Jeu vidéo
Cahier des charges.

DE

BONI Bryan | WATTIN Tristan | GALLAND Alexis |
CHOUANIA Mehdi | LEBLOIS Vincent

Contents

Partie 1 : Ce que l'on veut réaliser

I. Histoire & Type de jeu

Le type de jeu vidéo que l'on veut réaliser est un jeu mi de **combat** (exemple : Street fighter), mi **RPG** (exemple : Pokémon) dont l'histoire sera la suivante (résumée) :

« Dans la partie sud du royaume d'Ascential, dans la décharge mécanique de la ville de Taruto vie notre héros de 15ans et ses grands parent. Un jour où il travaillé il trouve un corps qui s'avérai être un androïde de la garde impérial, en essayant de le réactivé celui-ci ce réveil et essaye de tuer notre héros, qui tente de s'enfuir en vain. Alors qu'il pensait sa fin arrivé un nouvel acteur entre en scène, Dan (un célèbre héros de guerre, une idole aux yeux de notre jeune héros) qui avait assisté à la scène de loin, et le sauve in extrémiste d'une mort certaine. S'en suis un furieux combat contre l'automate (tutoriel).
Quelque minute plus tard le héros qui, c'était évanoui un peu après le début du combat ce réveil, les deux combattant avait disparu et seul l'épée de Dan était encore là. Plusieurs jours après l'incident Dan fut déclaré mort.

L'histoire se passe 5ans plus tard où l'on joue cette fois si le héros désormais âgée de 20ans qui à travers plusieurs mésaventure ce retrouvera à participée au tournoi commémoratif du grand héros de guerre Dan Gears ».

II. Liste des choses à réaliser

Notre jeu devra comporter plusieurs choses essentiel pour son bon fonctionnement, ça bonne jouabilité et pour une bonne appréciation.

Il devra être doté d'une IHM structuré et organisé pour les parties combat et déplacement (menu, inventaire, point de vie, etc…) elles aussi optimisé et implémenté au jeu, la gestion d'objet, de statistiques, d'IA (amies et/ou ennemies) et des combos & coups spéciaux (partie jeu de combat).

Il faudra aussi réaliser différents sprites (surtout pour la partie déplacement où certain auront des effets différents en fonction de la situation.

De la musique d'ambiance ainsi que des bruitages pour améliorer l'expérience de jeu de l'utilisateur et ainsi rendre celui-ci plus appréciable.

Une partie du développement sera aussi consacrée à la création d'animation et cinématique (pas très dur à réaliser, mais permettant une compréhension plus facile des choses pour le joueur).

Partie 2 : La partie déplacement (hors

(4 mois 3personnes)

I. Mécanique de jeu

Le déplacement :

Le personnage aura distinctement deux types de déplacements : le déplacement hors-combat, à savoir dans les villes par exemple, sur le déplacement en combat (voir rubrique concernée).
Les deux déplacements sont différents, on a donc deux fonctions totalement différentes à coder.
Parlons tout d'abord du déplacement hors-combat.
Le personnage joueur devra pouvoir se déplacer à travers la ville, selon un axe X,Y, dans les quatre directions. Arrivé à certains endroits de la carte, le jeu devra effectuer une transition de carte (cf. les tuiles) pour que le personnage puisse continuer à se déplacer.
Lors de ses déplacements, le personnage joueur ne doit pas avoir la possibilité de passer à travers un mur, ou une maison : on doit donc gérer les collisions et les éléments de la carte.

Une Interaction avec les PNJ :

Le personnage jouable aura l'occasion, au fil de ses aventure, de croisé plusieurs « PNJ » (personnage non joueur) avec lesquels il pourra leurs parler, soit pour la suite de l'histoire, soit pour du « background » ou des phrases anodine, soit pour recevoir des objets ou entrée en combat avec eux, et cela dépendra du type de PNJ, amie ou ennemie et principale ou secondaire.

Une IHM :

Il nous faudra réaliser dans un dernier temps une **IHM** qui permettra au joueur de naviguer entre les différentes options que lui offre le jeu (comme la gestion des objets et de ses statistiques quand il passera un niveau), mais aussi si il rencontre quelque problème ,ou le souhaite juste, créer une IHM pour les options général (volume, résolution, touches, etc…).

