

Rapport - Bombeirb

PG110

Pigaglio Matthieu et Sergent Adrien

Le programme rendu n'a, à notre connaissance, aucun bug à proprement parler. L'ensemble des fonctionnalités attendues ayant été implémentées nous nous attarderons sur ce que l'on pourrait voir comme les spécificités de notre programme.

I les déplacements

Nous avons changé la manière de recevoir les entrées clavier, afin de ne plus reposer sur le `keyboard_repeat` de SDL. De plus le joueur a maintenant un attribut `cooldown` qui lui interdit de se placer deux en un laps de temps donné (on contrôle ainsi aisément sa vitesse).

II déplacement des caisses

On ne peut pas écraser un monstre avec une caisse, et lorsque une caisse coince ou sur une clé ou sur un bonus on a fait le choix de les ajouter à l'inventaire du joueur.

III les mondes

Le programme supporte des cartes de différentes tailles.

Lors d'un changement de carte, le programme cherche la porte (dans la carte vers laquelle le joueur se dirige) qui permet de revenir au monde que le joueur est en train de quitter, si cette porte n'existe pas des erreurs peuvent survenir.

IV les bonus

Le nombre de vies, de bombes et de clé est limité par 9 pour des raisons d'affichage.

V l’affichage

Le sol utilise une fonction de x_{sol}, y_{sol} et du niveau pour choisir la variante de sol à afficher. Celle-ci est complexe afin de faire croire à une distribution aléatoire.

Il est désormais possible de faire en sorte que certains éléments du décor aient une animation, comme en témoignent les murs avec torches.

On a ajouté un premier plan: celui-ci est représenté par une liste chaînée qui est générée au chargement de la carte. Cette liste est ensuite parcourue après chaque affichage du jeu pour apparaître au premier plan.

VI les bombes

Les bombes fonctionnent en dehors du monde courant et modifient bien le niveau concerné.

VII les animations

Nous avons ajouté un système d’animations qui lui aussi est représenté par une liste chaînée qui est parcourue à chaque affichage.

Sont implémentés: des animations qui se jouent sur le joueur et des animations qui se jouent à une position donnée dans un monde donné. Cependant du fait de la construction modulaire de cette structure d’animation il serait simple d’ajouter un nouveau type d’animation (sur les monstres ou dont le déplacement est dicté par une fonction extérieure au module).

VIII fin de jeu

L’objectif est un coffre, il faut une clé pour l’ouvrir.

IX un menu

Nous avons ajouté un menu au début du jeu. On revient à celui-ci en cas de défaite, victoire ou en quittant le jeu.

X Les monstres

Les monstres sont paramétrés via des documents textes. Ils sont stockés dans une matrice lors du chargement. L’information du sous type de l’objet monstre dans la carte et du niveau actuelle permet de déterminer la

position du monstre dans cette matrice. On peut ainsi avoir jusqu'à 16 monstres différents et dont le caractère est déterminé dans un niveau (limite lié à la taille du sous type) dont 9 peuvent être dans des caisses (taille du sous type moins les valeurs de sous de type de caisse déjà occupé par les bonus).

Les monstres peuvent avoir plusieurs vies, ont une vitesse paramétrable et sont étourdies lorsque qu'ils reçoivent des dégâts (durant un temps dépendant de leur vitesse).

Les monstres ne se déplacent pas dans une carte où le joueur n'est pas présent.

Vis à vis de l'intelligence artificielle:

lorsque la distance qui sépare le monstre et le joueur est inférieure à l'attribut range du monstre. Le monstre traquera le joueur. Cette traque est dirigée par une matrice créée par un parcours de carte que l'on pourrait assimiler à un Dijkstra simplifié (car le poids de chaque arête est identique). Ce parcours n'est fait qu'une fois par déplacement du joueur, pose ou explosion de bombe (elle est commune à tous les monstres de la carte).

Cette intelligence artificielle a 4 variantes :

- La première ignore le danger.
- La deuxième fuit le danger et essaye de rejoindre la zone sécurisée la plus proche uniquement lorsqu'elle traque le joueur.
- La troisième fuit le danger lorsqu'elle ne traque pas le joueur.
- La dernière tentera toujours d'éviter le danger.

Pour finir nous avons fournis 3 versions du code pour lesquelles la notions de danger varie:

- Dans la première toute case dans la zone d'effet d'une bombe est jugée dangereuse.
- Dans la deuxième seul les bombes qui sont dans la seconde précédant leur explosion sont considérées. (La version la plus "jouable" les monstres sont "trop fort" dans les autres)
- La troisième est identique à la deuxième sauf qu'elle prévoit en plus les effets en chaîne.

XI La sauvegarde

La sauvegarde que nous avons implémenté ne contient pas seulement l'état des cartes et du joueur mais aussi les monstres et les bombes.

XII le son

Nous avons ajouté la possibilité de jouer des sons.