**Projet Pac-Man en Javascript**

Petit workflow sur ce projet :

* Apprendre à dessiner sur la Canvas HTML5
* Créer la boucle de jeu qui sera bien sur l’élément central qui va nous permettre de mettre à jour les images en continu
* Charger la carte du jeu grâce au Tilemap, c’est une technique qui utilise une grille de petites images pour construire le fameux labyrinthe
* Ajouter les gestionnaires d’évènements au clics pour diriger les déplacements du Pac-Man
* Créer un algorithme pour déplacer les fantômes de manière aléatoire
* Détecter les collisions entre Pac-Man et les fantômes
* Faire en sorte que Pac-Man mange les pastilles de nourriture
* Ajouter un score dynamique qui va s’actualiser au fur et à mesure que l’on va jouer
* Réinitialiser le jeu lorsque Pac-Man entre en collision avec un fantôme
* Passer au niveau suivant une fois que Pac-Man a mangé toutes les pastilles de nourriture du niveau actuel

Pourquoi ces étapes vont être importante pour vous ?

* **La "Game Loop"** : C'est le cœur de tout programme interactif. En tant que développeur web, comprendre comment rafraîchir une interface 60 fois par seconde est un énorme plus.
* **La "Tilemap"** : C'est très utile en développement web classique aussi (pour créer des interfaces modulaires ou des grilles complexes).
* **La détection de collisions** : C'est de la logique pure. Savoir calculer si deux objets se touchent sur un écran est un exercice de mathématiques appliquées très formateur.

Allez c’est parti, vous allez donc créer un dossier sur votre bureau que vous allez appeler PACMAN par exemple, et créer un dossier dedans qui va contenir les images nécessaire, pour les images vous allez soi les récupérer sur internat soi directement sur mon GitHub ce sera peut-être plus rapide dans le projet Pac-Man il y la dossier images et elles sont toutes dedans. Donc ensuite vous ouvrez ce dossier sur VsCode et vous créez le fichier index.html, le file style.css et le file index.js. Vous créer la structure type du html avec le raccourci ! et entrée. Et on se lance dans le code.

Dans le body vous allez créer le canvas html 5, pour faire simple, le **Canvas HTML5** est une **ardoise magique** que l’on places sur la page web, mais au lieu de dessiner dessus avec un feutre, on dessine dessus avec du **code JavaScript**. C’est exactement ce dont on a besoin.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Voilà pour le début, on va lui rajouter un peu de style,

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

On va passer au JS, on va également penser au plateau de jeu, au nombre de colonnes, de lignes et les case ou on ne peut pas aller donc les case avec un mur, ça va ressembler à quelque chose comme cela,

Une image contenant motif, capture d’écran, Symétrie, carré

Description générée automatiquement

Il y a les points de départ des fantômes, de notre Pac-Man etc... Allez, on passe au JS,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La fin,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Voilà pour le début, c’est déjà pas mal. Maintenant on va regarder la map et penser au positionnement des fantômes ainsi que de Pac-Man, on va penser axe x et axe y ou abscisse ordonnée pour les plus matheux d’entre vous. Et que chaque case vide il devra y avoir de la nourriture pour Pac-Man, oui ca fait pas mal de chose a penser mais n’oubliez pas on utilise le canvas HTML5 qui est prévu pour cela, donc pas de panique on va créer cela.

Je vous montre ou la placer regarder bien la cap écran, Une image contenant texte, capture d’écran

Description générée automatiquement

Voici la map, on va ensuite se créer une classe pour le plateau de jeu, vous la mettrez tout en bas, avant l’accolade fermante,

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Maintenant on va créer le mur, on va le placer juste en dessous de la map,

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Pourquoi j’ai choisi Set plutôt qu’un tableau, et bien en fait c’est un choix très intéressant, pour plusieurs raison, (pas en débutant car on ne connait pas)

* Unicité : un set ne peut pas contenir deux fois exactement le même objet
* Performance : dans Pac-Man, on va vérifier constamment si Pac-Man touche un mur ou une nourriture. Supprimer un élément d'un SET (quand une pastille est mangée) est souvent plus rapide que dans un tableau classique.