Des objets :

De plusieurs type, consommables, équipements et objets clé qui pour les deux premiers auront des effets bien précis sur le personnage, qui pourront être consulté à partir de l'inventaire, quand au dernier ils auront une incidence sur le déroulement de l'histoire et parfois devrons être obligatoirement obtenu pour permettre au joueur d'avancer.

II. A réaliser

Le déplacement :

Pour coder la partie déplacement hors-combat, nous nous aiderons de la méthode d'utilisation des tuiles (explication à la rubrique concernée), c'est-à-dire ce déplacée de tuiles en tuiles et assigner les pnj à certaines tuiles et aussi si nécessaire leurs affecter certain effet différents (changement de zone, infranchissable, effet sur le personnage).

Une Interaction avec les PNJ :

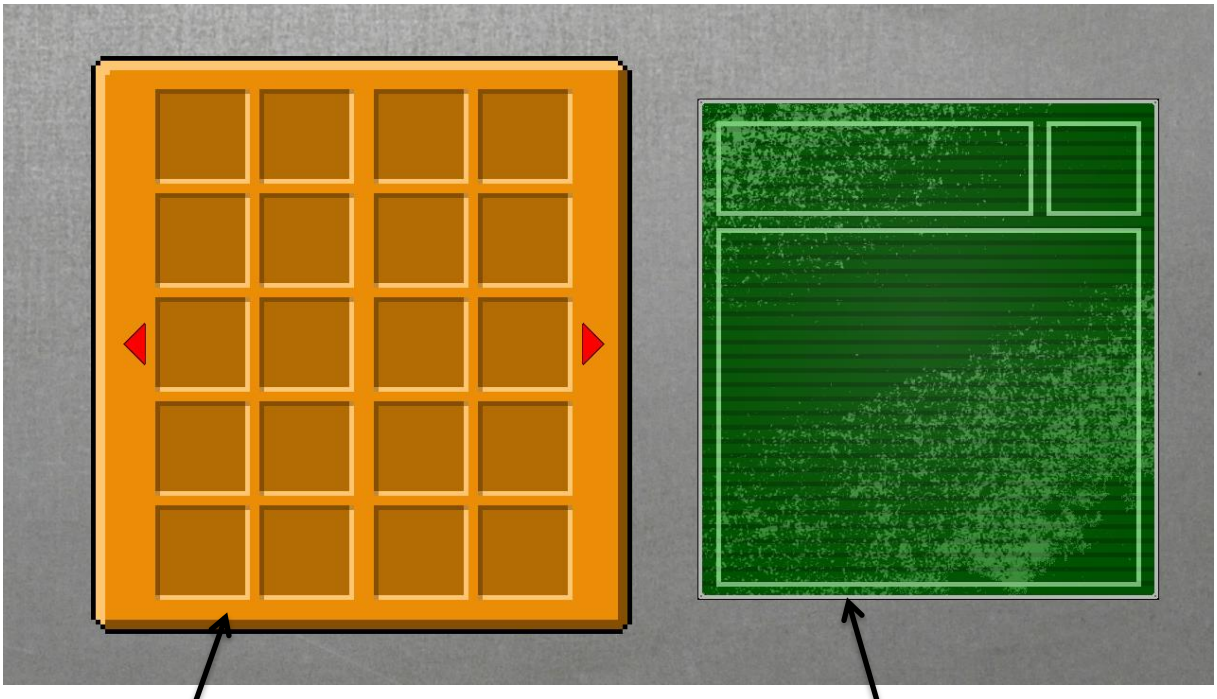
Une IA devra être créée pour les PNJ et sera capable d'exécuter les actions ci-dessus, il faudra donc créer une (ou plusieurs) ligne(s) de dialogue pour tous les PNJ important (principaux) et quelque une différentes pour les moins important, il faudra aussi implémenté une action spécifique pour chaque type de PNJ et ainsi classer ceux-ci, tous ça afin de pouvoir avoir une homogénéité dans la répartition des tâches pour chaque PNJ.

L'IHM :

Nous coderons plusieurs IHM spécifiques à cette partie afin de permettre au joueur toutes les actions listé ci-dessus, c'est-à-dire une IHM lorsque l'on se déplace dans les différentes zones du jeu (qui devra contenir : l'information sur la zone actuel, une mini map et un menu d'interaction nous permettant de switcher entre les différentes options/actions), une IHM sur les statistiques du personnage & les points de compétences à lui attribuer, un IHM sur l'inventaire (contiendra les différents objets acquis au cours du jeu ainsi que leurs information), et une IHM pour les options de jeu.

