

Rapport de projet

- Incidence -

*CHAMBONNET Kevin
GAUTHIER Silvère
MARTINEZ Thierry
MOKHRETEAR Amin*

May 14, 2014

Contents

I Remerciements	4
II Cahier des charges	5
1 Introduction	5
2 Mecanismes de jeu	5
3 Structure du jeu	5
3.1 Moteur	5
3.1.1 Moteur multi-agent	5
3.1.2 Moteur de carte	5
3.2 Scripts	5
4 Elements graphiques	5
5 Elements sonores	5
III Gestion du projet	6
6 Gestion de l'équipe	6
7 Découpage en tâches	6
8 Assignation	6
9 Gestion du temps	7
10 Profil de risques	8
11 Choix technologiques	8
11.1 Langages de programmation	8
11.2 Bibliothèques	8
12 Gestion des fichiers	12
12.1 Format des Fichiers	12
12.1.1 Animation	12
12.1.2 Tileset	12
12.1.3 Carte	12
12.1.4 Sauvegarde	12
12.2 Commentaires	13
12.3 Convention de Nommage	13
12.4 Gestion du code source	13
IV Développement	14

13 Moteur de jeu	14
13.1 Gestion des états	14
13.2 Gestion de l'interface utilisateur	14
13.3 Gestion des ressources et animations	14
14 Moteur de carte	14
14.1 Les Tuiles	14
14.1.1 Les Sols	14
14.1.2 Les Eléments	14
14.2 Le Tileset	15
14.3 Météo	16
14.4 Incidences	16
14.4.1 Dilatation	16
14.4.2 Erosion	16
14.4.3 Aléatoire	16
15 Moteur multi-agent	28
15.1 Gestion des entités	28
15.1.1 Santé	28
15.1.2 Incidences	28
15.2 Gestion des scripts	28
15.3 Liste des scripts	28
16 Architecture	28
16.1 Machine à états	28
V Post-Mortem	29
17 Réalisations non abouties	29
18 Améliorations réalisables	29

Part I

Remerciements

Part II

Cahier des charges

- 1 Introduction**
- 2 Mecanismes de jeu**
- 3 Structure du jeu**
 - 3.1 Moteur**
 - 3.1.1 Moteur multi-agent**
 - 3.1.2 Moteur de carte**
 - 3.2 Scripts**
- 4 Elements graphiques**
- 5 Elements sonores**

Part III

Gestion du projet

6 Gestion de l'équipe

Tous les membres se connaissant et étant supposés être capable de travailler en équipe, nous n'avons fait aucune élection de chef de projet.

Nous avons opté pour travailler de manière collégiale, et ainsi garder une cohésion de groupe sans pour autant avoir de hiérarchie instaurée au sein du groupe, qui pourrait au contraire déservir la réalisation de nos objectifs.

Chaque membre a donc autant de pouvoir que les autres, et peut donc participer activement au projet, autant lors de la conception que du développement. Toutes les décisions seront prises suivant la majorité lors de votes.

Pour ce qui est des réunions de projets, nous avons convenu avec notre tuteur d'une réunion, allant d'environ trente minutes à une heure, toutes les une à deux semaines, afin de mettre au point l'avancement du projet. En parallèle, tous les membres de notre équipe se retrouvent une fois par semaine afin de discuter des points clés effectués ou à venir, donner lieu aux votes pour les prises de décisions, ou encore, lors de la phase de développement, travailler en collaboration afin d'optimiser notre travail.

Au niveau du travail collaboratif, nous avons mis en place un dépôt sur github, contenant tant la documentation telle que ce rapport que les sources de notre jeu. Par ailleurs, nous mettrons sur ce dépôt uniquement les fichiers sources, les images et les sons, mais en aucun cas les fichiers temporaires ou les exécutables. Un fichier "makefile" sera disponible pour quiconque voudrait compiler le programme chez lui. Les seuls fichiers binaires disponibles seront les PDF de la documentation, pour un soucis de facilité d'accès.

7 Découpage en tâches

Afin de préparer le développement du jeu, il était nécessaire de séparer les fonctionnalités les unes des autres. Nous avons abouti à ce diagramme, qui résume notre choix de découpage :

8 Assignation

Le projet étant maintenant découpé en un certain nombre de modules, il ne restait plus qu'à assigner chaque tâche à un ou plusieurs membres de l'équipe. Nous nous sommes organisés comme ceci :

- Scripts de l'IA : MARTINEZ Thierry, MOKHRETAR Amin.
- Moteur :
 - Gestion de la carte : GAUTHIER Silvère.
 - Gestion des entités : CHAMBONNET Kevin.

