

# 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:

### # Fonctionnement des balles

Tous les éléments du jeu sont stockés dans une structure nommée state. Il en va de même pour les balles, la raquette et les briques.

Les données des balles présentes à l'écran (4 maximum) sont stockées dans un tableau. Ces données sont : la position de la balle, son rayon, sa vitesse (en pixel par secondes) et sa vélocité. La vélocité correspond à la vitesse de la balle, mais sur les axes x et y. Pour actualiser la position de la balle, il suffit donc simplement d'ajouter sa vélocité à sa position.

### # Rebond contre la raquette

Lorsque la balle se trouve en bas de l'écran et est alignée avec la paddle, la fonction PaddleCollision (calc.cc:81) est appelée. Cette fonction recalcule la vélocité de la balle en fonction de sa vitesse et de sa position par rapport à la raquette. Un angle est tout d'abord déterminé, allant de 45° à l'extrémité droite de la raquette jusqu'à 135° à l'extrémité gauche. Cette fonction est ensuite convertie en radians grâce à la formule  $\text{rad} = \text{deg} / 180 * \pi$ . Enfin, cet angle est converti en vecteur avec les fonctions sinus et cosinus, vecteur qui est ensuite multiplié par la vitesse.

### # Rebond sur les murs

Pour faire rebondir la balle contre un bord de l'écran, il suffit en fait simplement d'inverser sa vélocité sur l'axe normal de ce bord (axe x pour les bords gauche et droite, axe y pour le bord supérieur).

### # Rebond contre les briques

A chaque image (fonction GameCalc), le programme itère sur toutes les briques et détecte celles entrant en collision avec la balle. Il détermine ensuite la distance séparant le centre de la balle du centre de la brique. La distance la plus grande correspond à l'axe sur lequel la balle doit rebondir, et le signe de cette distance est le nouveau sens de la balle sur cet axe. A nouveau, il suffit simplement d'inverser la vélocité de la balle sur cet axe.

## - 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. Si le drop touche la paddle alors le score augmente de 1 un switch regarde alors de quel drop il s'agit, si c'est DROP\_PADDLE\_PLUS la taille de la paddle augmente de 50 px.

L'effet peut se cumuler jusqu'à ce que la paddle atteigne la taille définie dans MAXPADDLE.

Si c'est DROP\_STICK la paddle devient collante.

Si c'est DROP\_CLONE on fait appel à la fonction AddBall qui va créer une balle à laquelle elle va assigner une vitesse et le type BALL\_CLASSIC puis il incrémente la variable ballCount qui a pour maximum 4, au-delà les balles ne seront pas créés.

Si c'est DROP\_EXPLODE on change le type de la balle en BALL\_EXPLODE.

Si c'est DROP\_LOSE la vie descend de 1 et il recommence sur une nouvelle vie.

Si c'est DROP\_PADDLE\_LESS la paddle diminue de 50 px, cet effet peut se cumuler jusqu'à la valeur de MINPADDLE.

Si c'est DROP\_SPEED\_LESS la vitesse de la balle est divisée par 1.5 et la vitesse diminue de 3, cet effet est cumulable jusqu'à la valeur de MINSPEED. Si c'est DROP\_SPEED\_PLUS la vitesse est multipliée par 1.5 et la vitesse augmente de 3 jusqu'à atteindre la valeur de MAXSPEED.