

Contents

Introduction	1
Niveau 1 : Script Kiddie	3
Comment créer votre premier jeu	3
Problème	3
Solution	3
Comment ajouter une image	5
Problème	5
Solution	5
Discussion	5
Comment ajouter un sprite	5
Problème	5
Solution	5
Discussion	6
Comment ajouter une spritesheet	6
Problème	6
Solution	6
Discussion	6
Niveau 2 : Medium	7
Quelle est la meilleur taille pour votre jeu	7
Problème	7
Solution	7
Qu'est-ce qu'un état	9
Problème	9
Solution	9
Quelle est la meilleur manière de définir un état	10
Problème	10
Solution	10
Niveau 3 : Cordon Bleu	13
Comment créer une animation avec un Atlas Json Array	13
Problème	13
Solution	13

Introduction

"Quand j'ai découvert les algorithmes pour la première fois en pensant à comment cuisiner, j'imaginai mon programme comme une recette de cuisine de ma première assiette à l'université, les Tagliatelles à la crème fraîche, une escalope de poulet et mon dessert préféré, la mousse au chocolat. L'entrée n'est pas ma spécialité, mais j'aime parfois une salade près des tagliatelles. Aujourd'hui, je suis loin de mes débuts en cuisine et mes compétences en matière de programmation se sont également améliorées, bien sûr plus ces dernières années sur GitHub et plus spécialement. dans la communauté PhaserJS.

Je vous laisse découvrir comment coder et penser comme faire une assiette avec un reçu de cuisine. Dans ce livre, nous verrons certains problèmes auxquels nous sommes confrontés lors du codage avec Phaser CE. Ce livre ne traite pas de javascript lui-même, mais explique comment coder avec Phaser CE et se concentre uniquement sur les problèmes auxquels vous pouvez faire face lors du codage de votre jeu."

Ce livre est composé de trois chapitres, chaque chapitre est un niveau, rien d'autre.

Les niveaux sont: bien sûr le niveau script kiddie, puis le niveau moyen et bien sûr le niveau Cordon Bleu

Niveau 1 : Script Kiddie

Ce chapitre explique les fonctionnalités de base de Phaser CE, comment ajouter un sprite, une image, du texte et des sons, puis comment déplacer un sprite et de nombreux autres problèmes

Comment créer votre premier jeu

Problème

Vous voulez créer une page html et afficher une image avec laquelle jouer

Solution

Une page HTML de base avec Phaser CE est une page `index.html` où vous ajoutez une balise de script avec la source pointant vers `phaser.min.js`

![Screenshot of the Atom text editor showing the content of index.html. The editor window title is 'index.html - C:\Users\admin\Documents\GitHub\Basic Game - Atom'. The left sidebar shows a project tree with folders 'Basic Game', 'assets', and 'src', and files 'index.html', 'phaser.min.js', and 'game.js'. The main editor area shows the following HTML code: <pre>1 <!DOCTYPE html>
2 <html lang=](game.js)

