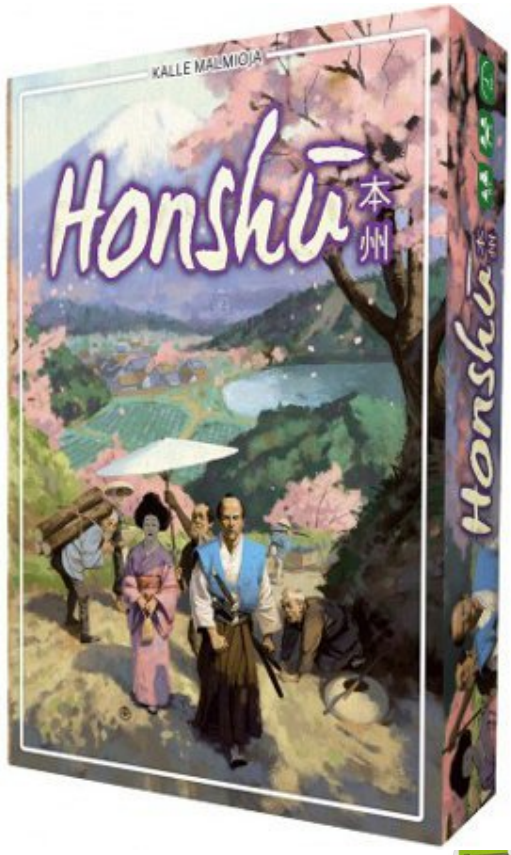
Equipe STRANGER C
Jiahui XU, Kevin XU, Heykel Rajhi, Benoît SCHOLL

Projet Honshu

Rapport LOT D



Sommaire

Introduction

Tâche à effectuer

Nous avons choisi pour le lot D d'implémenter une interface graphique de notre jeu. On a alors dû apprendre à utiliser les bibliothèques SDL.
Voici alors la liste des tâches à effectuer :

- Création d'un nouveau menu graphique
- Définition de notre Interface
- Sélection de tuile sur l'interface graphique
- Poser une tuile sur la grille
- Implémentation d'un nouveau solveur

Répartition des tâches

Kevin s'est occupé du nouveau solveur et aussi de quelques parties de la SDL (identification position souris).
Benoît s'est occupé de la partie SDL de façon globale.
Jiahui a écrit des fonctions de test et de dessin pour la partie SDL (affiche Deck Plein écran).
Heykel s'est occupé de la fin de partie code et partie SDL.

1. Interface graphique

Pour réaliser l'interface graphique, il a fallu implémenter des nouvelles fonctions d'affichage pour la grille et les tuiles. Il a aussi fallu créer nos fichiers images utilisés pour le jeu. Nous avons pu compter sur l'aide de Mlle Jennyfer Regnier, artiste de la région stéphanoise pour le dessin de nos tuiles.

1.1 Menu

Dans la première version de notre menu, l'utilisateur ne pouvait que lancer une partie ou quitter le jeu. La partie ainsi lancé était donc statique : on ne pouvait pas changer la taille du deck ou la taille de la grille. Cela était en partie dû à notre damier de jeux, qui était un fichier image de la totalité de ce dernier. Cette solution ne nous convenait pas, et nous avons donc décidé de dessiner la grille, case par case, afin de pouvoir rendre sa taille dynamique. En ce qui concerne le Deck, Kevin a travaillé sur des fonctions de génération de deck aléatoire (avec comme contrainte qu'une tuile ne peut être présente dans le deck plus d'une fois). Nous avons quand même dû limiter la taille de la grille à une borne min (5) et max (15) afin qu'elle ne dépasse pas de l'écran. Tout ceci nous a amené à créer le deuxième écran du menu.

1.2 Affichage de la grille et des tuiles

Nous avons d'abord créé des fonctions simples, une qui dessine une grille vide, puis une qui dessine une tuile. Puis à l'aide de ses fonctions, et d'autres qui sont venues se rajouté, l'interface de notre version de Honshu à commencé à prendre forme.

1.3 Mode sélection de tuile

La sélection de tuile se fait à partir du deck. Il se compose d'un tas de départ, d'une tuile en cours et d'un tas de fin. Deux flèches se situes de part et d'autre du deck afin de pouvoir naviguer dedans.

Voici les touches que le joueur peut utiliser :

- h : utilisation du solveur (3 utilisations mx possible)
- d : affichage du deck en entier
- Flèches droite ou gauche pour sélectionner une tuile dans le deck
- Entrer : sélection de la tuile en afficher

Sinon, le joueur peut aussi cliquer directement sur le deck afin de sélectionner la bonne tuile, puis cliquer sur la tuile pour la sélectionner

1.4 Mode poser une tuile

Après que le joueur ait sélectionné une tuile, il peut déplacer la tuile sur la grille avec les flèches directionnelles. Il peut aussi effectuer une rotation de 90° en appuyant sur la touche o.

A l'aide de la souris, le simple fait de survoler la grille positionne la tuile aux coordonnées les plus proche de celle de la souris, et une vérification de « potabilité » est immédiatement lancée.

Si la position de la tuile répond au règle et satisfait le joueur, il pourra valider en appuyant sur entré ou en cliquant à l'aide du bouton gauche de sa souris.

