Rapport PPFA

Table des matières

Manuel utilisateur

Menu

Lorsque vous lancez le jeu vous arrivez sur le menu ci-dessus.

Pour naviguer sur le menu voici les touches :
- Z : Allez au bouton au-dessus de celui sélectionné.
- S : Allez au bouton en-dessous de celui sélectionné.
- D ou Espace : Appuyez sur le bouton sélectionné.
- Q : Allez au bouton à gauche de celui sélectionné.

Il y a, en tout, 8 niveaux différent (généré aléatoirement par des algorithmes).

Le mode « ENDLESS » permet de lancer une partie de parcourant les niveaux 1 à 8, puis des niveaux aléatoires avec une récupérations des 3 vies à chaque fin de niveau.
Le mode « NIVEAUX » permet de lancer un niveau sélectionner.
Le mode « RANDOM » permet de lancer une partie ayant une suite de niveau aléatoire.

Jeu

Le but du jeu est d'esquiver une suite d'obstacle.

On peut catégoriser 3 types d'obstacles :
- Les obstacles « BASIQUE » qui blessent le personnage lorsqu'il le touche et qui se déplacent d'une direction à l'autre de l'écran. Ces derniers ont toujours la couleur du niveau.
- Les obstacles « MUR » qui permettent de déranger physiquement le personnage. Ces derniers sont blancs.
- Les obstacles « BOMBE » qui explose après 1,5 secondes, créant un obstacle dans chaque direction. Ces derniers sont jaunes et deviennent de plus en plus rouge jusqu'à exploser.

Les touches sont :
- Z : Déplacer le personnage vers le haut.
- S : Déplacer le personnage vers le bas.
- D : Déplacer le personnage vers la droite.
- Q : Déplacer le personnage vers le gauche.
- R : Retourner au menu principale (quitter le niveau).

En haut à gauche de l'écran, il y a les vies du personnage. En haut droite de l'écran, il y a le temps restant avant que le niveau se termine. (Voir l'interface en annexe)

Il y a 8 niveaux aléatoire différents (chacun identifié par une couleur) :
- Niveau 1 (jaune) : Des obstacles sortent aléatoirement de chaque mur.
- Niveau 2 (rose) : Des lignes d'obstacles sortent aléatoirement de chaque mur.
- Niveau 3 (rouge) : Des lignes d'obstacles sortent du mur de gauche et des obstacles sortent du mur du haut.
- Niveau 4 (cyan) : Des diagonales d'obstacles sortent des murs de droite et gauches et des murs gêne le personnage.
- Niveau 5 (vert) : Des lignes d'obstacles sortent du mur du haut et des obstacles sortent des murs de droite et gauche des murs poussable gêne le personnage.
- Niveau 6 (orange) : Des obstacles en diagonale sortent aléatoirement de chaque mur des murs gêne le personnage.
- Niveau 7 (rose pâle) : Des murs gêne le personnage et font apparaitre de obstacles périodiquement dans chaque direction et des lignes d'obstacles sortent du mur de gauche.
- Niveau 8 (vert clair) : Des obstacles « BOMBE » apparaissent aléatoirement.

Lorsque le personnage touche un obstacle, il perd une vie et devient invincible pendant environ 1 seconde.

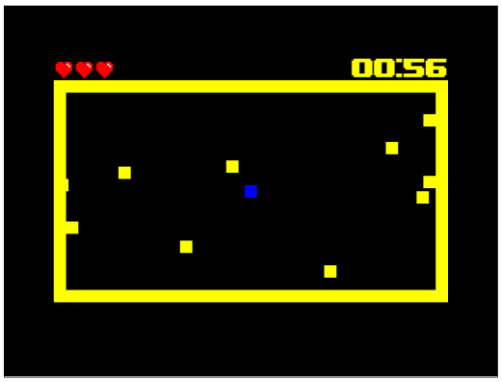
Game Over

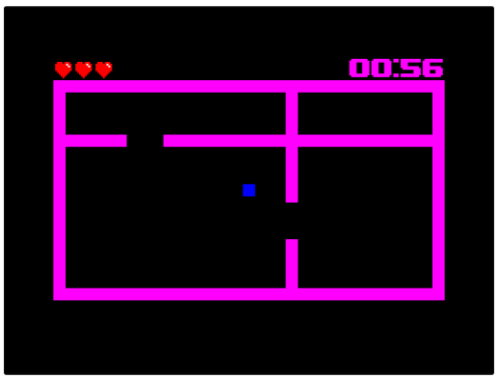
Lorsque le personnage n’a plus de vie, il arrive sur une interface de Game Over.

