Jeux d’aventure en javascript base sur l’environnement

Idée : jeux d’aventure 2D en javascript sur le thème de l’environnement

A screenshot of a video game

Description automatically generated with medium confidence

Table des matières

[Introduction 3](#_Toc82697604)

[L’informatique 3](#_Toc82697605)

[Description de mon jeu 3](#_Toc82697606)

[Réalisation 3](#_Toc82697607)

[Difficulté engendrée pendant la réalisation de mon jeu 3](#_Toc82697608)

[Github 3](#_Toc82697609)

[La Page Menu 3](#_Toc82697610)

[La Map 3](#_Toc82697611)

[Les Niveaux 3](#_Toc82697612)

[La page de paramètre 3](#_Toc82697613)

[Graphisme 3](#_Toc82697614)

[Méthode et Manière 3](#_Toc82697615)

[Organisation de mon code 3](#_Toc82697616)

[Le Canevas 4](#_Toc82697617)

[Qu’est-ce un cookie ? 5](#_Toc82697618)

[Sauvegarder le progrès du joueur 5](#_Toc82697619)

[Matrix de la map 6](#_Toc82697620)

[Détection de collision 7](#_Toc82697621)

[Interaction avec les objets 7](#_Toc82697622)

[Méthode de l’objet 7](#_Toc82697623)

[L’input 7](#_Toc82697624)

[Déplacement du joueur 7](#_Toc82697625)

[Résultats 7](#_Toc82697626)

[Bilan personnel 7](#_Toc82697627)

[Conclusion 7](#_Toc82697628)

[Bibliographie 8](#_Toc82697629)

[Annexes 8](#_Toc82697630)

# Introduction

## L’informatique

J’ai choisi la branche d’informatique car cela est un sujet qui me passionne beaucoup. L’informatique est une sorte de conversation entre humain et machine. Mes objectifs étaient de programmé un jeu vidéo en 2D en utilisant le canevas sur HTML. J’ai choisi l’environnement comme thème principal de mon jeu car c’est un sujet pertinent en ce moment.

## Description de mon jeu

Le jeu vidéo que j’ai programmé est un jeu d’aventure en 2D. Le joueur incarne un jeune personnage qui vient des trouver les solutions à tous les problèmes qui concernent l’environnement et le climat. Le but est de transmettre ce message au monde entier. Les six niveaux de mon jeu représentent les six continents avec l’antarctique exclus. Le joueur devra éviter les « polluer » , collecter des déchets et apporter les solutions a chaque continent. Mon jeu est capable de sauvegarder le progrès du joueur.

# Réalisation

## Difficulté engendrée pendant la réalisation de mon jeu

## Github

Github est un service de gestion de versions basé sur le web pour les développeurs. J’ai utilisé ce site pour pouvoir accéder a mon code partout et de pouvoir ensuite…

## La Page Menu

## La Map

## Les Niveaux

## La page de paramètre

## Graphisme

# Méthode et Manière

## Organisation de mon code

Text

Description automatically generatedPour pouvoir mieux naviguer entre les fichiers, j’ai créé neuf dossier. Ces dossiers s’appellent Menu, Map, Level\_1, Level\_2, Level\_3, Level\_4, Level\_5, Level\_6, GlobalScripts et Assets. Les huit premiers dossiers on chacun un fichier HTML et des fichiers JavaScript propre à la page HTML. Le dossier Assets est composé d’images et de fichiers audio utilisé dans mon jeu. Le dossier GlobalScripts contient tous les fichiers JavaScript qui sont utiliser dans plusieurs des pages HTML.

Tous les dossiers, sauf Assets et GlobalScripts, ont chacun un fichier .HTML, un fichier index.js, un fichier game.js et des fichiers JavaScript propre à leur fonctionnalité.

Les dossiers des niveaux ont chacun un fichier map.txt et IObjects.txt. ces fichiers textes sont rempli d’un matrix de nombre qui représente une tuile de la map ou un objet interactif.

Les fichiers .HTML charge et nomment toutes les images, le canevas, les audio et les fichiers index.js. Ils cachent aussi les images et décrit les différents éléments présents dans la page .HTML.

Les fichiers index.js définit la tailler du canevas, appele la fonction start() du script game.js et fais tourner la fonction GameLoop qui est appelé à chaque fois que la page est redessinée. Dans la fonction GameLoop, les les fonctions draw() et Update() du fichiers game.js sont appeler à chaque image. Pour appeler la fonction GameLoop() a chaque image on ecrit :

Text

Description automatically generated

Le timestamp représente un point dans le temps indépendant de tout fuseau horaire ou calendrier. RequestAnimationFrame appelle la fonction GameLoop() a chaque fois que l’image est redessiner sur l’ordinateur.

Les fichiers game.js importent tous les scripts nécessaires pour sa page .HTML. Le code ressemble a ceci :

Text

Description automatically generated

Le ./ importe un fichier qui réside dans le même dossier que game.js et le ../ importe les fichiers qui résident en dehors du dossier.

Une classe est ensuite définit. Elle comporte un constructeur et les différents fonctions comme start(), update() et draw(). La fonction start() est utiliser pour appeler ou nommer certaine variable quand le page est ouverte. La fonction update() me tous ces variables a jour à chaque image. La fonction draw() dessine tous les objets du jeu, comme par exemple le jour, la map, les ennemies, les objets interactive, etc.

Tous les fichiers, sauf le fichier Input.js et PlayerProgress.js, JavaScript comporte chacun une classe avec une fonction update() et draw() et divers autres fonctions propre a chaque script.

