Projet Logiciel Transversal

Anand Candassamy & Paul Estano

Table des matières

1 Présentation Générale

1.1 Archétype

Notre jeu s'inspirera principalement du jeu *Pokemon Donjon Mystère*. En effet, nous avons prévu de conserver le mécanisme des combats et de donjon de ce jeu.
Dans notre logiciel l'utilisateur incarnera un pokemon dans les salles d'un donjon qui contiennent chacune des pokemons qui peuvent l'"agresser".
Pour simplifier le jeu nous abandonnons également d'évolution des pokemons.

1.2 Règles du jeu

Le donjon contient un nombre fini de salle et le joueur gagne lorsqu'il sort de la dernière salle du donjon.
Lorsque l'utilisateur est provoqué en duel automatiquement par les pokémons qui sont autour de lui.
Les combats fonctionnent en tour par tour. A chaque tour, l'utilisateur peut effectuer qu'une seule action.

- attaquer
- soigner le pokemon
- se déplacer

1.3 Ressources

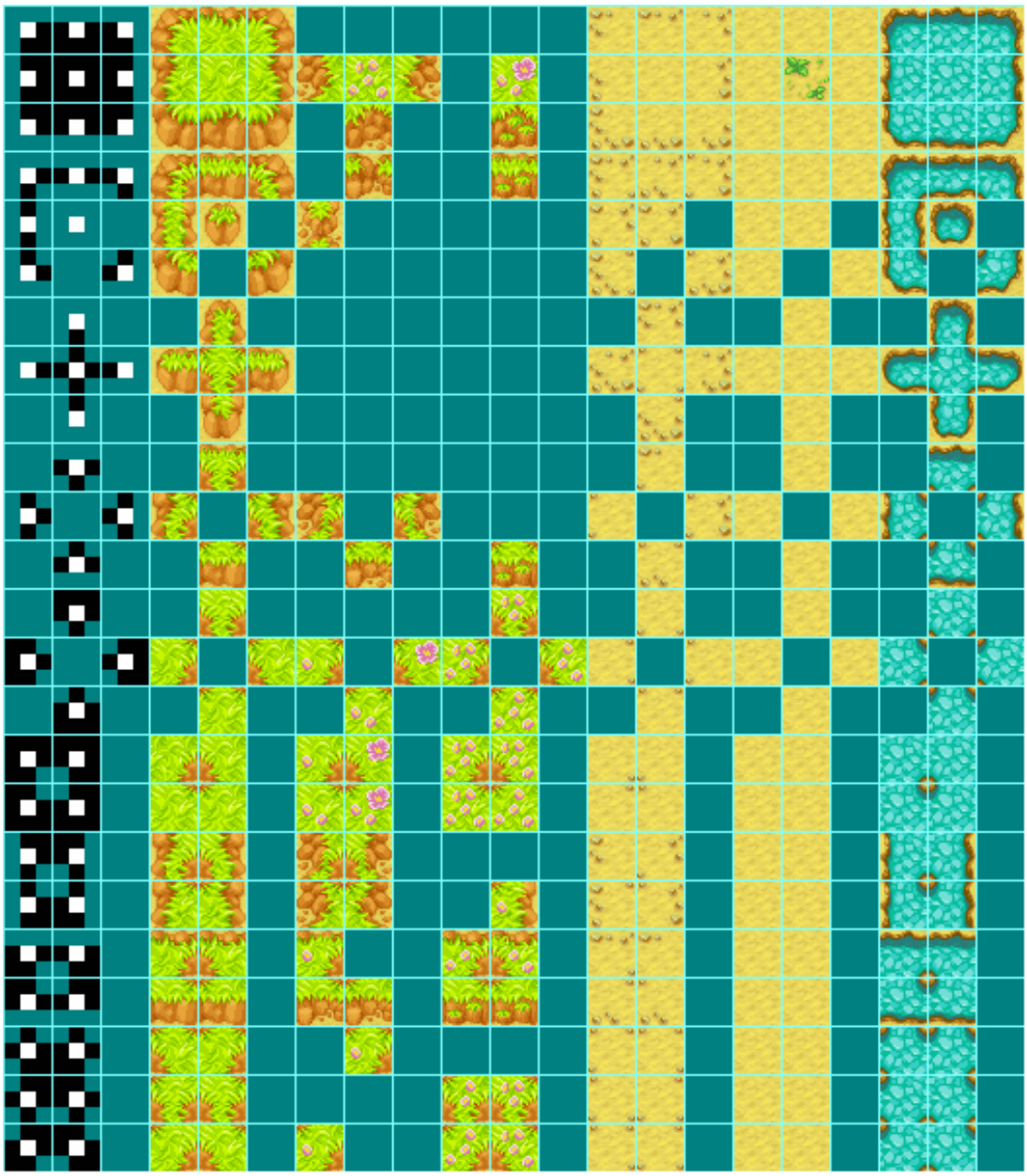

FIGURE 1 – Tileset utilisé pour construire un monde

FIGURE 2 – Tileset utilisé pour les pokémons

2 Description et conception des états

2.1 Description des états

Un état du jeu est formé d'une carte qui est statique au cours du temps et d'une liste de joueurs dont l'état varie au cours du jeu.

2.1.1 Etat de la carte

Une carte est composée de :
— plusieurs couches
— d'une longueur en nombre de tiles
— d'une largeur en nombre de tiles
— de la largeur de ses tiles en pixels
— de la longueur de ses tiles en pixels
— d'un tileset

Une tile correspond à un motif de pixels utilisé sur une ou plusieurs cases de la carte par une ou plusieurs couches.

Chaque couche de la carte comporte :
— un nom
— une largeur en nombre de tiles
— une longueur en nombre de tiles
— une position correspondant à la position de sa première case