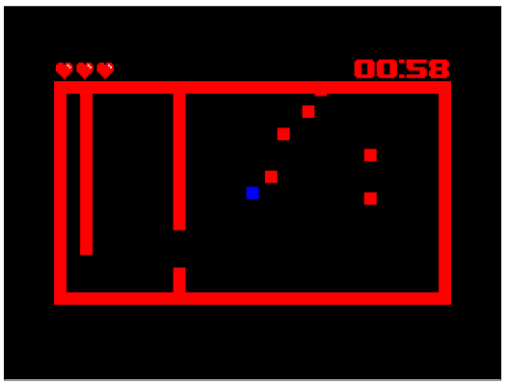
On peut y voir le nombre d’obstacle esquivé.

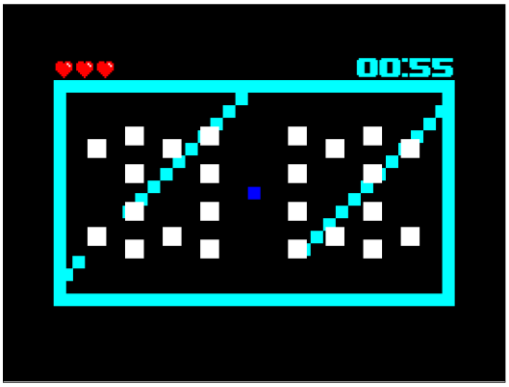
Il suffit d’appuyer sur espace pour revenir au menu principal.

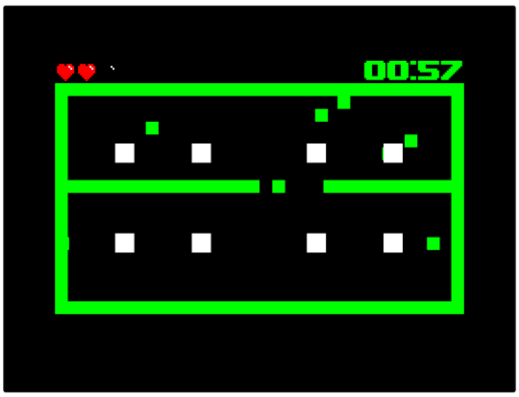
Annexe (screenshot des niveaux)