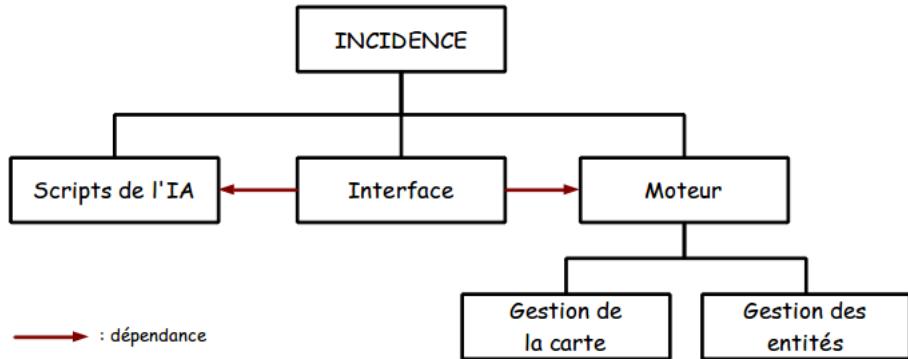


Figure 1: Découpage du projet en sous-tâches

- **Interface :** Tous les membres.

Bien entendu, cette répartition n'est pas totalement fixée, elle concerne en réalité l'affectation de responsables de parties, qui seront en charge de celle-ci mais pourront évidemment faire appel aux autres membres pour trouver une solution à un problème par exemple.

Le détail complet des tâches et assignations se situe dans la section Gestion du temps, page 7.

9 Gestion du temps

Afin de clarifier notre gestion du temps, un diagramme de Gantt est disponible dans la documentation de notre projet, et sera mis à jour en fonction de l'avancée du projet.

10 Profil de risques

Nature du risque	Degré du risque pour le projet
	0 1 2 3 4 5
Taille du projet	●
Difficulté technique	●
Degré d'intégration	●
Configuration organisationnelle	●
Changement	●
Instabilité de l'équipe de projet	●

11 Choix technologiques

11.1 Langages de programmation

Pour des besoins de performances, nous avons comparé différents langages. Pour réduire le temps de recherche et de comparaison, nous nous sommes appuyé sur des tests déjà effectués par d'autre.

Voici des tests de performances concernant un large panel de langages, comparés ici dans quatre contextes différents :

Nous pouvons observer que globalement, le langage le plus rapide est ici C++. L'utilisation de ce langage étant très fréquente dans les jeux vidéos, de part sa réputation d'un des langages les plus performants, et tous les membres de notre équipe sachant l'utiliser, nous avons fait le choix de programmer le moteur du jeu en C++.

Afin d'optimiser encore la rapidité du moteur, nous avons cherché à associer son coeur écrit en C++ avec un langage de scripting qui permettra de mettre en place les différentes actions du jeu.

D'après les graphiques ci-dessus, nous avons opté pour le langage LUA, performant et facile d'utilisation (syntaxe proche du C++). En effet, même si Python est très prisé et offre beaucoup plus de possibilités que LUA, nous l'avons estimé bien trop lourd pour l'utilisation que nous allons en faire.

Les deux langages C++ et LUA sont souvent associés dans les jeux vidéos, notre choix suit donc la tendance, ce qui nous offre une certaine assurance.

11.2 Bibliothèques

Pour la gestion graphique des tuiles composant la carte et des différentes entités, nous avons cherché une bibliothèque relativement simple d'utilisation mais surtout performante afin de garder la fluidité gagnée avec le choix des langages de programmation.

Connaissant la bibliothèque OpenGL, qui est bas niveau et performante dans les affichages deux et trois dimensions, nous nous sommes tournés vers deux bibliothèques utilisant OpenGL : SDL et SFML.

D'après plusieurs sites web et forums, les dernières versions (respectivement 2.0 et 2.1) de ces deux bibliothèques se valent en terme de performance.

