UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Moteurs de Jeu

Rapport de Projet : Flappy Bird Like

Université Paris 8 Saint-Denis - Vincennes

Caroline Bernard

4 Janvier 2021

Table des matières

Première partie

Introduction

Mon projet consiste à faire comme le jeu de Flappy Bird, dont le but est de faire voler un oiseau entre des tuyaux sans qu'il les touche, et d'avoir le plus grand score avant de toucher un des tuyaux.

Ce rapport est structuré autour de trois parties. La première comporte le développement du code. La deuxième comporte la conclusion. Et la dernière partie montre les sources utilisées.

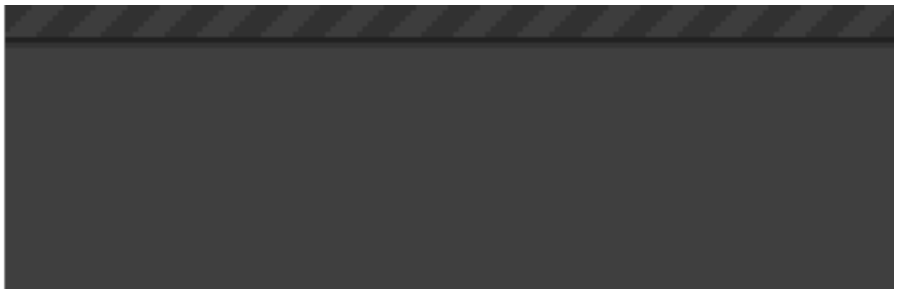
Deuxième partie

Développement

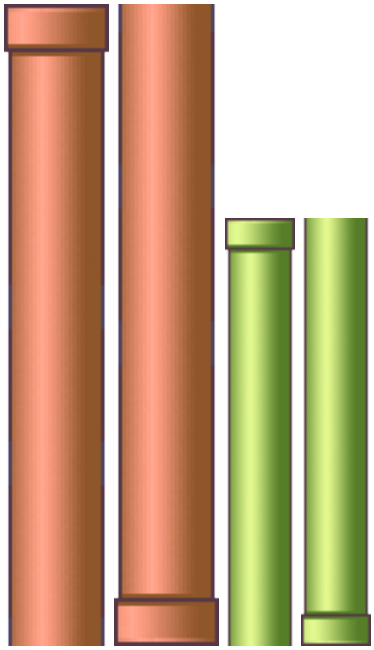
1 Sprites

Pour mon projet j'ai pris les sprites suivant :

Ce sprite est utilisé comme arrière plan du jeu.



Ce sprite est utilisé comme le bas du jeu.



Ces sprites sont utilisés pour les tuyaux du jeu.

Ce sprite est utilisé pour redémarrer le jeu en cas d'échec.

Ce sprite est utilisé comme personnage du jeu.

2 Programme

Pour mon projet j'ai du créer trois fonctions importantes :

- l'animation (interaction avec l'utilisateur)
- une génération procédurale
- et un test de collisions

2.1 Animation

Voici le code de l'animation :

```
// Animation -> Clic
var clicSouris = function(e){
  // Clic (x;y) coin supÃ©rieur gauche du canvas
  var x = e.layerX/echelle;
  var y = e.layerY/echelle;
  if(etatJeu === 'ready'){
    etatJeu = 'play';
  }
  if(etatJeu === 'play' && oiseau.y+oiseau.h > 0){
    oiseau.v = Math.min(0, oiseau.v);
    oiseau.v -= 3;
    oiseau.v = Math.max(-3, oiseau.v);
  }
  else if(etatJeu == 'gameover'){
    if((x > pb.x && x < pb.x+pb.l) && (y > pb.y && y < pb.y+pb.h)){
      reinitialisationJeu();
    }
  }
};
canvas.addEventListener('mousedown', clicSouris);
```

L'interaction avec l'utilisateur est effectué avec un clic de la souris pour faire voler l'oiseau.

2.2 Génération procédurale

Voici le code de la génération procédurale :

```
// Generation de tuyaux
var genereTuayau = function(){
  var tuyau={
    x: pc.start_x,
    y: pc.min_y + Math.random() * (pc.max_y - pc.min_y),
  };
  tuyaux.push(tuyau);
};
```

La génération procédurale est utile pour que les tuyaux s'affiche de manière aléatoire entre chaque nouvelle partie ou actualisation de la page de jeu.