Ci-dessous ce trouve plusieurs maquettes d'IHM et d'objets pour cette partie :

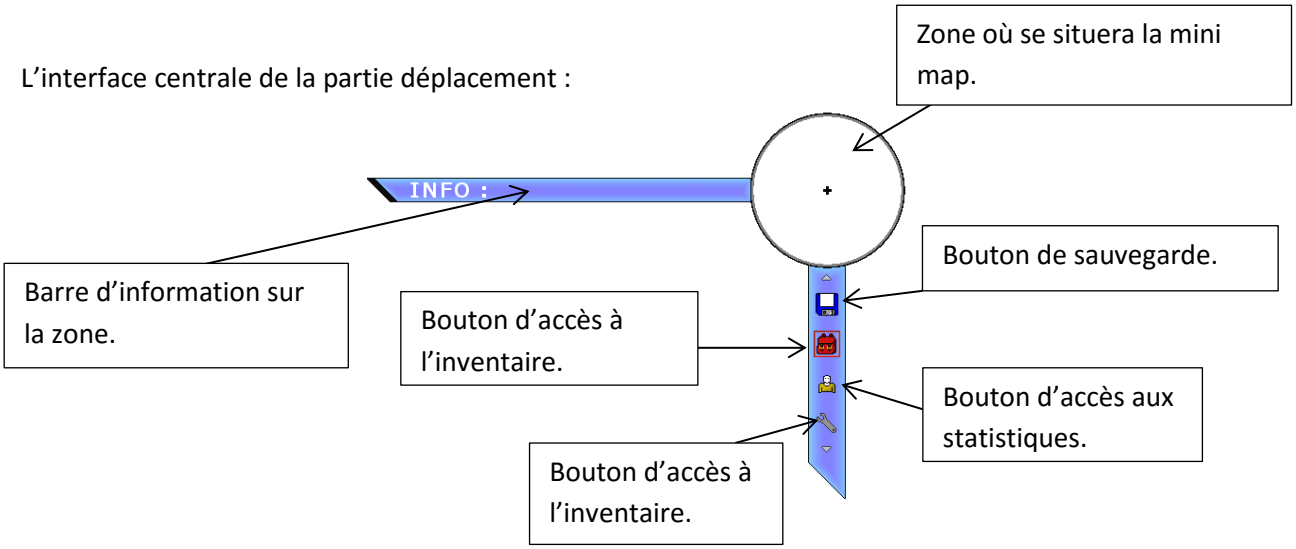
L'inventaire :



Ici ce trouverons les différent objets acquis au cours du jeu.

Ici ce trouve la page de description de l'objet sélectionnée.

L'interface centrale de la partie déplacement :



Différents objets présents dans le jeu :

La carte numérique (objet clé) : son obtention débloquera l'affichage de la mini map sur l'interface centrale.

Les potions de soins & de mana (consommable): permettra au personnage de récupérer un certain nombre de points de vie/mana.

Potion de vie

Potion de soin

Partie 3 : La partie Combat

(4mois 2 personne)

I. Mécanique de jeu

Il faudra dans un premier temps que le joueur puisse réaliser un certain nombre d'action (attaquer, se défendre, effectuer un combo, etc…) tous comme l'IA ce doit être capable de la même action (voir ci-dessous) et :

Pour ce qui est du déplacement en combat :

Le personnage joueur pourra déplacer son personnage selon un axe horizontal X uniquement, soit à gauche ou à droite. Comme n'importe quel jeu de combat qui se respecte, les espaces de combat selon limité dans la zone, à savoir que la zone de combat est prédéfini à l'avance et que le personnage joueur ne pourra dépasser cette limite.

L'IA (intelligence artificielle), dans le cas du jeu de combat, devra répondre à deux questions primordiales :

-Quelles sont les actions que je peux réaliser ?

-Quand dois-je réaliser les-dîtes actions ?

il faudra que la-dîte IA puisse déterminer à quel moment réaliser quelles actions, en d'autres termes s'adapter à la situation à laquelle elle sera soumise.

Par exemple, si elle reçoit un coup de poing, doit-elle parer ? Ou doit-elle plutôt esquiver ? Et ensuite, doit-elle contre-attaquer directement ? Et si oui, par quel coup ?

Cette dernière partie sera probablement la plus complexe à réaliser, et prendra donc plus de temps, en théorie.