En confrontant nos préférences personnelles quant au choix de l'une ou l'autre, nous nous sommes finalement mis d'accord pour utiliser la bibliothèque graphique SFML 2.1, qui paraît plus simple d'utilisation que la SDL. De plus, elle permet une gestion aisée des fichiers audio, ce qui sera un plus pour la

	Nom	Durée	Début	Nom de la Ressource
1	Etude Préalable	30 jours	20/01/14 08:00	CHAMBONNET Kevin;GAUTHIER ..
2	■Moteur	42 jours?	03/03/14 08:00	
3	■Gestion des Entités	42 jours?	03/03/14 08:00	
4	Moteur multiagent	20 jours?	03/03/14 08:00	
5	Implémenter les Entités	5 jours	14/04/14 08:00	CHAMBONNET Kevin
6	Placement des Entités	1 jour	21/04/14 08:00	MOKHRETAR Amin
7	Gestion des Maladies	2 jours	25/04/14 08:00	MARTINEZ Thierry
8	Incidences de la Santé	1 jour	29/04/14 08:00	MARTINEZ Thierry
9	■Gestion de la Carte	41 jours	03/03/14 08:00	
10	■Moteur de Carte	39 jours	03/03/14 08:00	
11	Conception	5 jours	03/03/14 08:00	GAUTHIER Silvère
12	■Implementation des Classes	26 jours	17/03/14 08:00	
13	Gestion du TileSet	4 jours	17/03/14 08:00	GAUTHIER Silvère
14	Sols et Elements	3 jours	21/03/14 08:00	GAUTHIER Silvère
15	Génération de Carte	3 jours	26/03/14 08:00	GAUTHIER Silvère
16	Optimisations Diverses	1 jour	15/04/14 08:00	GAUTHIER Silvère
17	Méthode "freePlace"	1 jour	21/04/14 08:00	GAUTHIER Silvère
18	Affichages	1 jour	14/04/14 08:00	GAUTHIER Silvère
19	Gestion des Bâtiments	2 jours	22/04/14 08:00	MOKHRETAR Amin
20	Affichage des Bâtiments	1 jour	24/04/14 08:00	MOKHRETAR Amin
21	Incidences sur le Territoire	3 jours	16/04/14 08:00	GAUTHIER Silvère
22	Incidences sur les Entités	2 jours	22/04/14 08:00	GAUTHIER Silvère
23	Sauvegarder / Charger	3 jours	24/04/14 08:00	GAUTHIER Silvère
24	■Cycle Jour/Nuit	4 jours	21/04/14 08:00	
25	Jour	1 jour	21/04/14 08:00	CHAMBONNET Kevin
26	Nuit	3 jours	22/04/14 08:00	CHAMBONNET Kevin
27	Météo	1 jour	21/04/14 08:00	MARTINEZ Thierry
28	■Points d'incidence	5 jours	22/04/14 08:00	
29	Ajout	3 jours	22/04/14 08:00	MARTINEZ Thierry
30	Suppression	2 jours	25/04/14 08:00	MOKHRETAR Amin
31	Sauvegarder / Charger	2 jours	25/04/14 08:00	CHAMBONNET Kevin
32	■Scripting	35 jours?	03/03/14 08:00	
33	Lister les Scripts	5 jours	03/03/14 08:00	MARTINEZ Thierry;MOKHRETAR ..
34	Etude préalable	5 jours?	10/03/14 08:00	MARTINEZ Thierry;MOKHRETAR ..
35	Concevoir les Scripts	25 jours?	17/03/14 08:00	MARTINEZ Thierry[50 %];MOKH..
36	Implémenter les Scripts	25 jours?	17/03/14 08:00	MARTINEZ Thierry[50 %];MOKH..
37	■Interface Graphique	12 jours	01/05/14 08:00	
38	Widgets	3 jours	01/05/14 08:00	CHAMBONNET Kevin
39	Menus	5 jours	05/05/14 08:00	MOKHRETAR Amin
40	Gestion Animations	3 jours	07/05/14 08:00	MARTINEZ Thierry
41	Gestion Sons	2 jours	01/05/14 08:00	MARTINEZ Thierry
42	Informations contextuelles	5 jours	12/05/14 08:00	CHAMBONNET Kevin;MARTINEZ ..
43	■Graphismes	45 jours	08/03/14 08:00	
44	Elements de Carte	5 jours	08/03/14 08:00	GAUTHIER Silvère
45	Entites	2 jours	29/04/14 08:00	GAUTHIER Silvère
46	Elements d'interface	7 jours	01/05/14 08:00	GAUTHIER Silvère
47	Equilibrage du Jeu	5 jours	19/05/14 08:00	CHAMBONNET Kevin;GAUTHIER ..

