement et auront des statistiques différentes et accomplir des rôles différentes.

Toutes les unités peuvent être endommagées ou détruites par les combattants, et leur destruction rapporte des points et de l’argent au joueur.

### Stats de base

* Points de vie
* Dégâts / Portée / Type de dégâts.
* Défense() -> fonction du type de dégâts.
* Points de déplacement / tour.

### Types d’unités

* Combattants : unités au sol permettant la destruction de structures et d’autres unités.
* Transporteurs : unités permettant le transport d’unités au travers de zones particulières (mer, zone ralentissante etc…).
* Mineurs : unités au sol permettant l’obtention d’argent et de points lorsqu’ils minent des structures particulières.
* Constructeurs : Permet la fabrication de structures (fabrique d’unités / réparation de mines).
* Fabriques d’unités : Permet la fabrication d’unités en échange d’argent.

## Gameplay

## Déplacement d’unités / Intéractions

Pour déplacer une unité, le joueur effectue un premier clic sur l’unité. Les cases possibles pour le déplacement sont alors mises en valeur par effet graphique. Un deuxième clic gauche permet à l’unité d’effectuer le mouvement. Un clic droit annule action précédente.

Les déplacements ne sont possibles que vers des cases traversables, ne comportant aucune unité. Par défaut, les unités empruntent **le chemin le plus court**. Les unités peuvent effectuer plusieurs déplacements mais sont limitées par le nombre de cases traversées.

Si le deuxième clic est effectué sur une unité avec laquelle plusieurs actions sont possibles, un menu apparaît permettant au joueur de sélectionner l’action désirée.

Une unité ne peut effectuer qu’une attaque par tour, et ne peut pas interagir plus d’une fois avec le même élément au cours du même tour.

## Fabrique d’unités

Lorsqu’un joueur clique sur une fabrique, un menu apparaît indiquant la liste des unités fabricables ainsi que leur coût. L’unité fabriquée peut être invoquée sur l’une des quatre cases directement adjacentes à la fabrique. Si aucune case n’est libre, la fabrication ne peut pas avoir lieu.

Les unités ne peuvent être déplacées au premier tour de leur apparition.

## HUD

Seront affichées :

* Nom, Score et Argent des différents joueurs. Le joueur actuellement entrain de jouer est mis en surbrillance.
* L’icône de l’unité sélectionnée ainsi que ses stats.

Au dessus de chacune des unités peuvent être affichées (activable / désactivable) leur jauge de santé. Les jauges de santé sont affichées lorsque les unités sont soignées ou subissent des dégâts.

# Découpage du projet

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Livrable | Temps estimé / Date de livraison | Priorité |
| Spécification fonctionnelle | Document indiquant la quasi-totalité des fonctions du programme. | 05/02/2014 | +++ |
| Modélisation objet du programme. | Diagramme UML. | 08/02/2014 | +++ |
| Réalisation d’une maquette | Fonctions implémentées :   * Chargement/Affichage de la map. * Déplacement des unités. * Le moteur tour par tour. * Affichage du HUD. | 10/03/2014 | +++ |
| Implémentation des règles du jeu. | Fonctions implémentées :   * Gestion des attaques et de la portée des attaques. -> Guigui * Les conditions de victoire. -> Guigui | 17/03/2014 | +++ |
| Implémentation d’unités de base, de certaines portions de l’environnement. | Ajout des unités de base, et des morceaux d’environnement tels que la mer, les zones ralentissantes. |  | +++ |
| Implémentation d’une IA de base. | Ajout d’une IA capable de prendre des décisions simples afin de maximiser le nombre de points gagnés. |  | ++ |
| Réalisation d’un menu principal | Menu principal permettant de sélectionner le mode de jeu, le nombre de participants, la carte etc… |  | + |
| Réalisation d’animations | Animation des tirs, des unités, du monde. |  | + |
| Réalisation d’Easter Eggs | Parce que c’est cool. |  | + |