Une IHM :
Cette IHM permettra d'afficher les points de vie du personnage et de son adverse ainsi que l'état de leurs jauges de mana, de guard et le timer, contrairement à la partie précédentes les joueurs n'interagira pas directement dessus (bouton de menu par exemple), mais indirectement par l'utilisation de coups spéciaux et/ou de dégât donné/reçus par les combattants.

II. A réaliser

En d'autres termes, le développement de l'IA se fera en deux temps :

Tout d'abord, il faudra que **l'IA puisse effectuer tous les mouvements** que peut réaliser le personnage qui lui est affecté. Cela inclut la parade / garde, les coups de poings, les coups de pieds, les coups spéciaux, les roulades / esquives, les sauts (et attaques sautées), etc...

IA puisse déterminer quand réaliser tel action :

Puis, nous avons deux "solutions" pour coder la deuxième partie de cette IA : soit nous codons chaque personnage / chaque type de combattant, et donc il faut aussi coder chaque réaction pour chaque coup pour chaque personnage adverse, ce qui rallonge énormément le code mais nous offre une certaine sécurité vis à vis des potentiels bugs, soit nous codons une seule IA, en partant du principe que chaque personnage possède plus ou moins la même palette de coups, ce qui rend le code moins long et plus facile à réaliser, mais qui ne tient pas compte des différences de « Hitbox » par rapport aux divers personnages, et donc une IA moins développée.

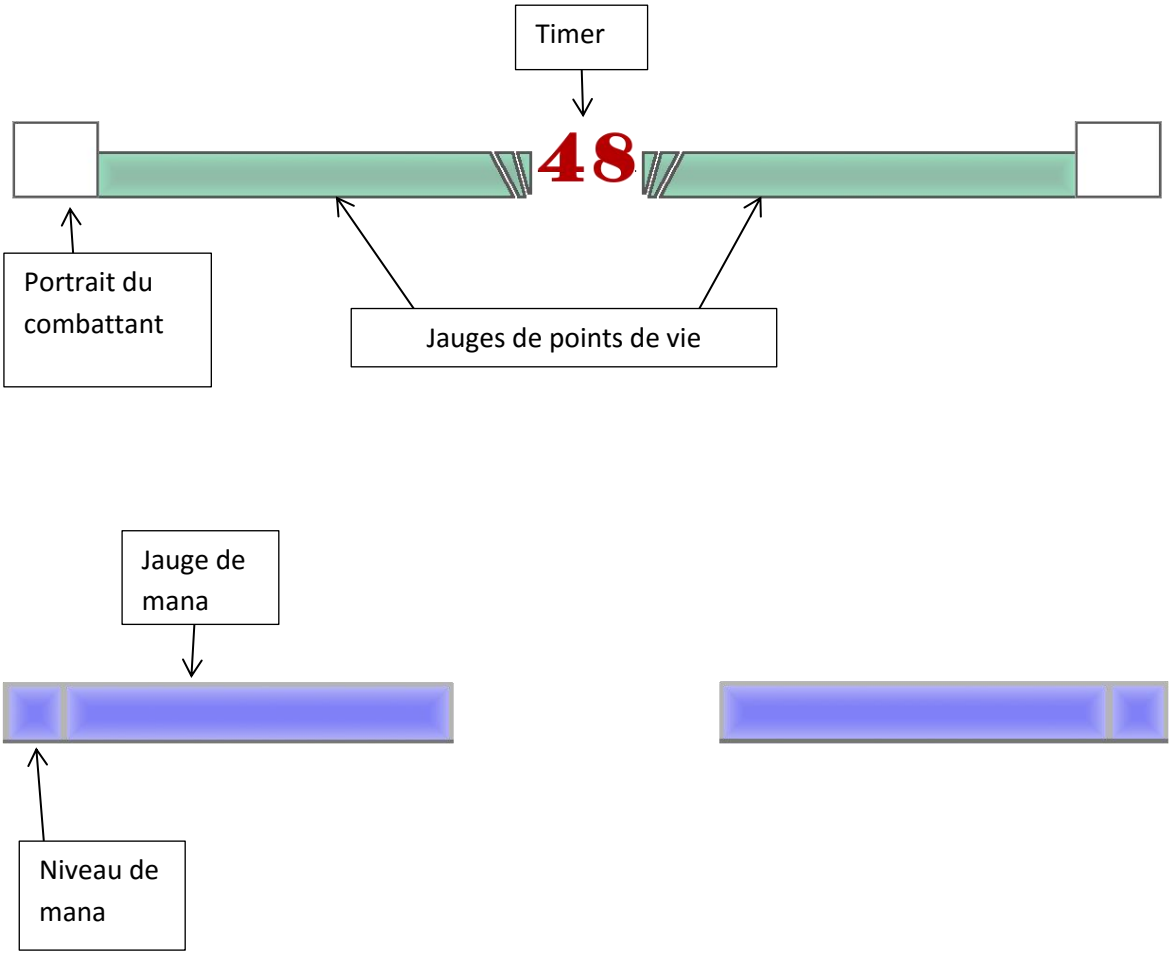
Au niveau des collisions, on essaiera de faire en sorte que le personnage joueur ne puisse pas se superposer avec l'ennemi, c'est-à-dire que le personnage joueur ne doit pas être en mesure d'avancer si son adversaire est devant lui, on devra donc gérer de nouvelles collisions, cela rejoint certainement l'idée de « Hitbox ».

