

Finir le TD1 avant que commencer le TD2

1 Introduction

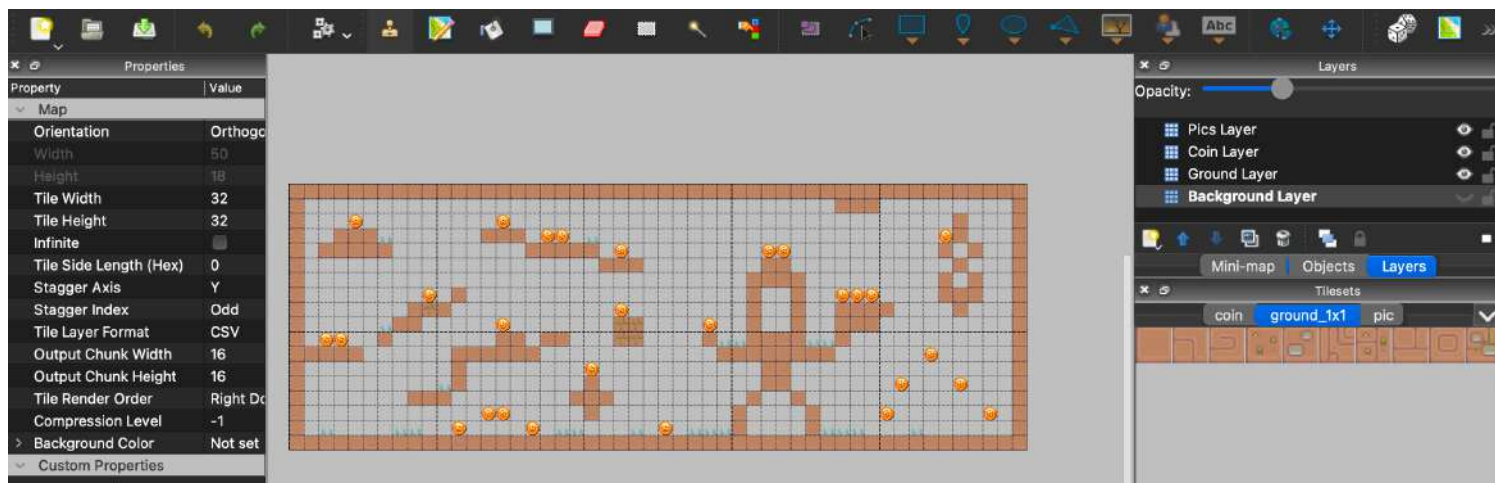
Vous allez réaliser un jeu de plateforme avec défilement horizontal. Il faudra donc gérer une caméra et une map.



Le but du jeu est de récupérer l'ensemble des pièces sans toucher les pics. En haut à gauche, votre barre de vie et en haut à droite le nombre de pièces récupérées.

2 Tilemap

La map est plus grande que la fenêtre de prévisualisation et est contenue dans un JSON avec le jeu de tuiles associé :



Etape 1 :

Définir une fenêtre de rendu de 1024x576 pixels

Mettre la physique en arcade et la gravité à 300

Préchargez les tuiles ground_1x1.png

Préchargez l'image spike

Préchargez les spritesheet coin et dude

Préchargez la tilemap tile-collision-test :

```
this.load.tilemapTiledJSON('map', 'assets/tile-collision-test.json');
```

Etape 2 :

Création de la map et des calques de collision :

```
map = this.make.tilemap({ key: 'map' });
var groundTiles = map.addTilesetImage('ground_1x1');
var coinTiles = map.addTilesetImage('coin');
var picsTiles = map.addTilesetImage('pic');

map.createLayer('Background Layer', groundTiles, 0, 0);

groundLayer = map.createLayer('Ground Layer', groundTiles, 0, 0);
coinLayer = map.createLayer('Coin Layer', coinTiles, 0, 0);
picLayer = map.createLayer('Pics Layer', picsTiles, 0, 0);
```

Attention respectez le nom des TilesetImage (c'est celui du JSON)

Définissez la collision avec les tuiles 1 à 25 du sol :

```
// collision avec les tuiles 1 à 25 : ground_1x1
groundLayer.setCollisionBetween(1,25);
```