## Le Canevas

Le canevas est un élément HMTL ou l’on peut dessiner des images et des objets. Pour dessiner sur le canevas on doit définir contexte du canevas comme étant en 2d. La variable du contexte se nomme ctx. Pour le système de coordonné du canevas le 0 de l’axe X et de l’axe Y est en haut a gauche. On peut ensuite écrire ceci pour dessiner une image sur le canevas :

Text

Description automatically generated

Le x\_image et le y\_image représentent la position de l’image sur le canevas. Et le width\_image et le height\_image représentent la taille l’image. On écrit ceci pour recadrer l’image :



X\_ recadrer et y\_recadrer représentent le point ou on va commencer le recadrage. Width\_recadrer et height\_recadrer représentent la taille du recadrage. Pour effacer le canevas on écrit ceci :



Ce code effacera un rectangle du canevas. La taille et la position sont définis par x, y, width et height.

## Qu’est-ce un cookie ?

## Sauvegarder le progrès du joueur

Le progrès du joueur est sauvegardé dans des cookies. En utilisant un code optimiser et des functions bien precises, la tache de sauvegarder le progrès devient très simple. Comme la plupart des fichiers JavaScript de mon jeu, PlayerProgress contient une classe PlayerProgress avec un constructor et des fonctions. Le constructeur est composé de quatres array : newPlayerProgress, newPlayerVariables, playerProgress et playerVariables. Les deux premiers stocks les variables et les valeurs de base et les deux autres sont vides. Il y a quatre fonctions, getSavedPlayer, updatePlayerVariables, getCookie et changeCookie. GetSavedPlayer écrit les cookies si aucun jeu sauvegardé n’est detecté. Si il y a un jeu sauvegardé, la fonction appellera la fonction updatePlayerVariables. Text

Description automatically generated

La fonction updatePlayerVariables est utilisé quand on lance le jeu ou quand on modifie un cookie.

Text

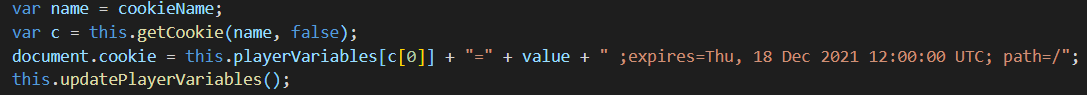
Description automatically generated

La fonction getCookie est utilisé pour récupérer une variable stocker dans les cookies ou une variable de base.

Text

Description automatically generated

La dernière fonction changeCookie modifie un certain cookie avec une valeur donné.



## Matrix de la map

Un matrix de chiffre est utilisé pour créer, changer et dessiner facilement la map des niveaux. Un code général lira ensuite ce matrix et dessinera les tuiles de la map dans la bonne position. Le matrix ressemble à ceci :

A screenshot of a computer

Description automatically generated with medium confidence

Shape

Description automatically generatedLe matrix a 200 colonnes et 23 lignes. Ces nombres correspondent à une tuile dans la map des tuiles. La map des tuiles est une image avec plusieurs tuiles et elles ressemblent a ceci :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 |

Un code dessinera la tuile correspondant aux chiffres presents dans la matrix en recadrant la photo sur la bonne tuile.

Text

Description automatically generated

Les variables sources représente les coordonnées pour recadrer la map de tuile et les variables x et y sont les coordonnées de la tuile sur le canevas. Cette méthode est aussi utiliser pour les objets interactifs.

## Détection de collision

## Interaction avec les objets

## Méthode de l’objet

## L’input

## Déplacement du joueur

Le déplacement du joueur se fait en deux parties, la première est quand le joueur est aux extrémités a droite ou a gauches et la deuxième est quand le joueur est entre ces deux extrémités. Quand le joueur se déplace dans les extrémités ce sont les coordonnés du personnage qui change et la map reste statique, mais quand le joueur se déplace entre ces deux bornes ce sont la map et les objets qui changent de coordonne. Cela donne l’impression que le joueur se déplace alors qu’il reste statique sur l’écran. Cela donne la possibilité de crée des niveaux plus longs avec un meilleure visuel.

Text

Description automatically generated

This.offsetX represent le décalage de la map par rapport a la position du joueur quand il est au milieu de l’écran. Le décalage est négative car la map se déplace de droite a gauche.

# Résultats

# Bilan personnel

Je trouve que ma méthode de travail était bien, je me suis organisé correctement. Je travailler en moyenne quatre heures par semaines. J’ai consacré les deux premiers mois à la découverte du canevas et de son fonctionnement. J’ai pris ensuite trois mois pour poser les bases de mon jeu. Le mois d’août, j’ai principalement optimisé mon code et régler quelque beugue. Le mois de septembre fut consacré à mon rapport et la création de la map de mes niveaux. Le dernier mois j’ai finis mon rapport et a finis mon jeu. Mon seul souci est que je n’étais pas assez précis sur le fil rouge et le but de mon jeu. J’étais très ambitieux de créer un jeu avec un six niveau différents correspondant à un problème lié a la dégradation de l’environnement. Malheureusement ce n’étaient pas mes capacités à coder qui ne m’a pas permis de réaliser jeu mais plutôt les graphismes. J’avais beaucoup de peine à trouver des images uniformes et convenant a ce que je voulais pour mon jeu. J’ai passé autant de temps à chercher pour des images qu’a coder mon jeu. Créer toutes les images moi-même n’était pas une option car je ne suis pas un bon artiste. Finalement je suis satisfait de ce que j’ai réussi à réaliser même si le fil rouge est moins relié au combat contre la dégradation de l’environnement.

# Conclusion

# Bibliographie

# Annexes