2.3 Test de Collisions

Voici le code du test de collisions :

```
// Test de collision
var collision = function(oiseau, tuyau){
  return (oiseau.x+oiseau.l > tuyau.x && oiseau.x < tuyau.x+pc.l) && (oiseau.y+oiseau.h >
};
```

Le test de collisions est là pour savoir quand l'oiseau va toucher un tuyau. Et donc pour définir la fin du jeu quand le joueur a perdu.

3 Essai du Jeu

Voici trois capture d'écran du jeu :

Image du début du jeu.

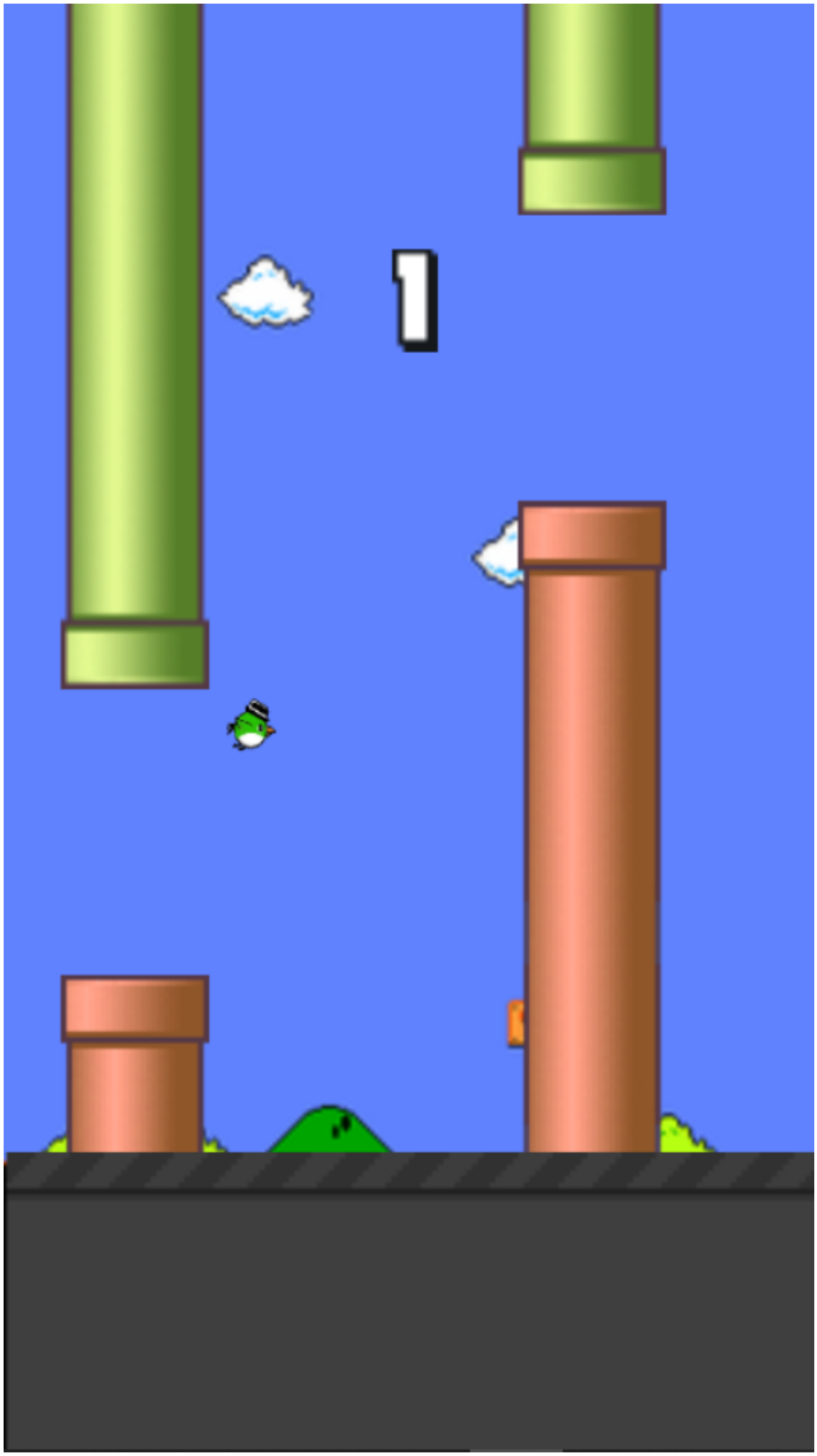


Image en plein jeu.

