Rapport Mortal Kombat

Fait par Cyrielle Ndongo Ngondi L3-B 19000288 & Rayane Malik L3-A 20011084

I. Animation et Input (Rayane)
II.Collisions,Barre de Vie et Audio (Cyrielle)

I. Animation et Input

Il m'a fallu commencer par trouver les spritesheets sur internet puis suite à cela j'ai dû supprimer le fond étant donné qu' il était vert.
Une fois avoir fait cela j'ai donc découpé les parties qui m'intéressent de la spritesheet et ai dû faire en sorte de bien placer les personnages dans les nouveaux fichiers étant donné que leurs emplacements n'étaient pas uniforme.

Une fois avoir fait cela j'ai commencé à coder l'animation des personnages pour qu'ils puissent apparaitrent dans la page html, sauf qu'en utilisant la requête window.requestanimationframe cela me faisait un affichage de 60fps, ce qui était trop rapide pour les animations utilisées.

```
+ function init() {
+   //drawFrame(0, 0, 0, 0);
+   window.requestAnimationFrame(walkKano);
+ }
```

Suite à cela j'ai fait en sorte de réduire le nombre de frames par seconde mais j'ai eu une multitude de bugs et ait donc changé ma façon d'afficher les images.

```
function gameLoop(timestamp) {
    if(timestamp > time + time_framerate) {
        time = timestamp;

        ctx.clearRect(0, 0, canvas.width, canvas.height);
```

Le code d'affichage des animations se trouvait donc au dessus du window.requestAnimationFrame qui me faisait des bugs

```
  }
  window.requestAnimationFrame(gameLoop);
```

J' ai donc utilisé une méthode plus ressemblante à celle que nous avions utilisée en classe et cela a supprimé les bugs existants.

```
function update() {

    ctx.clearRect(0, 0, canvas.width, canvas.height);
    kanoLife();
    subLife();
    animationsSubzero();
    animationsKano();

}
setInterval(update, 100);
```

Donc la fonction d'affichage est appelée toutes les 100 millisecondes.

Suite a ca j'ai donc fais le code de déplacement et d'attaques qui ne m'a pas posé de soucis particuliers.

```
let sprint = false;

//code de deplacements et d'attaques
if (keyPresses.ArrowLeft && !is_kano_dead && !babality && !freezed) {
    if (pos_kano>pos_sub+60){
        pos_kano -= 10;
    }
    k_hasMoved = true;
    forward = false;
} else if (keyPresses.ArrowRight && !is_kano_dead && !babality&& !freezed) {
    if (pos_kano + 90<canvas.width){
        pos_kano += 10;
    }
    k_hasMoved = true;
    forward = false;
```

Par exemple, lorsque la flèche droite est enclenché, Kano peut donc se déplacer uniquement entre la position de subZero+60 et sa position. Et lorsque la flèche de gauche est enclenché, sa position +90 et la fin du canvas
Subzero_+60__ Kano
Kano__+90_ EndOfCanvas

II - Collision,Barre de Vie et Audio

Pour ma part, une fois que les animations et déplacements m'ont été transmises j'ai dû travailler sur les collisions. Au départ j'étais parti sur l'utilisation de SAT.js.

```
class Box {
    constructor(posX, posY, w, h) {
        this.posX = posX;
        this.posY = posY;
        this.w = w;
        this.h = h;
    }
}
```

Le problème est survenu et que nous ne pouvions plus push sur GitHub notre code, et que pour Rayane ça ne fonctionnait pas sur son ordinateur. Nous avons donc convenus de pas utiliser la librairie de SAT.js et de trouver un autre moyen.

```
if (oneTwo_sub && (pos_sub + 102) >= (pos_Ermac)) {
     damaged = true;
     audio4.play();
     if (subOneTwoLoop[currentLoopIndexSub] == 4) {
          health -= 20;
          //console.log(health);
          if (health > 0) {
               HealthBarErmac.width = health;
          }
          if (health <= 0) {
               is_Ermac_dead = true;
               HealthBarErmac.width = 0;
               //console.log("Game Over Ermac");
          }
     }

}
```

Je suis donc parti sur l'utilisation de collision AABB mais avec le code que nous avons fait cela était lourd et nous aurions dû utiliser des classes pour que ça soit plus compréhensible. J'ai donc trouver la solutions des positions qui celle que j'ai utilisé toute au long de la création du GamePlay.

```
class HealthBar {
     constructor(x, y, width, height, maxHealth, color) {
          this.x = x;
          this.y = y;
          this.width = width;
          this.height = height;
          this.maxHealth = maxHealth;
          this.health = maxHealth;
          this.color = color;
     }
```

J'ai ensuite créé la barre vie des personnages,où j'ai créé une classe qui s'appelle Health Bar.
Et que j'ai ensuite appliqué pour chaque personnage. Cela à été un véritable gain de temps et de visibilité afin de se retrouver dans le code et les animations de combat. J'ai pu très rapidement les mettre au animations et créer une barre de vie efficace.

```
if(is_sub_dead){
     audio.play();
     ctx.drawImage(finishMode,180,80,300,100);

}
if(is_sub_dead && babality){
     audio.pause();
     ctx.clearRect(180,80,300,100);
     audio2.play();
     ctx.drawImage(babalityMode,180,80,300,100);
}
```

J'ai ensuite créer un audio pour chaque commande pour que quand l'un frappe un son soit émis ou bien lors du friendship ou du babality. Et j'ai bien sûr ajouter le célèbre Finish Him.