— une liste d'index correspondant aux tiles utilisé pour chaque case. Ces tiles doivent se trouver dans le tileset de la carte

Un tileset comprend : une chaîne de caractères correspondant au chemin de l'image du tileset et un identifiant.

2.1.2 État des joueurs

Un joueur est soit vivant, soit mort, soit un robot, soit un humain et il lui est attribué un pokemon en début de partie.

Un pokemon possède une position, un nombre de points de vie à l'état t, un nom et un ensemble de statistiques qui lui sont attribués en début de partie.

Ces statistiques correspondent aux attaques qu'il peut utiliser, à son nombre de point de vie en début de partie et son type (eau, feu, herbe).

2.2 Conception Logiciel

Le package état peut se diviser en trois sous-partie :
— Une partie gérant les personnages, en bleu
— Une partie gérant l'environnement, en rouge
— Une classe représentant l'état global du jeu, en vert

La classe *player* contient l'ensemble des éléments permettant de caractériser l'état d'un joueur. Chaque joueur est lié à un pokemon par une relation de composition : un pokémon ne peut pas exister sans joueur. Dans le cas où le pokémon est contrôlé par l'IA, l'IA est considérée comme un joueur.
Chaque pokémon possède une relation d'agrégation avec des statistiques qui lui sont attribués en début de partie.
La partie environnement est calqué sur le modèle proposé par les fichiers exporter par le logiciel *Tiled Map Editor*. Les objets les plus consommateurs en mémoire tels les *data* dans la classe *Layer* sont par ailleurs

stockés dans le tas pour éviter les copies entre les différentes classes.
L'état global contient un pointeur vers l'état de l'environnement et et une liste de pointeur de contenant l'état des différents joueurs de la parti