On va écrire une autre fonction que l’on va placer juste au-dessus de la classe Block, elle va servir à réinitialiser la grille si on change de niveau par exemple et récupéré la position exacte, puis également a construire le jeu.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police, logiciel

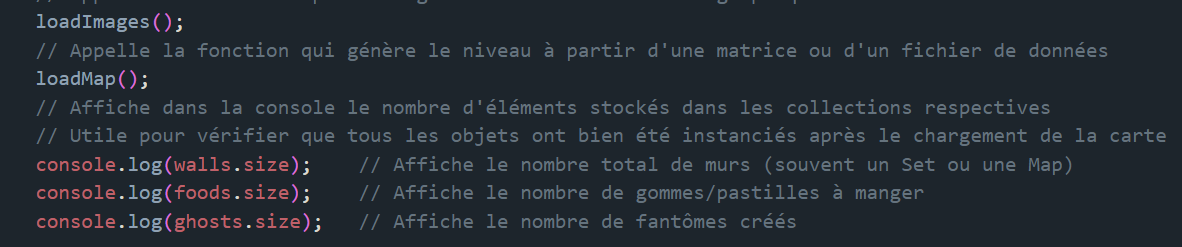
Description générée automatiquement

Et la fin,

Une image contenant capture d’écran, texte, Police

Description générée automatiquement

Attention entre le pinkGhost et Pac-Man, c’est un p et un P, faites attention à cela. Ensuite évidemment on va rajouter la fonction sur le window.onload, en dessous du loadImages(),



Et voilà, on va vérifier les consoles log,

Une image contenant ligne, capture d’écran, Police, texte

Description générée automatiquement

OK c’est bon pour moi, on va pouvoir continuer notre Js, il manque quelques fonctions encore eheh, er commencer à afficher certaines choses à l’écran. Voici les deux fonctions mais ce n’est que le début,

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Si vous ajouter les fonction update(), au window.onload() vous aurez une différence d’affichage, Pac-Man est apparu eheh, c’est un bon début. On continue, et on ajoute les fantômes,

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Voilà qui est mieux à l’écran, on va ajouter, les murs, et la nourriture bien sûr, je vais vous laisser le faire seul.

La voici complète avec les commentaires,

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

OK parfait votre plateau ressemble a quelque chose, il reste à gérer les mouvements aussi, du Pac-Man et des fantômes en aléatoire. On va devoir ajouter des choses à la classe Block,

Une image contenant texte, capture d’écran, Police

Description générée automatiquementVoilà pour la direction, on va créer les fonctions qui vont avec évidemment, voici pour la première,

Une image contenant texte, capture d’écran, Police

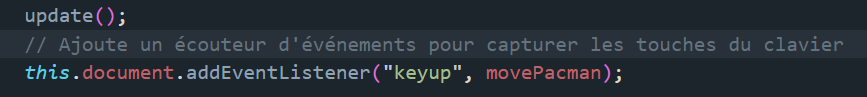
Description générée automatiquement

Et la seconde,

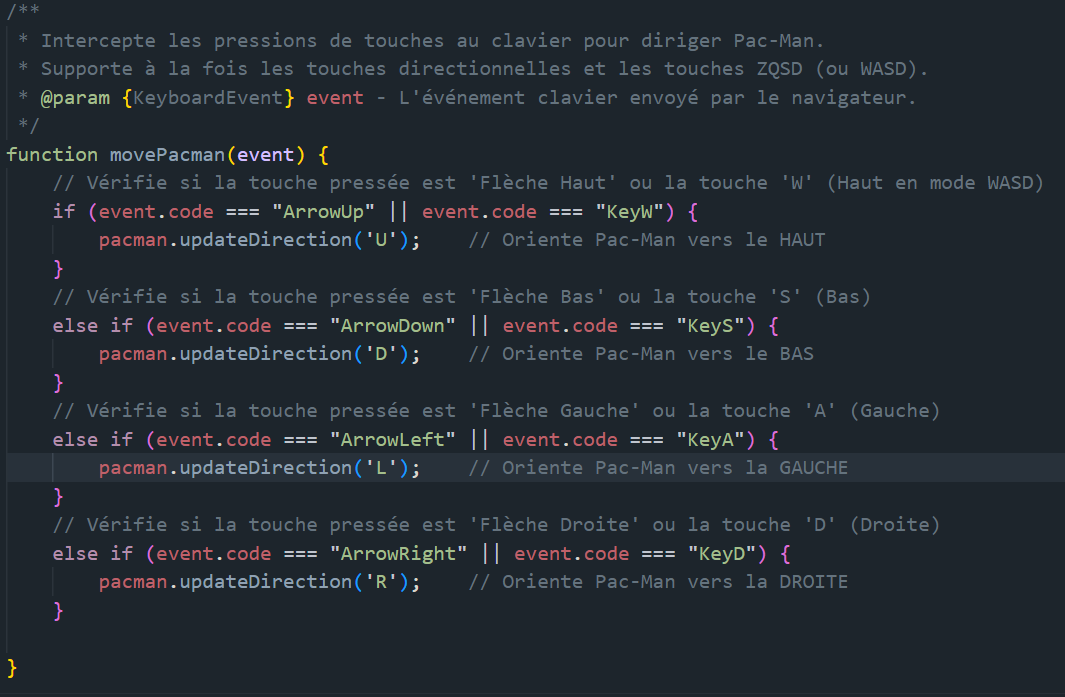
Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Voilà qui devrait être assez clair pour vous, on va pouvoir continuer et revenir sur le window.onload() et ajouter des choses. Déjà pour bouger PacMan,



Et donc créer la fonction movePacman(), on va se mettre sous la fonction draw() et donc avant la classe Block().



