

Projet Programmation 2

Mathieu Seurin
Elias Rhouzlane

December 19, 2014

Abstract

Ce jeu de ‘SameGame’ est grandement inspiré par le mode 2 joueurs du jeu Pokemon Puzzle Challenge sur GameBoy ([Jouable ici](#))

Ainsi le jeu a été pensé dans ce sens et donc beaucoup de choix ont été fait en prévision de l’implémentation des fonctionnalités du jeu d’origine. C’est pour cela que certains choix peuvent sembler étranges ou inadaptés, mais c’était en prévision de l’implémentation de fonctionnalités plus avancés.

Fonctionnement du jeu

Deux joueurs s’affrontent, chacun avec son plateau composé de diverses cases de couleurs. Les plateaux montent au fur et à mesure, de nouvelles lignes se rajoutent en bas. Une fois que le plateau atteint le haut du jeu, c’est perdu. Pour éviter ça les joueurs disposent d’un swapper (curseur) qui permet d’échanger deux cases. Lorsqu’au moins trois cases de la même couleur sont alignées (horizontalement ou verticalement) on supprime ces cases, permettant ainsi de libérer le plateau petit à petit. La montée du plateau est de plus en plus rapide, rendant ainsi l’affrontement de plus en plus tendu. En plus de cette montée, si les joueurs arrivent à faire des combinaisons de plus de 3 couleurs, ils peuvent envoyer des mauvais blocks à leurs adversaires. Ces mauvais blocks occupent plusieurs cases, ne peuvent être swappés, et ne sont pas désolidarisables. Le seul moyen de les détruire est de faire une combinaison de trois couleurs juste à côté d’eux, les transformant ainsi en cases de couleurs utilisables.

Objectifs atteints

- Avoir un menu avec des effets simples
- Avoir un moteur d’affichage permettant d’ajouter facilement de nouveau effets et de nouveaux menus
- Pourvoir générer des plateaux remplis de cases de couleurs
- Pouvoir afficher deux plateaux jouables en parallèles
- Pouvoir déplacer le curseur à l’écran, parcourir les cases, échanger les cases
- Avoir un algorithme de suppression des cases alignées en ligne et colonne (pas de diagonale) qui supprime en cas de chaîne de plus de 3 cases de même couleur
- Pouvoir générer des mauvais blocks (‘bad’ block) qui sont composés de plusieurs cases simples, censés vous bloquer puisqu’ils ne sont ni interchangeables (pas de swap) ni désolidarisables
- Avoir un algorithme qui gère la gravité du plateau non seulement pour les cases simples mais également pour les ‘bad’ blocks qui sont composés de cases simples ‘soudés’ entre elles, ainsi un ‘bad’ block ne peut tomber que si toutes les cases sous lui sont libres (non implémenté)
- Générer des lignes de cases
- Avoir un système de score
- Avoir un système d’affichage d’information pour chaque joueur
- Fin de jeu, lorsque la ligne du haut est dépassé

Objectifs restants

- Pouvoir envoyer de mauvais blocks à l’adversaire à l’aide de combos
- Quelques glitches d’animation à résoudre
- Pouvoir sauvegarder, charger, continuer et mettre en pause une partie
- Améliorer le système d’affichage d’information et le rendre plus lisible
- Enregistrer les scores et faire un tableau des scores
- Créer des modes histoire et en ligne en plus du mode arcade implémenté

Choix divers

Pourquoi une “hidden row” ?

Le plateau est censé monter petit à petit et non pas par à-coup, ainsi on est censé voir la ligne du bas se montrer petit à petit jusqu’à ce qu’elle soit active. Ainsi il fallait qu’elle soit déjà générée et disponible pour pouvoir l’afficher (même si elle aurait été grisée pour montrer qu’elle n’est pas encore active) Ceci a pas mal de conséquences, notamment de lors des parcours de board, les indices commencent souvent à 1 pour les lignes (row), bien qu’ils commencent à 0 pour les colonnes (col)

Les bad blocks

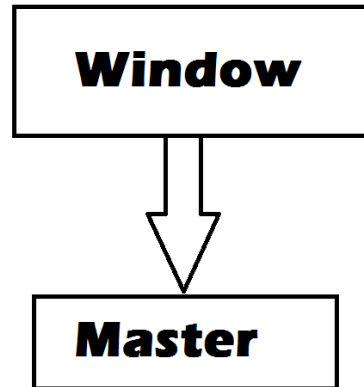
Les bad blocks ont posés un problème car lors de leur implémentation nous avions déjà fait la plupart des fonctions, notamment destruction, gravité et swap, il a donc fallu recoder ces fonctions pour pouvoir gérer les Bad blocks, puisque ceux-ci ont un comportement assez particulier, bien différents des simples cases de couleurs. Comment gérer la gravité : Il nous semblait plus simple d’appliquer la gravité sur tous les blocs, quitte à séparer les bad blocks, puis reconstruire le bad block, plutôt que de gérer chaque cas, car il y a aurait eu beaucoup de problème en cas de plusieurs bad block ou lorsque des cases se retrouvaient au-dessus d’un bad block