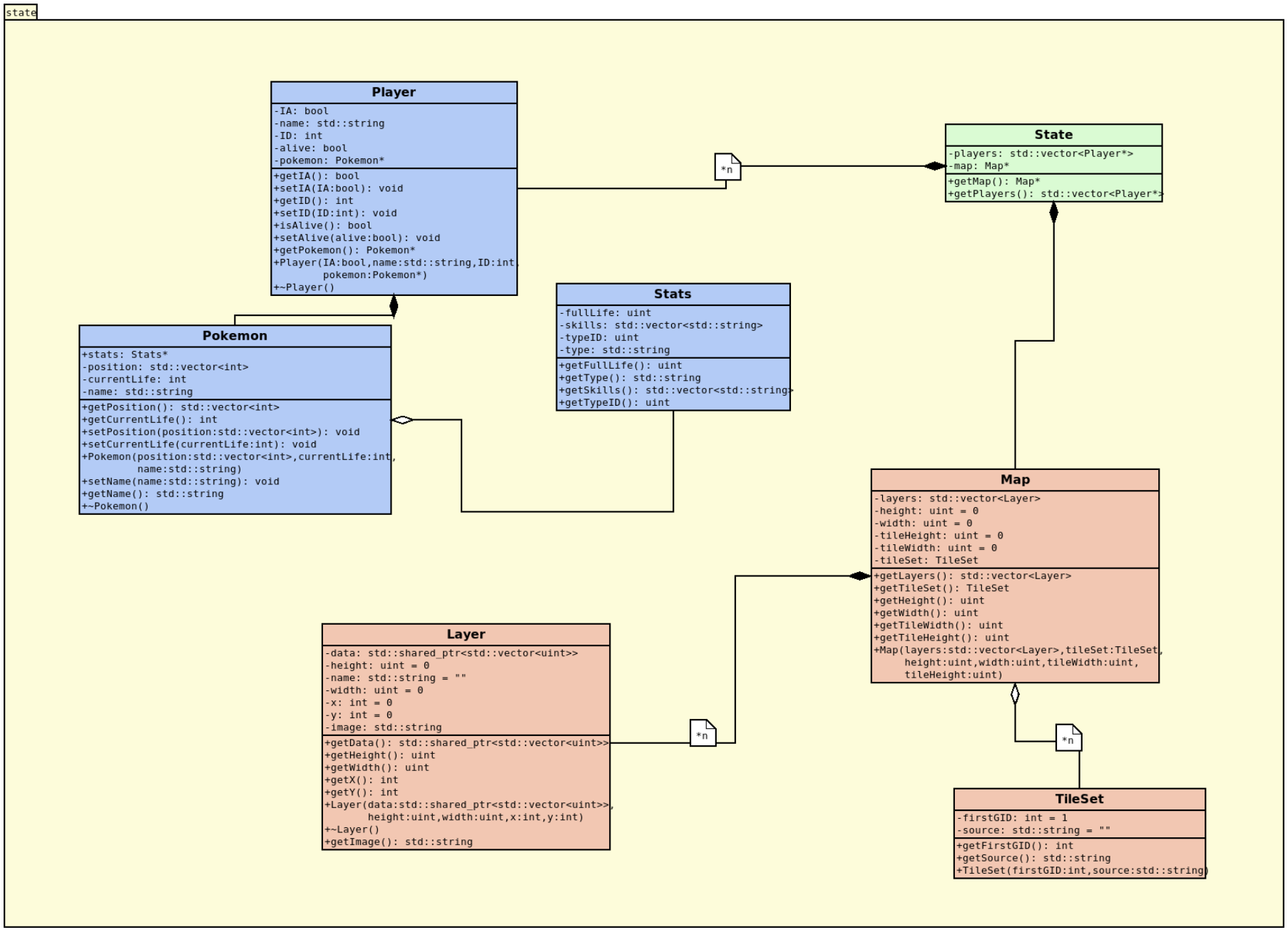

FIGURE 3 – Diagramme des classes d'état.

3 Rendu : Stratégie et Conception

3.1 Stratégie de rendu d'un état

3.2 Conception logiciel

4 Règles de changement d’états et moteur de jeu

4.1 Règles

4.2 Conception logiciel

5 Intelligence Artificielle

5.1 Stratégies

5.2 Conception logiciel

6 Modularisation

6.1 Organisation des modules

6.2 Conception logiciel