L'IHM :

Ici il faudra la relier aux statistiques deux personnages combattants afin qu'elle puisse montrer leurs états physique (jauges de : vie, guard et mana) et donc qu'elle soit en symbiose avec eux. Il faudra aussi implémenter un chronomètre dont on affichera l'avancement.

Ci-dessous ce trouve une maquette de cette partie qui nous permet de voir l'agencement des différentes composantes de l'IHM :

(Cependant la jauge de guard n'est pas présente sur cette maquette car ajouté plus tard, mais elle se trouvera normalement sur le bas de la jauge de PV et sera assez petite):



Partie 4 : Logiciels & Méthodes utilisé

(45 jours d'apprentissage tout le monde)

I. Java & Unity :

Le Java, pourquoi ? :

Nous avons choisi de coder notre jeu avec le langage java, car outre le fait d'être l'un des langages les plus utilisé sur unity, il s'agit du seul langage orienté objet que nous ayons vus et appris à maitriser en cours.

Interface Unity :

Qu'est-ce qu'Unity en premier lieu ? Il s'agit d'un logiciel 3D/2D en temps réel utilisé pour la création de jeux, d'animation, d'objets 3D/2D. Il s'agit d'un freeware et peut-être programmable sous plusieurs langages informatique tel que C#, C++ et ou java.

Comment est constituée l'interface ?

Celle-ci est composée de 5 grandes sections :