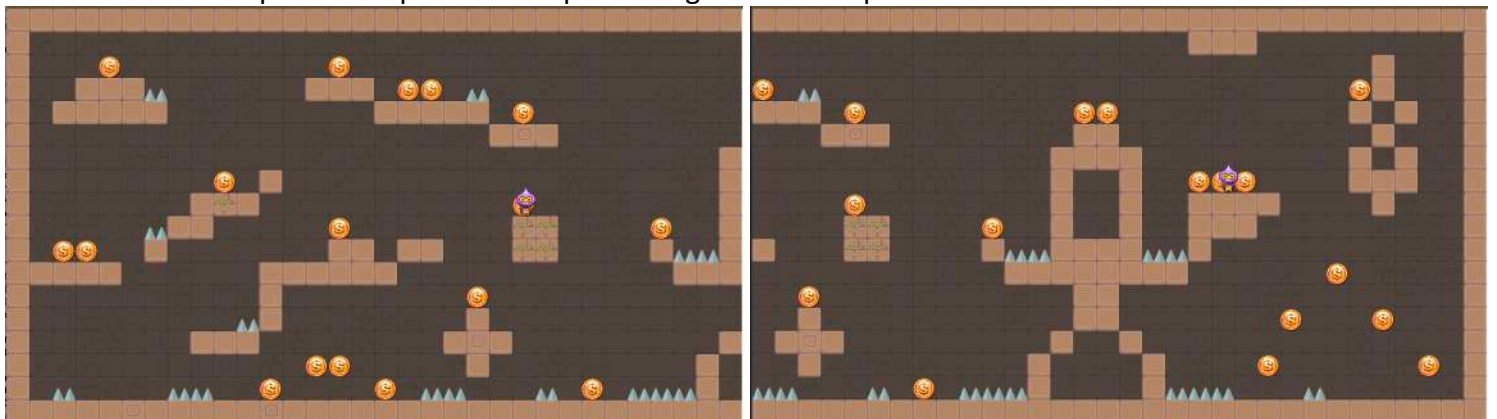
Définissez votre joueur (cf TD1) et et enfin ajoutez la physique de collision du joueur avec les murs :

```
this.physics.add.collider(player, groundLayer);
```

Ajustez la caméra au joueur :

```
// définition de la fenêtre de la caméra
this.cameras.main.setBounds(0, 0, map.widthInPixels, map.heightInPixels);
// la camera suit le joueur
this.cameras.main.startFollow(player);
```

Testez. Vous devez pouvoir déplacer votre personnage dans la map :



Rq : utilisez **player.body.onFloor()** pour le test de la collision du personnage sur le sol

3 Pièces et Pics

Étape 3 :

Déclarez la détection de collision entre le joueur, les pièces et les pics :

```
this.physics.add.overlap(player, coinLayer);  
this.physics.add.overlap(player, picLayer);
```

Définir le lancement des fonctions hitCoin et hitPic à la détection de collision avec les pièces (tuile 26) et les pics (tuile 32) :

```
coinLayer.setTileIndexCallback(26, hitCoin, this);  
picLayer.setTileIndexCallback(32, hitPic, this);
```

```
function hitCoin (sprite, tile)  
{  
    ...  
}  
  
function hitPic (sprite, tile)  
{  
    ...  
}
```

Pour retirer une pièce utilisez :

```
coinLayer.removeTileAt(tile.x, tile.y);
```

Vérifiez que vous pouvez récupérer les pièces :



4 Barre de vie

Étape 4 :

A l'aide d'un rectangle créez une barre de vie en haut à gauche :



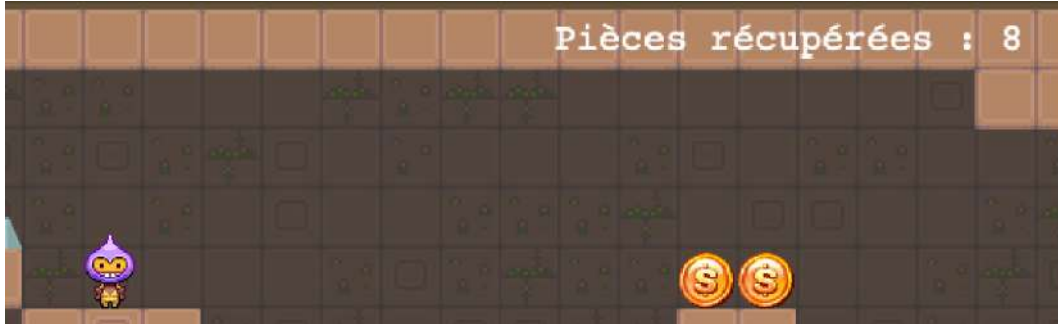
Remarque :

Utilisez `setScrollFactor(0)` pour figer le rectangle en haut à gauche.

A chaque fois que le joueur touche un pic, la barre de vie diminue (utiliser `setSize`) :



Affichez les pièces récupérées en haut en droite :



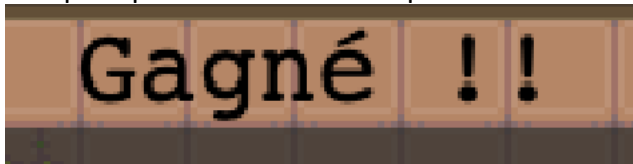
5 Fin du jeu

Étape 5 :

Affichez « Perdu » si vous mourrez avant de récupérer toutes les pièces :



Pour finir le jeu il faut récupérer toutes les pièces. Une fois les 28 pièces récupérées afficher un message pour indiquer qu'il faut trouver une porte cachée. Dès que vous entrez en collision avec cette porte affichez gagné :



Indications :

Utilisez `setTileLocationCallback` et `tile.alpha`

6 Prolongements

Faites jouer votre imagination :

Ajoutez des bruitages, des monstres, un chronomètre ...