Figure 2: Diagramme de Gantt (page 1)

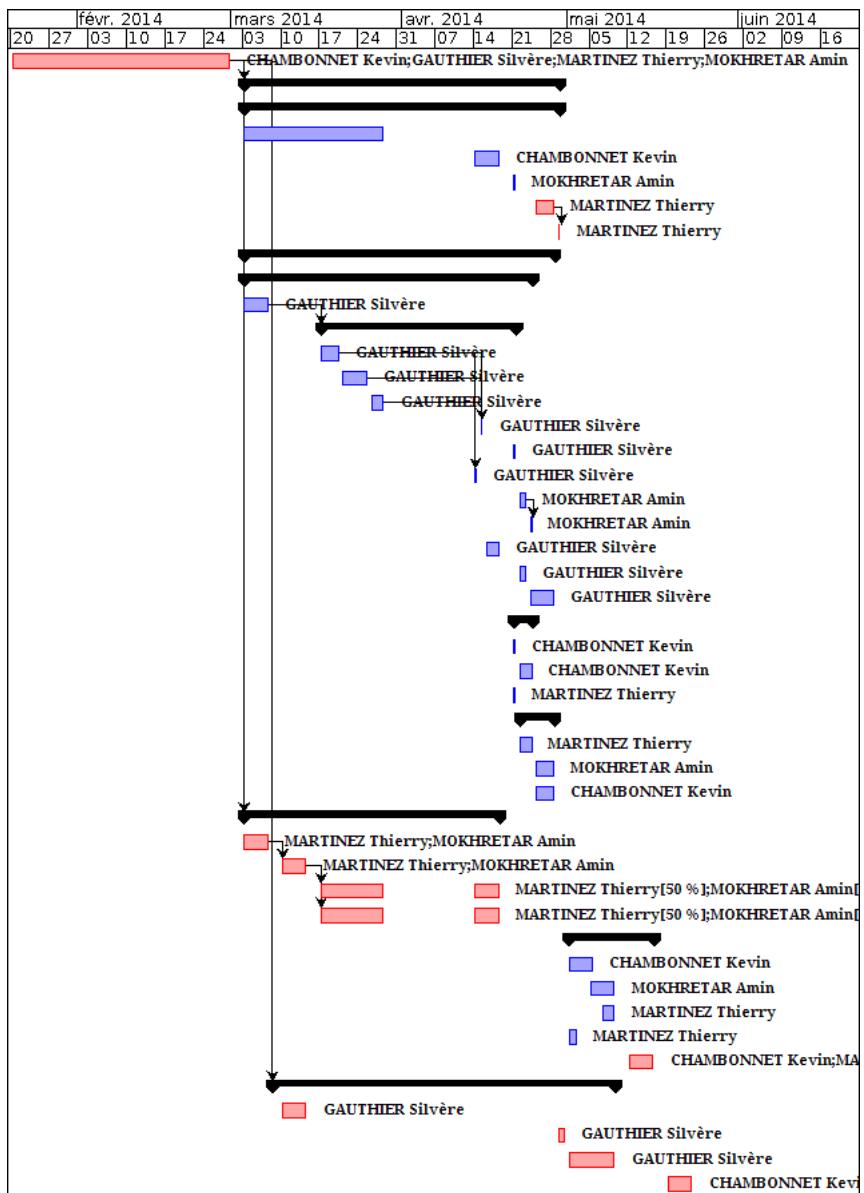


Figure 3: Diagramme de Gantt (page 2)

