Hexagon War

1. Introduction

Description

Plusieurs joueurs (jusqu'à 8) s'affrontent sur un terrain de type grille hexagonal. Le but est de conquérir le plus grand territoire et ainsi anéantir l'ennemi.

Chaque joueur joue sur son navigateur et la communication s'effectue en temps réel avec un serveur Node.js et les autres joueurs

Cahier des charges

Répartition du travail

Client : Matthieu Constant
Serveur : Ludovic Gindre

Technologies choisies

Client : HTML 5 avec le canvas, jQuery et socket.io
Serveur : node.js, express.js, socket.io

Réalisation individuelle

Matthieu Constant

Buts

Le but de ma partie était de réaliser la partie client du jeu.

Travail réalisé

J'ai commencé par lire une explication sur les hexagones sur cet excellent site interactif : http://www.redblobgames.com/grids/hexagons/

J'ai ensuite réalisé l'affichage du plateau, soit dessiner des hexagones selon l'orientation et la taille voulue avec la possibilité de choisir la couleur de remplissage et d'y écrire du texte si souhaité, pouvoir se déplacer sur le terrain si celui est trop grand pour s'afficher entièrement à l'écran à l'aide des touches et de la souris (gestion des touches et de la souris), la détection de l'hexagone dans lequel se trouve la souris (conversion de pixel à un index) ainsi que l'affichage du score.

J'ai terminé la partie client en ajoutant quelques fonctions logiques comme par exemple vérifier que si l'on clique sur une case au moins un de ses voisin est du même type et en ajoutant la connexion avec le serveur et l'envoie et la réception des données

Questions :

Qu'est-ce qui vous a positivement surpris ?

L'explication des hexagones et la simplicité de socket.io

Qu'est-ce qui vous a pris plus de temps ?

La compréhension des hexagones.

Quel est le pire bug que vous avez eu ?

Le redimensionnement de la fenêtre qui ne permet plus de détecter correctement l'hexagone dans lequel se trouve la souris. Ce bug est d'ailleurs toujours présent. Je pense qu'il faudrait regarder du côté de la translation pour pouvoir régler se problème.

Au final si vous deviez recommencer que changeriez-vous ?

J'utiliserai une autre manière de stocker la carte (probablement axiale au lieu de d'un simple numérotation ligne/colonne actuellement). Ceci simplifierai le code en évitant de faire des conversions pour le calcul des voisins.

Ludovic Gindre

Buts

Le but de ma partie était de réaliser le serveur du jeu avec Node.js. Il devait être capable d'accueillir des joueurs, de gérer le plateau de jeu et de communiquer avec les clients les différents changements d'état du jeu.

Travail réalisé

Dans un premier temps, j'ai surtout lu de la documentation et des tutoriels sur Node.js. Après toute cette lecture, j'ai pu faire un serveur simple qui retournait une page en fonction de l'url qui lui est demandé. Ensuite, il fallait faire un moteur basique pour le jeu. Il était capable de gérer le plateau de jeu et les nouveaux utilisateurs. Enfin, avec le moteur de jeu terminé, il fallait que le serveur puisse communiquer avec les clients pour leurs transmettre les mises à jour du jeu. Pour cela, il aura fallu repasser par la case lecture pour en apprendre plus sur socket.io. Cet outil étant relativement facile à prendre en main, il a été facile de mettre en place la communication en temps réel. Enfin il a fallu corriger les quelques bugs restants.

Questions :

Qu'est-ce qui vous a positivement surpris ?

La grande quantité de code disponible pour node.js grâce à NPM qui est également un excellent outil.

Qu'est-ce qui vous a pris plus de temps ?

Comprendre le fonctionnement de Node.js

Quel est le pire bug que vous avez eu ?

La déconnexion. Dans un premier temps les joueurs étaient simplement stockés en fin de tableau des joueurs. Le fait de pouvoir déconnecter les utilisateurs posait problème au niveau des index avec certains index indéfinis. Ceci a entrainé une refonte quasiment complète de la gestion des utilisateurs.

Au final si vous deviez recommencer que changeriez-vous ?

Je perdrais moins de temps à apprendre de nouvelles choses sur node.js qui ne sont pas utiles pour le projet. Je me suis un peu emporté dans la lecture et dans la découverte de nouvelles choses. Enfin je penserais mieux à la gestion des utilisateurs qui a dû être presque totalement réécrite.

1. Conclusion

Description du travail dans l'état actuel (avec les problèmes connus)

Le jeu fonctionne dans son état le plus simple. Ce qui veut dire que les joueurs peuvent se connecter acheter des territoires et se déconnecter.

Proposition d'amélioration

Ajouter des territoires qui ajoutent des bonus comme un cout plus élevé sur les cases possédées et des cases inatteignables comme des montagnes ou des plans d'eau.