Figure 1: index.html in atom editor

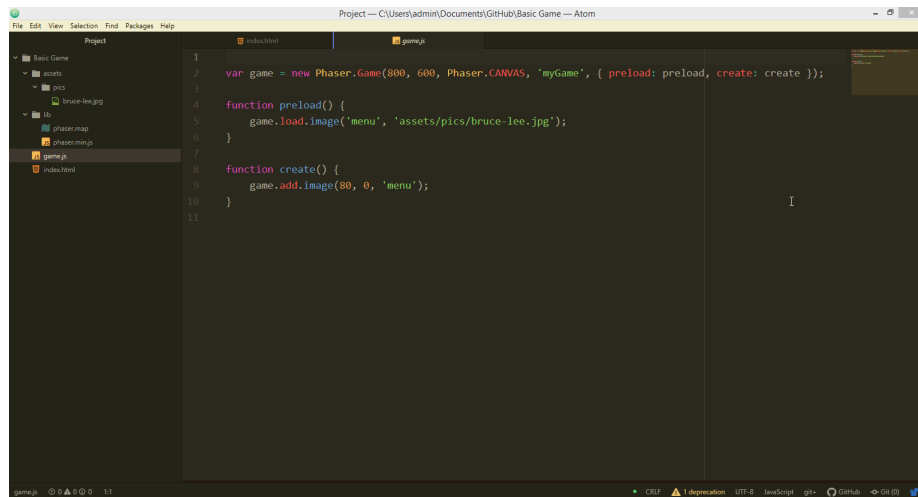


Figure 2: game.js in atom editor

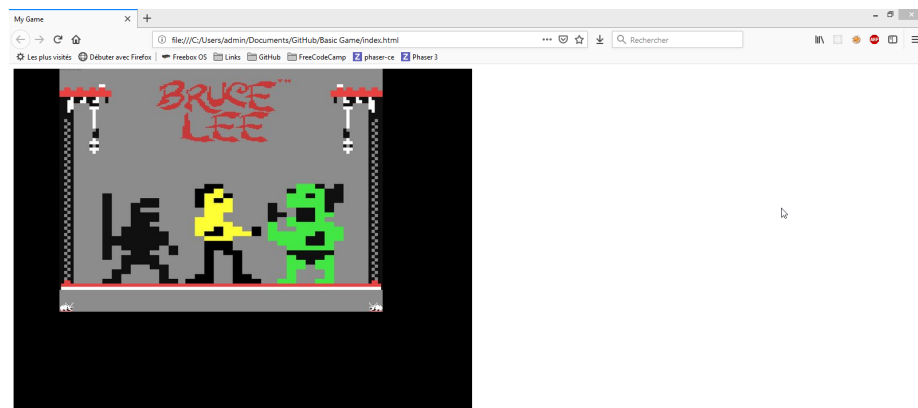


Figure 3: result in browser

Comment ajouter une image

Problème

Vous voulez ajouter une image en tant qu'image de fond

Solution

Votre jeu a peut être besoin d'avoir une image de fond, alors vous aurez besoin de faire ceci

```
var game = new Phaser.Game(640, 360);
var myGame = function () {};
myGame.prototype = {
  preload: function () {
    game.load.image('menu', 'assets/bruce-lee.png');
  },
  create: function () {
    game.add.image(0, 0, 'menu');
  }
}
```

Discussion

Pour ajouter une couleur de fond il suffit d'ajouter cette ligne de code à la méthode create()

```
game.stage.backgroundColor = "#424242";
```

Comment ajouter un sprite

Problème

Ajouter un sprite

Solution

Afin d'avoir un objet avec lequel jouer

```
var game = new Phaser.Game(640, 360);
var myGame = function () {};
myGame.prototype = {
  preload: function () {
    game.load.image('mario', 'assets/mario.png');
  },
  create: function () {
    game.add.sprite(0, 0, 'mario');
  }
}
```

```
}
```

Discussion

La différence entre un sprite et une image sera vu plus loin quand vous commencerez à jouer avec votre sprite en lui ajoutant un corps

Comment ajouter une spritesheet

Problème

Ajouter une spritesheet

Solution

C'est la même chose qu'ajouter un sprite, mais vous avez besoin de charger votre spritesheet différemment qu'une image ou un sprite

```
var game = new Phaser.Game(640, 360);
var myGame = function () {};
myGame.prototype = {
  preload: function () {
    game.load.spritesheet('zelda','assets/tunic.png', 24, 32, 24);
  },
  create: function () {
    game.add.sprite(0, 0, 'zelda');
  }
}
```

Discussion

Après avoir ajouté un sprite vous aurez juste besoin de l'animer (voir animer simplement un sprite)

Niveau 2 : Medium

Ce chapitre traitera des États, de la création d'un jeu Tilemap et d'un jeu à multiniveaux. Nous allons prendre Zelda et le jeu de Phaser CE comme exemple élémentaire.

Quelle est la meilleur taille pour votre jeu

Problème

Vous voulez créer un jeu et vous ne savez pas comment commencer

Solution

Vous pouvez comprendre qu'il existe trois tailles de jeux: Petit, Moyen et Grand.

Petit jeu

Il est bon d'utiliser la fonctionnalité prototype de javascript pour définir un petit jeu. Plus tard dans ce livre nous verrons comment créer un **state**

```
var game = new Phaser.Game(640, 480);
var myGame = function () {};
myGame.prototype = {
  preload: function () {
    this.load.image('logo', 'assets/phaser2.png');
  },
  create: function () {
    this.logo = this.add.image(0, 0, 'logo');
  },
  update: function () {
  }
}
```

Moyen jeu

Le fichier `game.js` ressemblera à ceci

```

bootState = {
  init: function() {
    game.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;
    game.scale.pageAlignHorizontally = true;
    game.scale.pageAlignVertically = true;
  },
  preload: function() {
    game.load.image('progressBar', 'assets/sprites/preloader.png'),
    game.load.image('progressBarBg', 'assets/sprites/preloaderbg.png'),
    game.load.image('loader', 'assets/sprites/loader.png')
  },
  create: function() {
    game.state.start('load')
  }
},
loadState = {
  preload: function() {
    var a = game.add.image(game.world.centerX, game.world.centerY, 'loader');
    a.anchor.setTo(.5, .5);
    var b = game.add.sprite(game.world.centerX, game.world.centerY + 50, 'progressBar');
    b.anchor.setTo(.5, .5);
    var c = game.add.sprite(game.world.centerX, game.world.centerY + 50, 'progressBarBg');
    c.anchor.setTo(.5, .5);
    game.load.setPreloadSprite(c),

    game.load.image('logo', 'assets/sprites/phaser2.png');

  },
  create: function() {
    game.state.start('play')
  }
},
playState = {
  create: function(){

  },
  update: function(){

  }
};

game = new Phaser.Game(320, 640);

game.state.add('boot', bootState);
game.state.add('load', loadState);
game.state.add('play', playState);

```

```
game.state.start('boot');
```

Grand jeu

Ici, je ne veux tout simplement pas mentir. Je n'ai pas encore travaillé sur une telle chose, mais j'ai donné un coup de pied oblique dans le code de Five Nations, un jeu dont j'ai une copie sur mon github avant qu'il ne devienne privé, juste Pensez aux types de jeux "quelque chose.io".

