

## Étape 0 – Préparation

Créez dans un dossier un index.html et un script.js, importez aussi les images player.webp et pieces.webp

Installer l'extension Live Server

Ajouter Phaser via HTML, dans le head mettre :

```
<script src="https://cdn.jsdelivr.net/npm/phaser@3.60.0/dist/phaser.js"> </script>
```

## Étape 1 - Configurer le jeu (10 min)

Dans script.js, créez une config Phaser et ajouter une largeur (width) et une hauteur (height) :

```
const config = {  
  type: Phaser.AUTO,  
  width: ?,  
  height: ?,  
  physics: {  
    default: 'arcade'  
  },  
  scene: { preload, create, update }  
};
```

Créez 3 fonction : preload, create et update (function)

## Étape 2 - Charger les images (5 min)

Dans la fonction **preload()**, charger player.webp et piece.webp ainsi que le background avec :

**this.load.image(name, url\_image)**

## Étape 3 — Ajouter le joueur et la pièce ainsi que le background (10 min)

Dans la fonction **create()**, créez le joueur avec **this.physics.add.sprite(coordX, coordY, name)**.

Créez la pièce de la même manière et faite en sorte que le joueur ne puisse pas sortir de l'écran

La fonction **setCollideWorldBounds(true)** empêche le joueur de sortir de l'écran

Après avoir ajouter le joueur avec le sprite, vous pouvez modifier leur taille avec **this.name.setScale(entre 0 et 1)**

Pour ajuster le background à la taille de votre projet (En supposant que vous avez appelé le fond 'background' dans le this.load.image()):

```
const bg = this.add.image(0, 0, 'background').setOrigin(0, 0);  
bg.setDisplaySize(this.scale.width, this.scale.height);
```

## Étape 4 — Déplacements du joueur (10 min)

Touche:

Dans la fonction **create()** permettez au joueur de se déplacer avec les flèches du clavier

Pour vérifier la touche pressée : **this.cursors = this.input.keyboard.createCursorsKeys()**

Dans la fonction **update()**, vérifier quelle touche est pressée avec :

**this.cursors.LaTouche.isDown** (Retourne un booléen)

Pour les flèches sur le clavier : left, right, up, down

Mouvement :

Pour faire bouger horizontalement notre joueur on doit modifier la valeur de : `this.player.x`

Pour faire bouger verticalement notre joueur on doit modifier la valeur de : `this.player.y`

## Étape 5 — Collision et score (10 min)

Dans la fonction **create()** Le but est que lorsque le joueur touche la pièce un score augmente, pour cela nous avons la fonction `this.physics.add.overlap(name1, name2, callback)` qui vérifie si il y a une collision entre nos 2 entités.

Nous devons donc créer un score, pour cela on utilise : `this.scoreText = this.add.text(coordX, coordY, texteAffiche, { fontSize: '24px', fill: '#fff' })`

N'oubliez pas d'instancier un score à 0

`this.name = ?`

Dans `this.physics.add.overlap()` faire :

Pour modifier le score d'un paragraphe on lui rajoute `setText(text)` (Pour utiliser une variable `setText(`Score : $this.variable`)`)

Pour générer des nombres aléatoires nous avons la fonction `Phaser.Math.Between(num1, num2)`

Il nous faut un nombre aléatoire pour la hauteur et pour la largeur qui rentre dans la zone du jeu.

`this.coin.x = ?`

`this.coin.y = ?`

## **Bonus facultatif**

Faire apparaître plusieurs pièces en même temps (tableau)

Ajoutez un ennemi qui bouge et réduit le score s'il touche le joueur (nouveau sprite)

Déplacement de la pièce (on bouge les coordonnées de la pièce)