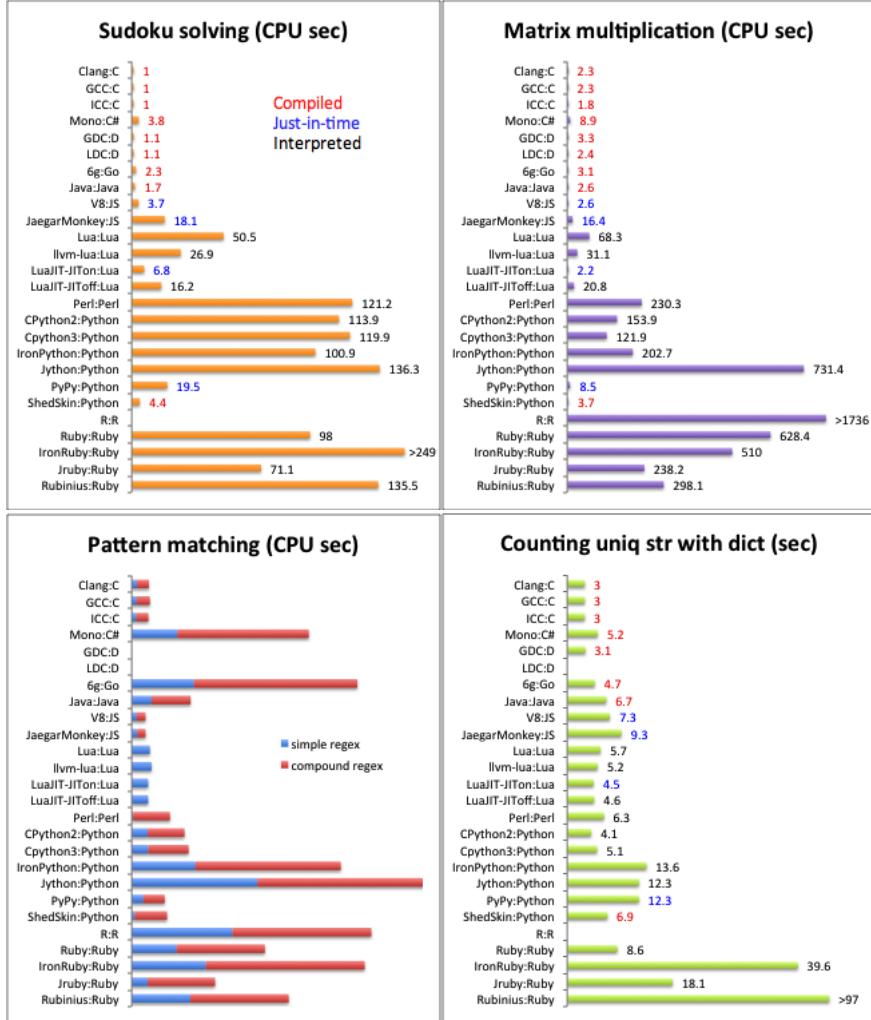


Figure 4: Comparaisons de performances de divers langages dans des cas donnés

finalité de notre jeu.

12 Gestion des fichiers

12.1 Format des Fichiers

Le moteur de jeu étant écrit en C++, nous utiliserons des fichiers d'en-tête au format HPP et des fichiers de définition au format CPP. Pour les scripts d'IA des entités, le langage étant Lua, ceux-ci seront au format LUA.

Toutes les images nécessaires au jeu seront au format PNG afin de pouvoir utiliser la transparence et garder la pleine qualité d'image (contrairement à JPEG qui perd de l'information à la compression).

Les sons quant à eux seront au format WAV afin d'éviter toute gestion de la compression de fichier pour de petits fichiers qui n'en ont aucunement besoin.

12.1.1 Animation

```
Chemin/vers/image.png size_x_sprite size_y_sprite play?(1/0) loop?(1/0)
+Frame position_x position_y temps_affichage
+Frame position_x position_y temps_affichage
+Frame position_x position_y temps_affichage
```

12.1.2 Tileset

Le tileset sera une unique image au format PNG représentant l'ensemble des tuiles possibles. Chaque type de sol constituera une ligne du tileset, avec toutes les variantes dépendant des rebords entre tuiles (ce fichier sera présenté plus en détails dans la section 14.2, page 15).

12.1.3 Carte

Les différentes cartes sauvegardées seront enregistrées dans des fichiers IMS (Incidence Map Save), dans lesquels seront stockés le tileset utilisé et la conformation de la carte (dimensions, sols et éléments).

12.1.4 Sauvegarde

Un fichier sera créé pour la sauvegarde du jeu et un autre pour la sauvegarde de la carte, cela permettant dans le futur de créer des cartes réutilisables, comme par exemple dans un éditeur de carte.

Le fichier de carte contiendra uniquement le lien vers le tileset ainsi que les types de sols et éléments présents sur la carte. Le fichier sera donc de la taille : $2 * \text{nombre_de_cases} * \text{sizeof(int)} + \text{sizeof(chemin_du_tileset)}$.

