

Documentation technique

- Moteur graphique :

La plateforme Native Client ayant un fonctionnement très spécifique, du à son fonctionnement à l'intérieur d'un navigateur, le portage d'un moteur existant est inenvisageable, car il s'agit d'un code trop complexe et trop lourd. Coder notre propre moteur en utilisant l'API Pepper était en fait une solution plus simple et plus rapide.

Fonctionnement

Le code de rendu est en fait divisé en plusieurs parties, correspondant à différents niveaux d'abstraction.

Tout d'abord, ce code ne part pas de rien mais se base sur l'API Graphics2D de NaCl. Cette API permet d'accéder à un contexte, qui correspond à un cadre dans le navigateur. A chaque frame, le programme crée un buffer et y écrit la nouvelle image, puis envoie ce buffer à Chrome afin de l'afficher dans la page.

Le buffer est une image, symbolisée de la manière la plus simple qui soit : un tableau, dans lequel chaque case correspond à un pixel. Ou plus précisément, la couleur de ces pixels, encodée sur 32 bits (Le rouge, le vert, le bleu et la transparence, chacun encodés sur 6 bits).

Plutôt que d'écrire directement dans ce buffer, un nouveau niveau d'abstraction est ajouté dans rendu.cc avec les fonctions DrawRect, DrawCircle et DrawTexture. Ces fonctions permettent, comme leur nom l'indique, de dessiner des rectangles, des cercles et des textures dans un contexte. Ces fonctions peuvent être appelées depuis le code de rendu du jeu pour le simplifier.

Les textures sont en fait des structures contenant 4 informations : la hauteur et la largeur de cette image, le nombre de canaux (3 canaux signifient que l'image n'utilise pas le canal alpha et n'a donc pas de transparence), et un buffer. Ce buffer contient les données de couleur de l'image, cependant pour faciliter l'application des textures (notamment la gestion de la transparence) les cases du tableau ne font que 6 bits et n'encodent donc qu'un seul canal. Au moment du dessin de la texture, il faut donc lire 3 ou 4 cases de ce tableau pour faire un pixel, en fonction du nombre de canaux.

Afin de faciliter le chargement des textures, un format de fichier a été créé. Les textures sont à la base des fichiers PNG, converties en fichier .tex par l'utilitaire png2tex dont le code se trouve dans le dossier tools. Celui-ci enregistre simplement la structure au format binaire dans le fichier, qui est ensuite chargé de la même manière dans le jeu par la fonction LoadTexture. Cette fonction monte le protocole HTTP comme un système de fichier afin de pouvoir accéder aux fichiers du serveur avec un simple fopen, et effectue ensuite l'opération inverse de png2tex. Une fois les textures chargées, elles sont stockées dans un tableau faisant partie de la structure principale du jeu, state.

Utilisation

Ces fonctions utilitaires sont ensuite utilisées dans title.cc (fonctions TitleDraw, PauseDraw et ScoreDraw) et game.cc (fonction GameDraw). Ces différentes fonctions sont appelées 60 fois par

secondes pour actualiser l'image, et n'ont en fait qu'à appeler ces sous-fonctions en déterminant simplement à quelle position dessiner quelle texture.

- Rebonds de la balle:

- Traitement des bonus/malus :

Les bonus, malus aussi appelés drops sont appelés lorsqu'une brique est détruite, ensuite il y a une chance sur 3 qu'on appelle la fonction `SpawnDrop` qui génère les drops. Cette fonction choisit aléatoirement un drop parmi les 8 possibles, ensuite celui-ci est dessiné et la position verticale augmente pour que le drop descende.