

Licence Informatique

Agar.io

Développement Web Avancé

Auteur : Bouchiha Abdelrahim
Numéro étudiant : 15608220

Table des matières

| | |
|--------------------------|---|
| 1 Introduction | 3 |
| 2 Présentation du projet | 3 |
| 3 Cahier de charges | 4 |
| 4 Description du code | 5 |

1 Introduction

Le but de ce projet consiste à programmer Agar.io qui est un jeu vidéo multijoueur free-to-play existant initialement développé par Matheus Valadares.

2 Présentation du jeu

Le joueur contrôle un cercle coloré représentant une cellule, dont le but est de grossir le plus possible en absorbant des cellules plus petites sans être absorbée par des cellules plus grosses.

3 Cahier de charges

Afin de réaliser ce projet, un cahier de charge concernant la création du code à été établi.

Fonctionnalités à implémenter :

- Programmer le jeu
 - Créé la classe Player
 - gérer le déplacement du joueur
 - pouvoir manger les petit cercle
 - pouvoir manger les autre joueur
 - gérer l'affichage
 - Créé canvas
 - dessiner le joueur principale
 - dessiner les autre joueur
 - dessiner les petits cercle
- Programmer le serveur
 - Gérer la connexion et déconnexion des joueur
 - La mise à jour de leur données
 - La mise à jour des petits cercle

4 Description du code

La classe Vector :

cette classe comme son nom l'indique gère les vecteurs et les différentes calculs qu'on peut utiliser sur eux (addition, soustraction, ...).

La classe Circle :

cette classe gère les cercles immobiles en générale, elle me permet aussi de créer une classe mère pour ma classe Player vu que c'est un cercle aussi.

La classe Player :

cette classe hérite de la classe Circle mais elle a des méthodes en plus qui permettent de faire bouger le joueur et de manger les autres cercles et joueurs.

Le fichier view.js :

ce fichier contient toutes les méthodes qui permettent de dessiner dans un canvas.

drawPlayer ⇒ pour dessiner le joueur principalement

drawEnemy ⇒ pour dessiner les autres joueurs

drawCircle ⇒ pour dessiner les petits cercles immobiles

drawGrid ⇒ pour dessiner une grille indiquant la carte sur laquelle les joueurs peuvent se déplacer

drawGameOver ⇒ pour afficher un « GAME OVER »

Le fichier toolbox.js :

ce fichier contient des petites méthodes et événements dont j'ai besoin dans le jeu.

randomColorHex ⇒ pour retourner une couleur hexadécimale aléatoire

random ⇒ pour retourner un nombre aléatoire entre deux nombres

createCanvas ⇒ pour créer et retourner un canvas

deleteCanvas ⇒ pour supprimer un canvas

window.addEventListener('resize', fun) ⇒ pour avoir les nouvelles dimensions de la fenêtre

document.addEventListener('mousemove', fun) ⇒ pour avoir les nouvelles coordonnées de la souris

Le fichier main.js :

c'est le fichier qui contient la méthode principale du jeu « main » qui permet de se connecter au serveur et de lancer le jeu, et il contient aussi la méthode de la boucle de l'animation frame « onFrame ».

Le fichier server.js :

c'est le fichier dédié au serveur avec le quel il gère tout les client connecté.