Voilà pour celle-ci, on passe à la suivante, elle ira juste au-dessus de celle-ci,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

OK là on est pas mal du tout, on va devoir ajouter cela a la fonction update(),

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

On peut tester, vous allez avoir un bug assez marrant je trouve on mélange Snake et Pac-Man, on va essayer de régler cela, ca se passe dans la fonction draw(),

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Voilà qui est beaucoup mieux déjà, maintenant on va penser au fait que l’on doit reconnaitre qu’il y a un mur ect, il ne faut pas oublier que notre Pac-Man est un carré a la base, on va donc créer une nouvelle fonction, on va la placer au-dessus de la class Block.

Une image contenant pixel, capture d’écran, Caractère coloré, jaune

Description générée automatiquement

Comme ca vous voyez bien qu’il y a des pixel et donc des carrés dont on va se servir pour savoir quand on arrive a un endroit ou il faut tourner ou être bloqué,

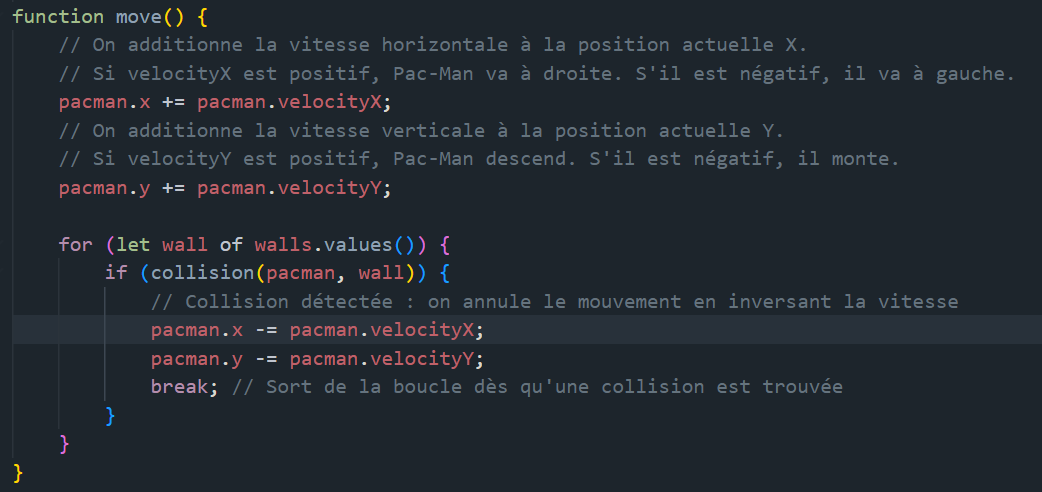
Une image contenant texte, capture d’écran, Police

Description générée automatiquement

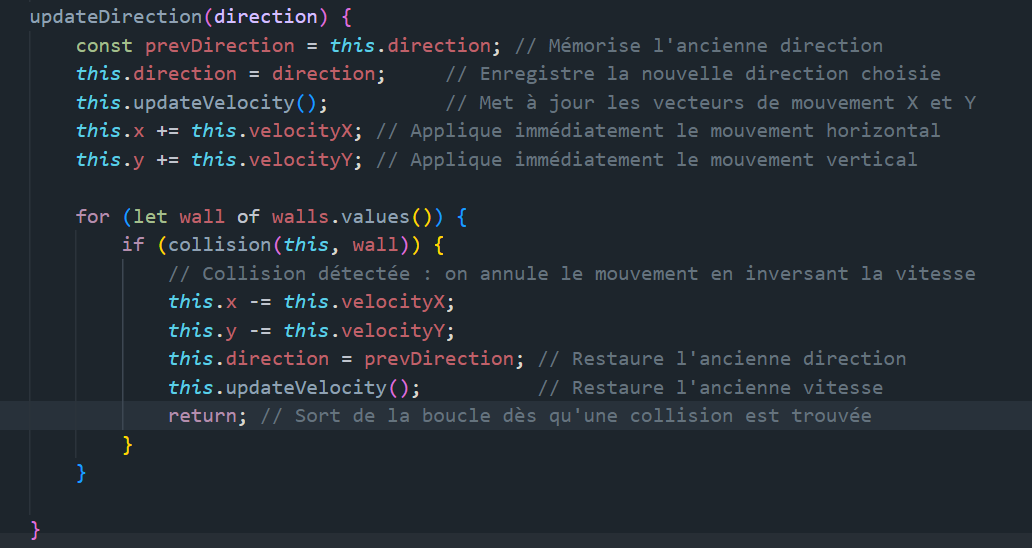
Petite explication supplémentaire, tout d’abord c’est une méthode répandue, cette technique s'appelle la **collision AABB**. Imaginons que chaque personnage est dans un rectangle invisible :