12.2 Commentaires

Si une méthode ou fonction, voir même un bloc, dépasse une certaine taille (environ 10 lignes) ou devient trop compliquée, un commentaire sera ajouté avant celle-ci expliquant brièvement son processus :

```
/** Description :  
 *** Entrée : ...  
 *** Sortie : ...  
 **/
```

Marqueur spécifique	Signification
TODO	A mettre à la place du code d'une fonctionnalité à implémenter
RECODE	A mettre au dessus du bloc d'une fonctionnalité à refaire
FIXME	A mettre au dessus du bloc d'une fonctionnalité contenant un bug

12.3 Convention de Nommage

Type de variable	Format du nom
Classe	Majuscule suivit de minuscules
Méthode et Fonction	Minuscules (pour les mots composés, chaque mots suivant est une majuscule suivit de minuscules)
Attribut de classe	m_-
Variable globale	g_-

12.4 Gestion du code source

Afin de faciliter le travail collaboratif, nous utilisons un dépôt utilisant le gestionnaire de version GIT, hébergé sur le site <https://github.com/>. Sur ce dépôt seront présents tous les fichiers sources nécessaires au développement du jeu ainsi que les documentations au format Latex et PDF (même si aucun fichier binaire ne devrait être présent, il est plus pratique de récupérer directement un tel fichier que de le compiler soit-même). De plus, y seront stockées toutes les données utilisées par le jeu telles que les images et les sons. Seuls les fichiers temporaires, exécutables et fichiers de sauvegarde ne seront pas stockés.

Part IV

Développement

13 Moteur de jeu

13.1 Gestion des états

13.2 Gestion de l'interface utilisateur

13.3 Gestion des ressources et animations

14 Moteur de carte

14.1 Les Tuiles

Les différents types de tuile seront tous recensés dans le tileset, qui se chargera de fournir les textures et attributs associés à ceux-ci. La carte n'utilisera que des pointeurs vers ces instances constantes de tuile, ce qui évitera le stockage inutile d'instances identiques.

14.1.1 Les Sols

Chaque sol aura différents attributs définis dans le fichier de configuration du tileset (cf section 14.2, page 15). Comme définis dans le cachier des charges, certains sols seront incompatibles avec d'autres.

Attribut	Description
Type	Entier définissant un identificateur du type de sol.
Coût	Entier définissant le coût en PI de la pose du sol.
Nom	Chaîne de caractères utilisées lors des affichages dans l'interface.
Comportement	Enumération de trois comportements : DEFAULT : comportement par défaut (tous les type de Terre). FLUID : les sols forment des zones (étendues d'eau). CLIFF : les sols sont fixes et forment des zones (falaises).
Franchissable	Booléen indiquant si le sol est franchissable ou non par les unités.
Rebords	Tableaux d'entiers indiquant tous les types de sol compatibles avec les bordures de l'image (cf section 14.2, page 15).

14.1.2 Les Eléments

Chaque sol aura différents attributs définis dans le fichier de configuration du tileset (cf section 14.2, page 15).

