Lier les scènes au index.html et monJeu.js

Lorsque l'on souhaite changer de scène dans Phaser 3, il est obligatoire de changer ces fichiers.

index.html

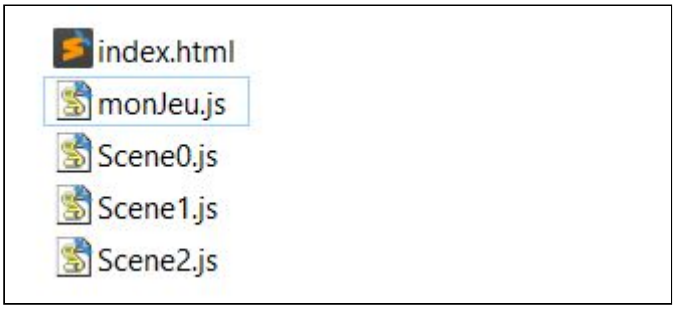
```html
<!DOCTYPE html>
<html>
    <head>
        <script src="phaser.js"></script>
        <script src="monJeu.js"></script>
    </head>

    <body>


    </body>
</html>
```

monJeu.js

```javascript
var config = {
    type: Phaser.AUTO,
    width: 800,
    height: 600,
    scene: {
        preload: preload,
        create: create,
        update: update
    },
    physics: {
        default: 'arcade',
        arcade: {
            gravity: { y: 300 },
            debug: true
        }
    }
};

var game = new Phaser.Game(config);


function preload() {
}

function create() {
}

function update() {
}
```

Il va falloir répertorier dans index.html, l'ensemble des scènes que l'on souhaite appeler.
Par exemple, nous voulons créer 3 scènes. Il va falloir créer 3 nouveaux fichiers :

- *Scene0.js* ;
- *Scene1.js* ;
- *Scene2.js* ;



Maintenant, il nous faut lier ces 3 fichiers au *index*.html car sinon les scènes ne pourront pas être appelés.

Retournons sur le *index.html* et appelons ces 3 scènes de la même manière que *monJeu.js*.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <script src="phaser.js"></script>
5
6          <script src="Scene0.js"></script>
7          <script src="Scene1.js"></script>
8          <script src="Scene2.js"></script>
9
10         <script src="monJeu.js"></script>
11     </head>
12
13     <body>
14
15
16     </body>
17 </html>
```

Retournons dans *monJeu.js* et supprimons les fonctions preload, create et update.

```
var config = {
    type: Phaser.AUTO,
    width: 800,
    height: 600,
    scene: {

        /*preload: preload,
        create: create,
        update: update
        */
    },
    physics: {
        default: 'arcade',
        arcade: {
            gravity: { y: 300 },
            debug: true
        }
    }
};

var game = new Phaser.Game(config);

/*
function preload() {
}

function create() {
}

function update() {
}
*/
```

Il ne reste plus qu'à lister l'ensemble de nos scènes dans *monJeu.js*. Pour cela, allons à la ligne 5 et remplaçons les accolades '{ }' par des crochets '[ ]'.

```
5       scene: [ ],
```

Plus qu'à lister l'ensemble des scènes.

```
5       scene: [Scene0, Scene1, Scene2],
```

Voilà maintenant *index.html* et *monJeu.js* devrait ressembler à cela.

```
<!DOCTYPE html>
<html>
    <head>
        <script src="phaser.js"></script>

        <script src="Scene0.js"></script>
        <script src="Scene1.js"></script>
        <script src="Scene2.js"></script>

        <script src="monJeu.js"></script>
    </head>

    <body>
    </body>
</html>
```

```
var config = {
    type: Phaser.AUTO,
    width: 800,
    height: 600,
    scene: [Scene0, Scene1, Scene2],
    physics: {
        default: 'arcade',
        arcade: {
            gravity: { y: 300 },
            debug: true
        }
    }
};

var game = new Phaser.Game(config);
```

Maintenant que nos scènes sont prêtes à être appeler, on peut créer ce qu'elles vont contenir comme des plateforme, du textes, etc…

Mais avant toute chose il va falloir préciser que ces 3 nouvelles scènes sont une “extension” de *monJeu.js*.

Ouvrons *Scene0.js*.

Pour créer cette extension, il va falloir faire une **classe** (c’est la même idée qu’en C++).

```
1   class Scene0 extends Phaser.Scene {
2
3   }
```

Une fois fait, on utilise un **constructor** pour donner un nom à cette scène. Cela nous permettra de l’appeler au sein même du code.

```
1   class Scene0 extends Phaser.Scene {
2       constructor() {
3           super("Scene_0");
4       }
5
6
7   }
```

Essayons d’afficher un texte dans Scene0.js pour voir si tout fonctionne.

Pour cela, réécrivons les **fonctions principales** qu’utilise Phaser 3 (preload, create et update).

```
class Scene0 extends Phaser.Scene {
    constructor() {
        super("Scene_0");
    }


    preload() {

    }

    create() {

    }

    update() {

    }

}
```

Ajoutons un **texte** et un **cursors** nous permettant de changer de scène lorsque l’on appuiera sur la **flèche du haut**.

```
class Scene0 extends Phaser.Scene {
    constructor() {
        super("Scene_0");
    }

   preload() {

    }

    create() {
        this.text = this.add.text(400,300,'Scene 0 - Changer de scene dans phaser');

        this.cursors = this.input.keyboard.createCursorKeys();
    }

    update() {
        if(this.cursors.up.isDown)
        {
            this.scene.start('Scene_1');
        }
    }

}
```

! Attention !
Lorsque l'on utilise des scènes, dès lors que l'on souhaite utiliser/créer une variable on n'utilise plus "*var*" mais *"this"*.

Habituellement on a vu cela :

```
14
15  var game = new Phaser.Game(config);
16
17  var text;
18  var cursors;
19
20  function preload() {
21
22  }
23
24
25  function create() {
26      text = this.add.text(400,300, 'texte sans importance');
27
28      cursors = this.input.keyboard.createCursorKeys();
29
30  }
31
32  function update() {
33      if(cursors.up.isDown)
34      {
35          /*Mon code*/
36      }
37
38  }
```

Maintenant, nous n’avons plus besoin d'initialiser nos variables que se soit avant la fonction preload ou dans init car le “this” le fait.

Cependant, avant chacunes de nos variables il faudra rajouter “this” pour que notre code marche (voir ci dessous lignes 11,13 et 17)

```
6       preload() {
7
8       }
9
10      create() {
11          this.text = this.add.text(400,300,'Scene 0 - Changer de scene dans phaser');
12
13          this.cursors = this.input.keyboard.createCursorKeys();
14      }
15
16      update() {
17          if(this.cursors.up.isDown)
18          {
19              this.scene.start('Scene_1');
20          }
21      }
22
23  }
```

Exercice :

Dupliquer le code *Scene0.js* dans *Scene1.js* et *Scene2.js* en modifiant le numéro correspondant au scène et lancer Moongoose.exe.