* **a.x** et **a.y** sont les coordonnées du coin en haut à gauche.
* **a.x + a.width** donne la position du bord droit.
* **a.y + a.height** donne la position du bord bas.

Si les **quatre conditions** sont vraies en même temps, cela signifie que les deux rectangles se chevauchent obligatoirement. Cela devrait vous avoir aider un peu plus. Continuons cela, on va retourner dans la function move, et ajouter une condition pour le mur, je vous la redonne en version actualisé,



Vous pouvez tester vous allez voir une grosse différence déjà, super on peut continuer, je viens de constater un petit bug, si on est dans une descente et qu’il y a un mur et qu’on essai de tourner vers ce mur et bien Pac-Man s’arrête on va donc devoir régler cela de suite, cela va se passer dans l’updateDirection().



On va pouvoir continuer on a réglé le problème précédent. On va devoir se rendre dans la function movePacman() et ajouter également des choses, il va bien falloir changer l’image de Pac-Man selon la direction choisi car la pour le moment elle ne change pas.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Voilà qui est beaucoup mieux, l’image change selon la direction super. On va pouvoir finaliser et passer au fantômes désormais, on va ajouter une const sous les premières, qui se trouvent sous le canvas du plateau.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Et on va prendre toutes ces const ainsi que celle du canva et les placer tout en haut ce sera mieux qu’elles soient toutes avant les fonctions, donc avant le window.onload().

ON va ensuite ajouter le mouvements des fantômes dans la loadMap(), on va ajouter un s à la const directions afin de ne pas rencontrer de conflit ensuite,



Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

On va ensuite ajouter cela dans la function move(), elle va être un peu plus complète comme on doit gérer la direction aléatoire des fantômes et penser a quand on arrive sur le mur,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

La voici, vous pouvez recharger votre page et vous allez voir la différence, ok super ça fonctionne bien, on va penser au score etc.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Le score est plutôt simple sur Pac-Man, quand il mange la nourriture ca ajoutes des points donc on va faire cela, on va gérer cela dans la function move() comme précédemment,

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Super, ça avance très bien, on va s’occuper du score dans la function draw(),

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

OK super cela fonctionne bien, on voit bien tout à l’écran. Maintenant on doit penser à quand Pac-Man rencontre un fantôme, donc on retourne dans move() et on gère cela,

On ajoute simplement une boucle, je vous ai mis la ligne du commentaire en dessous afin de vous guider quant à la place de la boucle for. Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Ok ensuite on va sur updateVelocity() et on ajoute quelques petites choses,

Une image contenant capture d’écran, texte

Description générée automatiquement

Ensuite il faut penser a la function resetPosition() qui n’existe pas encore, on va donc la créer juste au-dessus de la classe Block,

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Voilà qui est plutôt pas mal du tout, on doit penser au fait que nous n’ayons plus de vies et donc game over, on va ajouter une condition dans le move(), dans la boucle des fantômes,

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Voilà qui est mieux, on va également ajouter cette condition dans le update(),

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Encore un petit truc, c’est de relancer le jeu une fois que l’on est game over, on va faire cela dans movePacman(),

Une image contenant texte, logiciel, Police, Logiciel multimédia

Description générée automatiquement

Et on va anticiper le fait que vous puissiez créer d’autre map, et faire en sorte que si vous avez tout mange et bien ce soit le niveau d’après, vous pourrez le faire seul si vous le désirez, ce sera dans move() tout en bas,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Pour le moment j’ai mis que cela recharge la map mais a vous de jouer si vous voulez en créer d’autre et vous rendre dessus 😊. Après plusieurs test je viens de me rendre compte d’une erreur de ma part dans la fonction updateDirection(), on faisait une double vérification inutile des collisions avec les murs, en forçant avec this.x += this.velocity cela crée un ‘test de collision’ instantané, et si cela échoué, on remettait prevDirection, donc pacman refusait de tourner avant la fin du mur face à lui. J’ai donc modifié légèrement movePacman(), cela fonctionne parfaitement la voici. Je vous la remet entière avec les modifications,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La fin,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Là c’est fini, mais on va essayer d’implémenter les cerise qui nous permettent de manger les fantômes durant un temps limité. Donc on crée les variables globales,