Afin d'aider le joueur, nous avons aussi ajouter un panneau d'information qui indique si une tuile est posable à l'endroit où elle est provisoirement. En effet, elle indique précisément les règles qui sont ou ne sont pas satisfaites.

La rotation de la tuile nous a poser un petit problème. Si on décide de tourner la tuile trop proche des limites de la grille, on est obligé de décaler celle-ci pour la faire rentré (et éviter un segmentation faut ☺).

1.5 Fin de partie

A chaque tour de boucle, on vérifie si l'on a atteint la fin de partie (plus de tuiles dans le deck, ou plus de tuile posable). Si la fin de partie est déclarée, un écran indiquant le score finale est alors affiché.

1.6 Problèmes rencontrés

Dans la partie SDL, chaque nouveau pas que l'on faisait nous apportait son lot de difficultés. Les plus difficile à traiter ont été la gestion de la souris. Alors que beaucoup semble dire que la gestion à la souris est plus évidente à programmer que la gestion au clavier, ce ne fut pas notre cas. Nous avons même failli abandonner la souris pour n'avoir qu'une version jouable au clavier.

Finalement, nous avons avancé par étape simple. Tout d'abord, nous avons réussi à récupérer l'évènement clic de la souris. Cela peut paraître trivial, mais nous avons eu un bug (que nous n'expliquons toujours pas) entre le pc et le mac, Benoît travaillant sous Linux et Kevin sous Mac. Une fois cette première barrière franchie, nous avons découpée l'écran en différente zone. Cela nous a enfin permis de rendre cliquable les boutons du deck, puis la carte en cours.

Puis on a traité le cas de poser une tuile sur la grille. Lorsque le joueur clic sur la grille, il le fait en général en plein milieu d'une case. Il nous a donc fallu écrire une fonction qui permet de faire la translation entre les coordonnées écran des coordonnées grille de jeu.

Grâce a cette fonction, on a pu implémenter le clic droit qui permet de positionner la tuile sur la grille (sans la poser) afin de vérifier si elle est potable (grâce à notre petit écran de contrôle). Puis vient le clic gauche qui permit de poser la tuile.

On touchait au but ! On a donc voulu aller plus loin avec l'évènement mouse motion qui permet de récupérer le déplacement de la souris, et ainsi faire le survole automatique de la grille sans utiliser le clic droit… Ce fut une catastrophe !

Cette erreur nous permis de mettre le doigt sur un problème général de notre partie SDL : Le chargement de fichier. En effet, chaque fonction charge les fichier images, fonts dont elle a besoin, effectue les « blits », nécessaire sur le pointeur écran puis décharge les fichiers images. Hors toutes ces opérations de bas niveaux (utilisation de fonctions tels que F_OPEN) demandent du temps au système d'exploitation. Et à chaque mouvement de la souris, nous chargions les 30 images du jeu blittait puis déchargeaient celle-ci. Alors si cela ne posait pas de problème au clavier, c'est que ce dernier ne génère pas pas plusieurs milliers d'évènement à la seconde… Le déplacement de la souris si !

Pour pallier ce premier problème, nous avons donc décider d'utiliser un tableau (mis en variable globale) de l'ensemble des textures que l'on ne charge qu'une fois au tout début de la partie SDL, et que l'on décharge à la fin lorsque l'on quitte SDL.

Cela a grandement amélioré les choses mais ce n'était toujours pas suffisant, la souris nous produisait décidément vraiment trop d'évènements. Nous avons donc décidé d'utiliser une deuxième solution : en cas d'évènement déclenché par un mouvement de souris, on contrôle le temps qui s'est écouler avant le dernier évènement similaire. Si moins de 50 millisecondes se sont écoulée, alors on ignore simplement cet évènement.

Enfin notre Honshu est devenu jouable à la souris !

2. Solveur

Nous avons décider d'implémenter un nouveau solveur pour notre jeu. Voici le principe de ce solveur :

- On énumère le nombre d'occurrence de chaque type de terrains présentes sur les tuiles du deck
- Puis on utilise une fonction implémentée au lot C *Solveur_tour* qui renvoie la meilleure tuile à poser pour un tour en maximisant un type de terrain.

Le joueur aura alors possibilité d'utiliser ce solveur au cours de la partie en appuyant sur la touche h. De plus, on a décidé de mettre en place un système de vies qui va restreindre le joueur à seulement 3 appels du solveur au cours de la partie.

3 Remerciements

On tient à remercier les personnes suivantes sans qui notre Honshu ne serait pas tout à fait notre Honshu :

M Tellier : Pour son soutien au cours du développement principalement du LOT_A et du choix de nos structures, ainsi que pour le prêt de la 265.

M Mouillerons : Pour son analyse pertinente sur nos problèmes de ralentissement à la souris dû au trop grand nombre d'évènements.

Mlle Jennyfer Regnier pour la réactivité et la qualité de ses dessins de tuiles.

M Goilard pour son aide et sa disponibilité lors de nos différentes discussions pendant et en dehors des séances PIP.

A GCC et ses messages d'erreurs uniques lors de compilation à 2h du matin.
A SDL pour avoir décidé que non sur les macs c'est pas pareil.

Pour STRANGER C

Jiahui XU, Kevin XU, Heykel Rajhi, Benoît SCHOLL