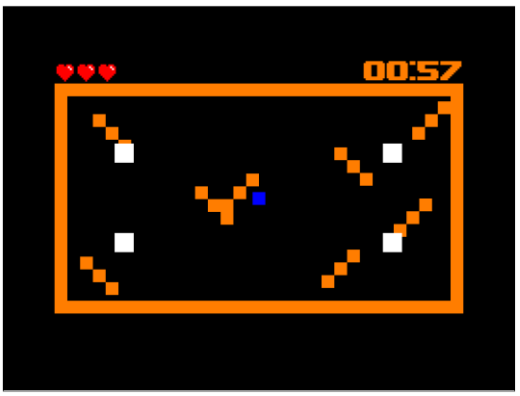


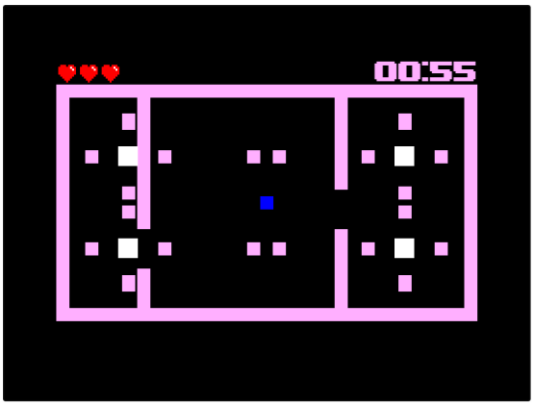


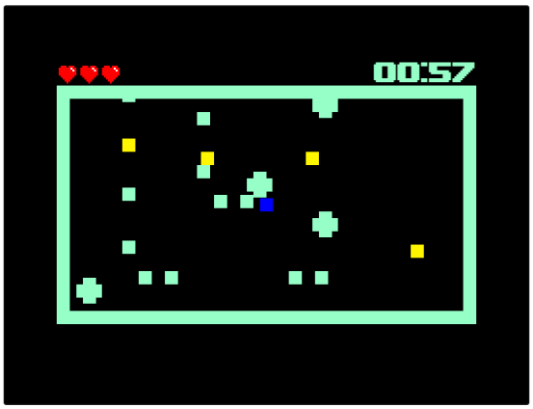












Développement

Architecture du projet

Fichiers	Type	Description
bg.ml bg.mli	Entité	Dessine le fond de l'écran (en noir).
coeur.ml coeur.mli	Entité	Dessine un cœur à l'écran.
data.ml data.mli	Donnée	Gère les données des niveaux (et création des niveau).
game_state.ml game_state.mli	Donnée	Gère les données pendant la phase de Jeu (personnage, obstacle, score, etc).
gameover_manager.ml gameover_manager.mli	Donnée	Gère les données pendant la phase de Menu (texte, événement de sélection de boutons).
menu_manager.ml menu_manager.mli	Donnée	Gère les données pendant la phase de Game Over (texte).
obstacle.ml obstacle.mli	Entité	Crée un obstacle faisant perdre une vie au personnage lorsqu'il y a une collision.
obstacle_bomb.ml obstacle_bomb.mli	Entité	Crée une bombe qui explose en lâchant un obstacle dans chaque direction
obstacle_wall.ml obstacle_wall.mli	Entité	Crée un mur pouvant être posé dans la zone de jeu (celui-ci peut être poussé en fonction de son poids).
player.ml player.mli	Entité	Crée le personnage.
text.ml text.mli	Entité	Crée un texte avec la police d'écriture « UpheavalPro ».
wall.ml wall.mli	Entité	Crée un mur permettant de délimiter la zone de jeu.
coeur_system.ml	Système	Permet de gérer les cœurs en fonction de la vie du personnage (cœur remplis ou non).
collision_system.ml	Système	Permet de gérer les collisions.
control_system.ml	Système	Permet de gérer les événements.
direction_system.ml	Système	Permet de gérer les directions du personnage et des obstacle (utile pour faire déplacer le personnage en diagonale).
draw_system.ml	Système	Permet de dessiner à l'écran.
invincibility_system.ml	Système	Permet de gérer les frames d'invincibilité lorsque le personnage est blessé.
move_system.ml	Système	Permet de gérer les déplacements.
obstacle_bomb_destroyer_system.ml	Système	Permet de gérer la transition de couleur des bombe (Jaune -> Rouge), ainsi que leur destruction après 1,5 secondes.
obstacle_destroyer_system.ml	Système	Permet de détruire les obstacles lorsqu'ils sortent de la zone de jeu.

Choix d'implémentation

Un niveau contient une couleur (rouge, vert, bleu), ainsi qu'une liste d'obstacles.

Un obstacle contient un type (« OBS », « WALL », « BOMB »,), une position (x, y), une vitesse, une direction, le moment auquel il doit apparaitre.

Le niveau est généré au lancement d'une partie, ainsi qu'à la fin de chaque niveau.

Les entités obstacles apparaissent au fur et à mesure que la partie avance, pour éviter de stocker trop d'entités en mémoire.

Difficultés rencontrées

Pendant le projet nous avons dû faire à plusieurs difficultés :

Difficultés	Solutions
Ajout d'une police d'écriture.	Il fallait précharger la police d'écriture dans le css de la page (index.html) pour qu'elle soit utilisable.
Fixer les FPS de la page à 60 (vue que l'un des binômes avait un écran 144Hz le jeu allez plus vite).	On a ajouté un compteur qui permet de limiter les rafraichissements des systèmes de jeu à 60/s.
On voulait pouvoir écrire dans un fichier (par exemple, pour pouvoir stocker des niveaux, et les scores du joueur).	Nous n'avons pas trouvé de solution à ce problème. Malgré tout nous avons toujours la possibilité de convertir les niveaux en chaînes de caractère, pour pouvoir les lires.
Lorsque nous faisions les chargements de nos niveaux via des fichiers texte, nous avions des difficultés pour pouvoir compiler avec dune.	Nous avons donc dû rajouter des « // » pour chaque ligne de nos fichiers dans « resources/static » pour ne plus avoir les erreurs. C'est-à-dire que l'on faisait en sorte que les fichiers ne soit pas compilé car l'on utilisez surement des caractère spéciaux (« @ », « ; », « , », « | ») pour pouvoir séparer nos éléments (voir le fichier « resources/static/data »).

Répartition des tâches

Nous avons fait des « commit » régulier. La totalité des « commit » réalisé le lundi pendant le cours représente le travail fournit par nous deux.

Nicolas s'est surtout occupé du développement des menu et du jeu en générale (obstacle, personnage, collision, etc).

Julien s'est surtout occupé de tout ce qui touché aux niveaux (chargement des niveaux, algorithme de création de niveaux, etc).

Malgré nous avions toujours un œil sur ce que faisait l'autre étant donné que nos travaux était étroitement lié.