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

La gestion des images,

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Et on va ajouter cela dans loadImages(),

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

On va ensuite les créer dans loadMap(), après les foods,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Voilà, maintenant on va dans draw() pour les afficher, toujours après les foods,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Voilà on a des jolies cerises à l’écran, maintenant on va devoir modifier la collision avec les fantômes bien sûr, (dans move()) cela change tout.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ok pas mal du tout on devrait être bien, il nous reste à gérer la partie cerise mangée, et ensuite les fantômes qui changent, on fait la partie cerise mangée. Toujours dans move, ce sera après le food,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

La suite,

Une image contenant texte, capture d’écran, Police

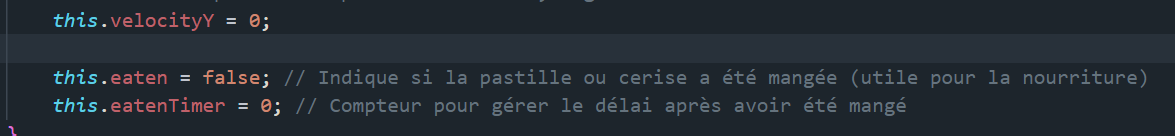
Description générée automatiquement

Voilà pour cela, il reste a géré la modification de l’image des fantômes, et ce sera dans draw(), avant que les fantômes soit dessinés. On va modifier l’étape 2,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Et voilà on est invincible quand on mage les cerises pendants le temps définit. Un seul petit truc me chiffonne dans le vrai Pac-Man quand on mange un fantôme il disparait et réapparait quelques secondes après dans la cage donc on va essayer de faire pareil . On pousse le détail jusqu’au plus prés. On va ajouter l’état mangé, et gérer cela, il va y avoir quelque modifications à effectuer encore mais ça va 😊. On à l’habitude. Donc dans la classe Block on va ajouter deux petites choses,



OK ensuite la partie ou le fantôme est mangé et qu’il réapparait à la cage, ce sera dans move() encore, on ajoute que deux lignes en fait.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Voilà qui ne me semble pas trop mal, il reste a géré le fait de ne pas redessiné des fantômes quand ils ont été mangés, ensuite on les fera réapparaitre dans leur cage de départ. On va donc dans draw(),

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

On a juste ajouté la condition, il reste à les ramener à la vie dans move(), on va ajouter cela après le timer du power up,

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

IL reste le plus sympa, le test, et bien après le test, c’est parfait, peut-être les fantômes restent apeurés un peu longtemps mais à vous de modifier cela seul et amusez vous bien. Je vais modifier le temps qui est trop long, je vous le met pour le plaisir, c’est dans move(),

Non en fait après tets cela vient du FPS de l’ordi je pourrai le faire mais cela deviendra assez technique donc je vais laisser comme cela le code étant déjà bien costaud.

Allez, on se lance dans la construction d’un autre niveau, j’ai modifié la const tilemap,

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

J’ai mis le premier niveau dans le tableau et j’ai créer un second tableau avec l’autre niveau, je vous le donne mais a vous de choisir ce que vous voulez,

Une image contenant texte, capture d’écran

Description générée automatiquement

J’ai également ajoute une variable, ensuite on a quelques modifications à effectuer bien sûr, tout d’abord sur loadMap(), premièrement j’ai ajouté cela,

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

Puis cela,

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

Ensuite on doit remplacer tileMap bien évidemment,

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

C’est un bon début, ensuite on va aller dans move(), et chercher cette partie,

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Et la remplacer par cela,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Puis modifier cela,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Je vous mets tout de manière a voir ou le mettre en place car si vous contrôlez la fin du niveau avant de vérifier si il n’y a plus de nourriture ou cerise vous aurez une erreur de chargement,

Une image contenant texte, capture d’écran, logiciel

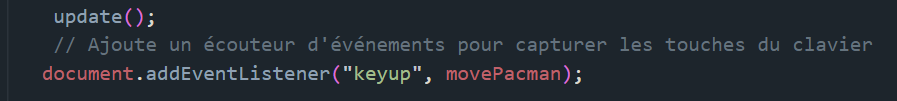
Description générée automatiquement

Ensuite on se rend dans draw(),

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Et on ajoute la dernière ligne, puis on ira dans le window.onload(),



On peut enlever le this devant le document.addEventListener(),