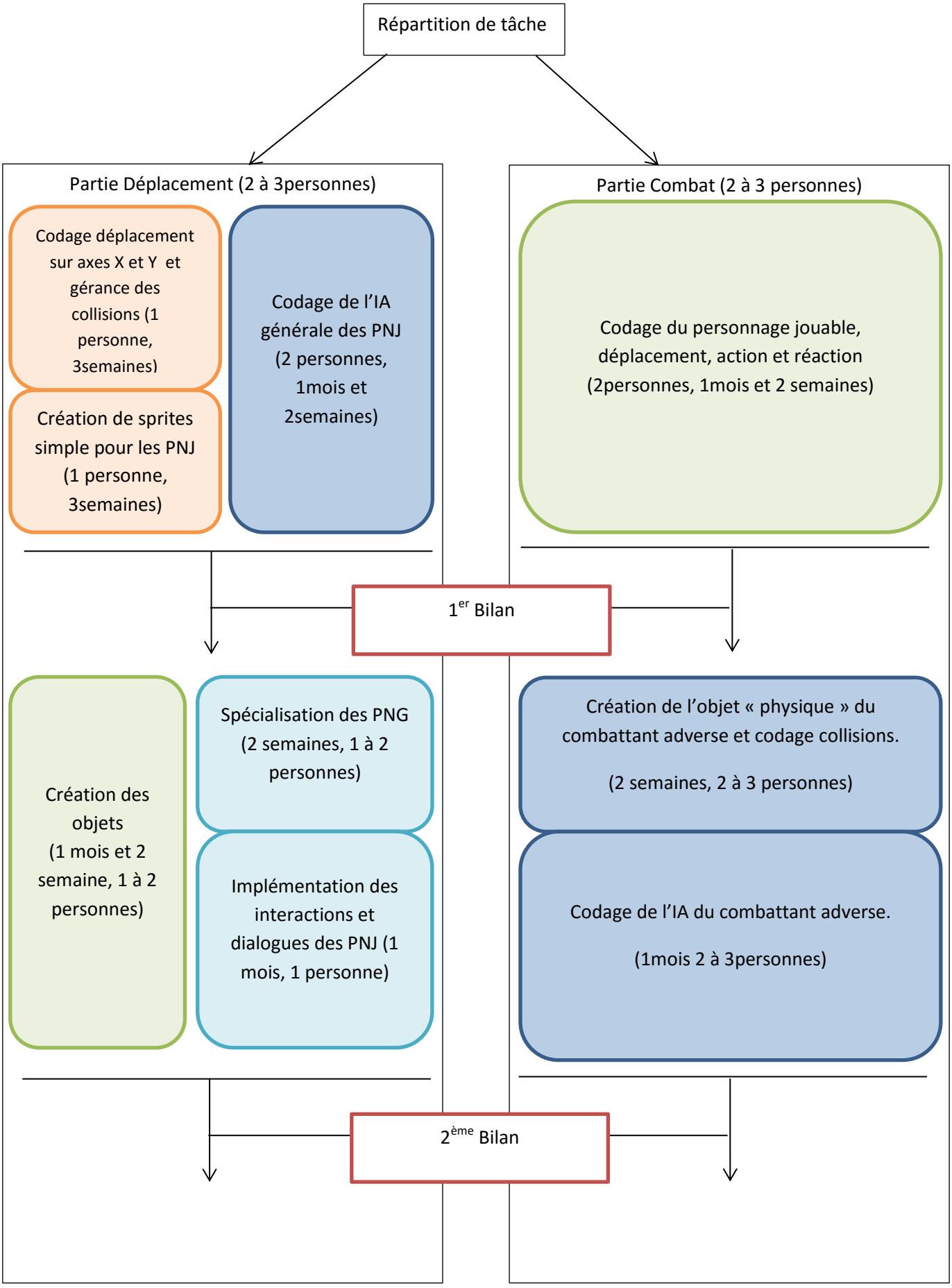
- La barre des taches (haut) : Basique avec sélection des grandes tâches.
- La scène (centre) : Sert de visuel au travail effectué et ce en direct.
- L'inspecteur (droite): Sert de modificateur principal d'option de rendement et d'ajustement.
- La zone de projet (centre bas): Regroupe tous(tes) les scènes / structures / objets / textures comporté dans le projet ouvert.
- La hiérarchie (bas à gauche) : Trie tous les types d'événements, d'objets en sous-parties nommable.