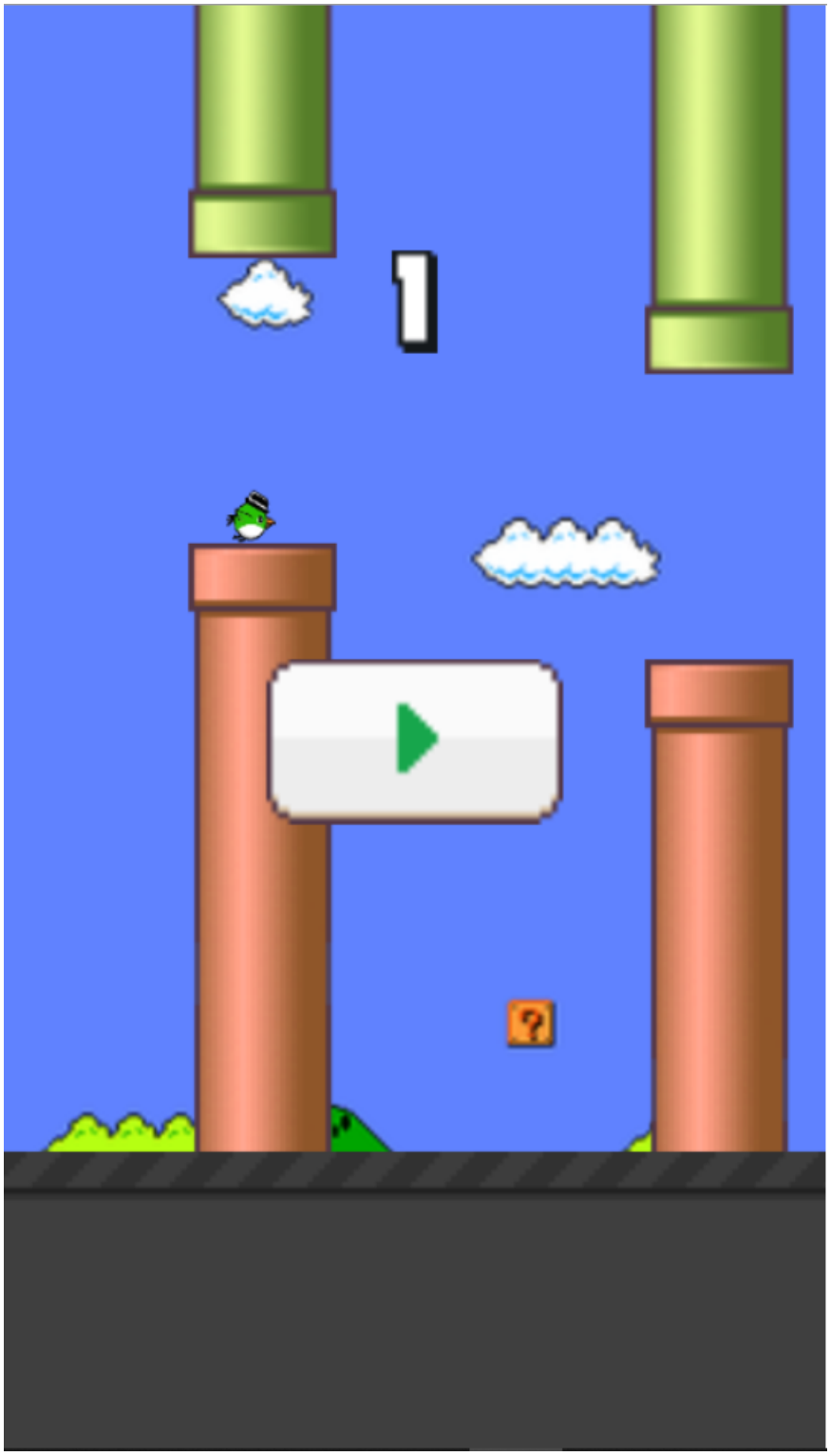


Image en fin de jeu.

Troisième partie

Conclusion

Ce projet m'a permis de m'améliorer dans les langages JavaScript et HTML.

Ce projet m'a permis de consolider mes connaissances sur les langages JavaScript et HTML. Cela m'a également permis de m'exercer à programmer davantage en JavaScript.

Je suis satisfais d'avoir choisi ce sujet. Tout d'abord pour créer un jeu de zéro, avec mes propres envie de sprites et de couleurs, la compréhension de l'énoncé était simple à comprendre pour réaliser ce projet.

J'ai pris du plaisir à faire ce projet mais ce n'était pas facile au début, j'ai du trouver les bons sprites et créer toutes les fonctions nécessaires.

J'ai fais des dizaines d'essais avant de pouvoir arriver au résultat final.

Quatrième partie

Source

Pour faire mon projet j'ai utilisé des Images de Google Image et de The Spriters Resource.