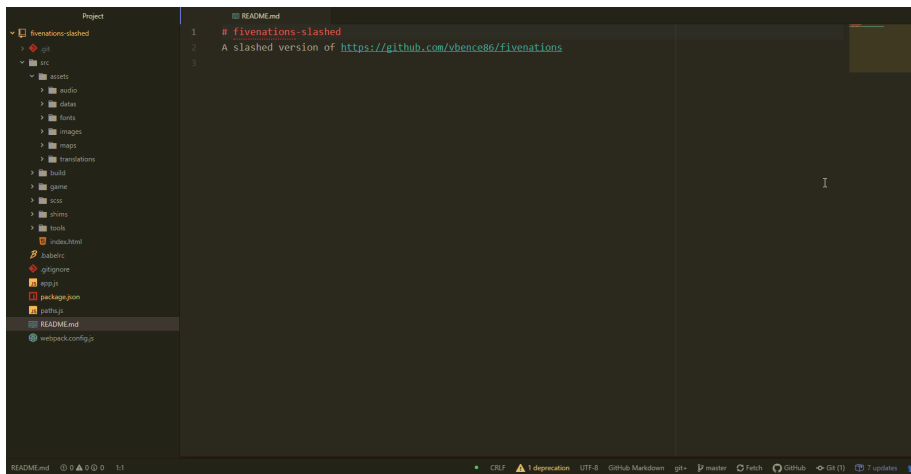


Figure 4: Five Nations

Qu'est-ce qu'un état

Problème

Vous voulez définir un jeu minimal qui pourra être étendu plus tard

Solution

Vous aurez besoin de définir un état, c'est ce à quoi il ressemble et nous utiliserons cette fonctionnalité tout au long du livre

```
var game = new Phaser.Game(640, 360);
var playState = {
  preload : function() {
    this.load.image('dude', 'assets/sprites/phaser-dude.png');
  },
  create : function(){
    this.add.image(0, 0, 'dude');
```

```

    }
};
game.state.add('play', playState);
game.state.start('play');

```

Quelle est la meilleur manière de définir un état

Problème

Vous avez commencé à coder avec des états et vous ne savez pas quel est le meilleur moyen de le coder

Solution

La première méthode est celle qui utilise la méthodologie de prototypage javascript

```

var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'myGame');
var myGame = {};
myGame.Play = function(game){};
myGame.Play.prototype = {
    preload: function () {
        this.load.image('dude', 'assets/sprites/phaser-dude.png');
    },
    create: function () {
        this.add.image(0, 0, 'dude');
    },
    update: function () {

    }
}
game.state.add('play', BasicGame.Play);
game.state.start('play');

```

La deuxième manière est plus codée comme dans le standard Phaser CE

```

var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'myGame');

menuState = function(){
};

menuState.prototype = Object.create(menuState.prototype);
menuState.prototype.constructor = menuState;

menuState.prototype.preload = function() {
    this.load.image('dude', 'assets/sprites/phaser-dude.png');
};

```

```
playState.prototype.create = function(){  
    this.add.image(0, 0, 'dude');  
};??  
  
game.state.add('menu', menuState);  
game.state.start('menu');
```


Niveau 3 : Cordon Bleu

Dans ce dernier chapitre, nous examinerons des expériences plus complexes, mais pas sur JavaScript lui même, nous allons continuer à nous concentrer sur Phaser CE.

Comment créer une animation avec un Atlas Json Array

Problème

Vous souhaitez animer une image-objet avec un fichier Atlas Json.

Pour cela, vous aurez besoin de texturepacker et d'un ensemble de sprites.

Prenons Raiden de l'exemple de combat mortel.

Solution

Premièrement il faut charger le tableau atlas json array dans la fonction preload

```
this.load.atlasJSONArray(  
    'raiden',  
    'raiden.png',  
    'raiden.json'  
);
```

Ajouter l'animation dans la fonction create

```
raiden = this.add.sprite(50, 100, 'raiden', '01.gif');  
anim = raiden.animations.add('idle', [0,1,2,3,4,5,6,7,8,9]);  
anim.play(10, true);
```



Figure 5: Mortal Combat

Conclusion

This book is **done** and ready for *the world to see*, hooray!