Le passage aux Sprites

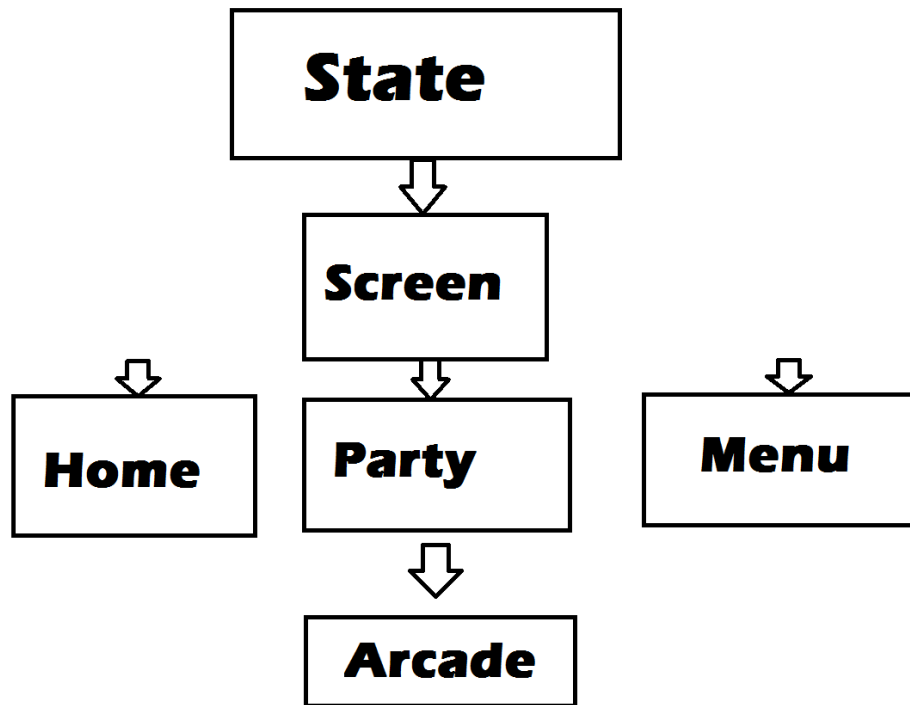
Dans cette version certains objets visuels sont des sprites, on double quasiment les fps de ce fait car on ne régénère pas tout le plateau à chaque tour de boucle.

Arbres d'héritage

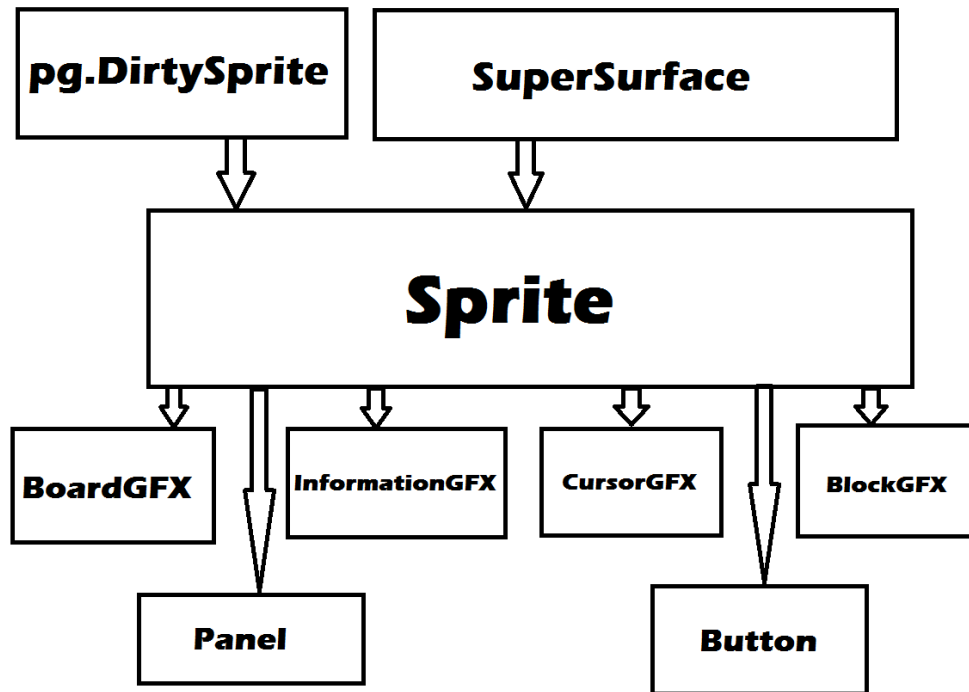
Window



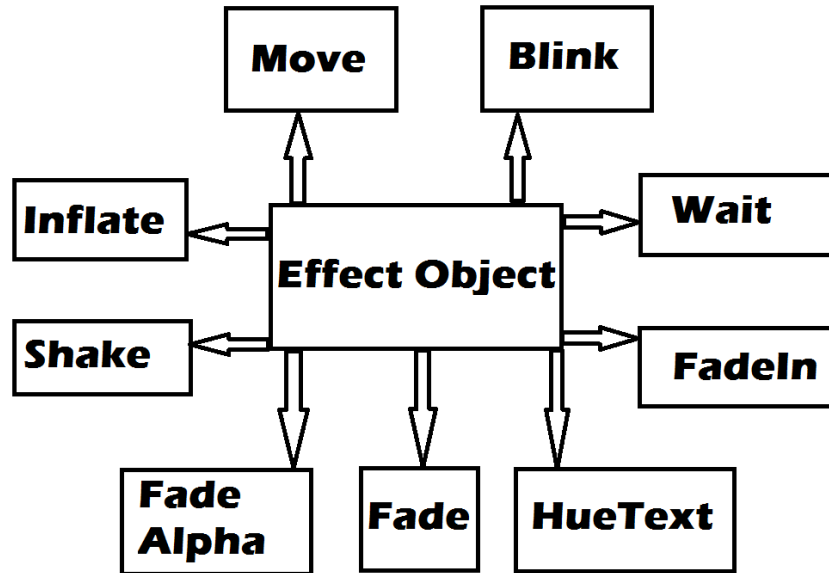
State



Sprite



Effect



PlayerInformation