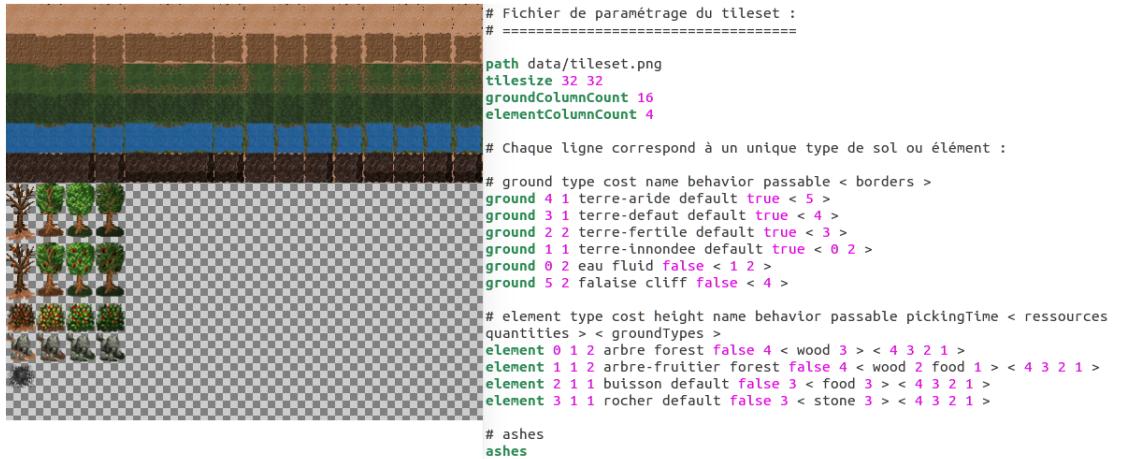


Figure 5: Texture et configuration du tileset

Attribut	Description
Type	Entier définissant un identificateur du type d'élément.
Type de sol	Entier définissant un identificateur du type de sol sur lequel se place l'élément.
Coût	Entier définissant le coût en PI de la pose de l'élément.
Nom	Chaîne de caractères utilisées lors des affichages dans l'interface.
Comportement	Enumération de deux comportements : DEFAULT : comportement par défaut. FOREST : les éléments forment des zones (arbres).
Franchissable	Booléen indiquant si le sol est franchissable ou non par les unités.
Temps de récolte	Valeur de temps durant laquelle le citoyen doit agir afin de récolter les ressources.
Ressources	Ressources (et leur quantité) associées à l'élément.

14.2 Le Tileset

Un unique fichier image au format PNG contiendra l'ensemble des tuiles utilisées dans le jeu pour les sols ou éléments. Cette image sera utilisée comme texture et certaines parties seront extraites par le programme au gré des besoins.

Dans le but de permettre à l'utilisateur d'aisément créer des mods de jeu ou changer de tileset, nous avons créé un fichier au format INI détaillant toutes les particularités du tileset.

Voici le tileset que nous utiliserons :

Fichier de configuration :

Chaque ligne commençant par "#" ne sera pas lue, ce sont les commentaires. Les mots en vert sont les mots clés utilisés pour la lecture qui se fait ligne par ligne :

- path : indique le chemin du fichier de texture du tileset
- tileSize : est suivi des dimensions (en pixel, largeur puis hauteur) d'une tuile
- groundColumnCount : nombre de colonnes pour chaque sol dans le fichier de texture (toutes les associations de rebords)
- elementColumnCount : nombre de colonnes pour chaque élément dans le fichier

de texture (tous les types de sols)

- ground : définit un sol et tous ses attributs
- element : définit un élément et tous ses attributs (height est la hauteur en nombre de cases dans la texture)
- ashes : fourni une texture pour les cendres (lorsque des éléments brûlent) (cf GDD)

Remarque : chaque ligne commençant par ground, element ou ashes incrémente un compteur afin de connaître la position de la zone de texture correspondante. L'ordre est donc important (définition des lignes de haut en bas).

14.3 Météo

14.4 Incidences

14.4.1 Dilatation

Les incidences basées sur la dilatation des zones prennent en compte les types de sols ou éléments afin d'élargir les étendues voulues en propageant les sols si besoin afin de garder les contraintes définies dans le GDD.

Voici quelques présentations des résultats de ces dilatations :

14.4.2 Erosion

Les incidences basées sur l'érosion des zones prennent en compte les types de sols ou éléments alentours afin de choisir les types en propageant les sols si besoin afin de garder les contraintes définies dans le GDD.

Voici quelques présentations des résultats de ces érosions :

14.4.3 Aléatoire

Les dernières incidences concernant le territoire sont basées sur un choix aléatoire de cases. Il s'agit ici d'apparition ou disparition de certaines ressources.

Voici quelques résultats de ces incidences particulières :

15 Moteur multi-agent

15.1 Gestion des entités

15.1.1 Santé

15.1.2 Incidences

15.2 Gestion des scripts

15.3 Liste des scripts

16 Architecture

16.1 Machine à états

pour chaque etat : interface + graphiques + sons + action possibles

Part V

Post-Mortem

17 Réalisations non abouties

18 Améliorations réalisables

Part VI

Annexes



Figure 6: Dilatation des étendues de fluides



Figure 7: Dilatation des zones humides



Figure 8: Dilatation des zones sèches



Figure 9: Dilatation des forêts



Figure 10: Erosion des étendues de fluides



Figure 11: Erosion des zones humides



Figure 12: Erosion des zones sèches



Figure 13: Erosion des forêts



Figure 14: Apparition de tout type de ressource



Figure 15: Disparition par le feu des ressources contenant du bois



Figure 16: Apparition de rochers uniquement