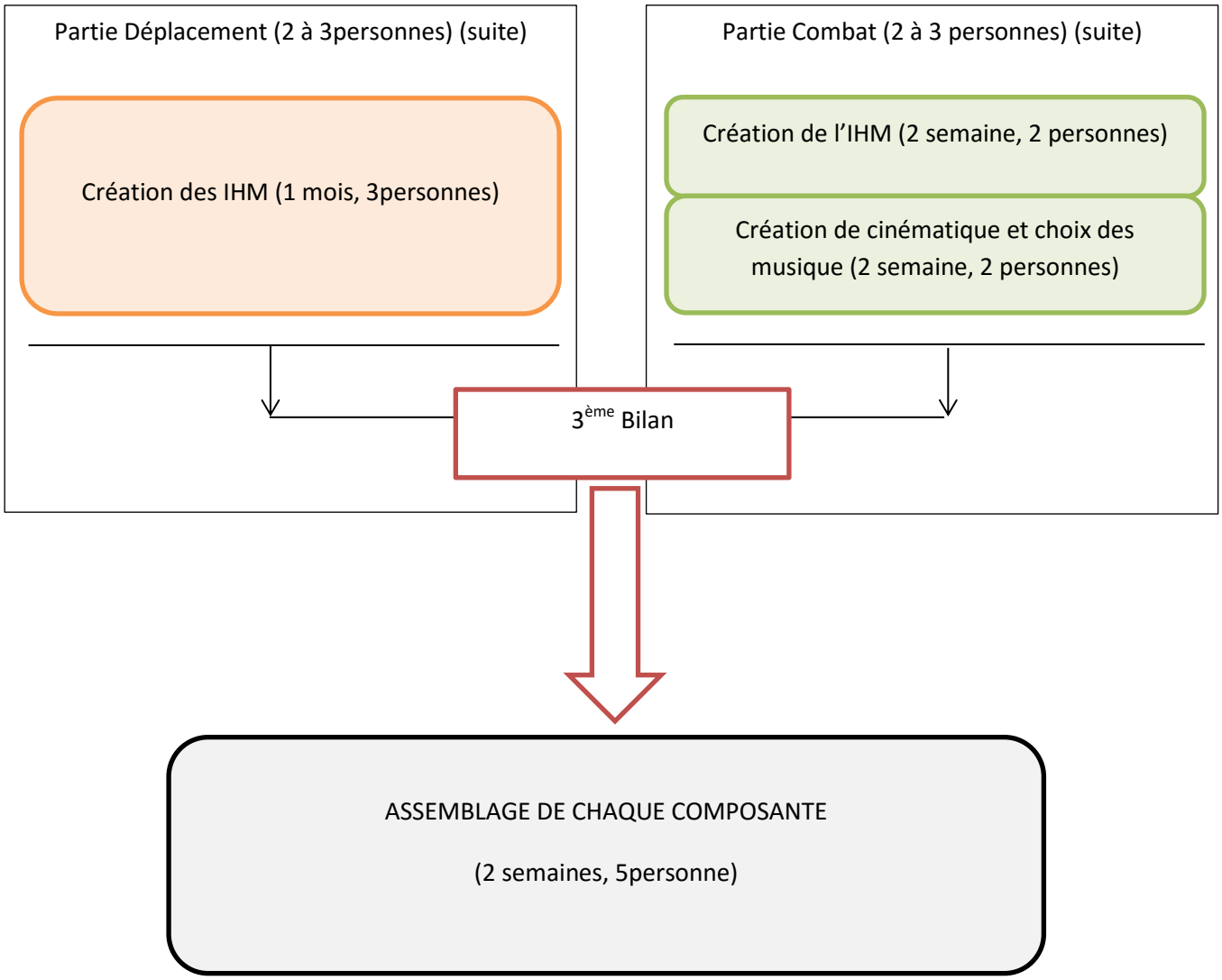
II. Les Tuiles

Les tuiles (ou *tile* en anglais) sont des éléments graphiques de dimensions similaires qui peuvent être placés dans un quadrillage afin de former un niveau. Chaque tuile peut disposer de caractéristiques différentes (solide / décor, fait des dégâts au héros,…) et d'une texture spécifique. Ce procédé est utilisé dans les jeux 2D, que la vue soit de haut (exemple : Zelda) ou de profil (exemple : Mario). Toutes les tuiles sont réunies dans une image que l'on appelle tileset. Un niveau peut être composé grâce à un tileset et un tableau de nombre : on associe un numéro à chaque tuile (ex : « 1 » pour le bloc ciel, « 2 » pour le bloc brique,…) et on place les tuiles dans le niveau grâce à leur numéro. La taille du tableau correspond à la taille du niveau. Les tuiles sont très utilisées du fait de leur simplicité d'utilisation.

Partie 5 : Organisation

Nous commencerons par mètre en commun nos connaissances et maîtrises pour voir quelle partie du travail siéra le mieux à chacun puis nous commencerons par répartir en 2 groupes modulables les différente partie, la réalisation du projet dans le temps ce fera comme ceci :


Répartition de tâche
Partie Déplacement (2 à 3personnes)
Codage déplacement sur axes X et Y et gérance des collisions (1 personne, 3semaines)
Création de sprites simple pour les PNJ (1 personne, 3semaines)
Codage de l'IA générale des PNJ (2 personnes, 1mois et 2semaines)
Partie Combat (2 à 3 personnes)
Codage du personnage jouable, déplacement, action et réaction (2personnes, 1mois et 2 semaines)
1er Bilan
Création des objets (1 mois et 2 semaine, 1 à 2 personnes)
Spécialisation des PNG (2 semaines, 1 à 2 personnes)
Implémentation des interactions et dialogues des PNJ (1 mois, 1 personne)
Création de l'objet « physique » du combattant adverse et codage collisions.
(2 semaines, 2 à 3 personnes)
Codage de l'IA du combattant adverse.
(1mois 2 à 3personnes)
2ème Bilan


Partie Déplacement (2 à 3personnes) (suite)
Création des IHM (1 mois, 3personnes)
Partie Combat (2 à 3 personnes) (suite)
Création de l'IHM (2 semaine, 2 personnes)
Création de cinématique et choix des musique (2 semaine, 2 personnes)
3ème Bilan
ASSEMBLAGE DE CHAQUE COMPOSANTE
(2 semaines, 5personne)