

Phaser 3 : contrôles

La [documentation de Phaser](#) est un peu effrayante. C'est parce qu'elle décrit un produit conséquent, entièrement codé en javascript orienté objet ; il y a donc un foisonnement de classes et d'objets disponibles qui défie la compréhension d'un-e dev débutant-e. C'est pour cela que nous allons y chercher ensemble ce dont nous avons besoin : pour apprendre à l'utiliser, et pouvoir mieux comprendre la manière dont on peut exploiter la documentation d'une API (*interface de programmation d'applications*) telle que Phaser.

Gérer les contrôles

Si nous cherchons dans l'API, nous trouvons rapidement un [InputPlugin](#). Il est créé automatiquement lors de l'instanciation `Phaser.Game(config)`. On y accède via l'objet `this.input`. Dans le menu à droite, on peut constater qu'il possède un certain nombre de membres, c'est à dire d'attributs. Certains d'entre eux nous intéressent particulièrement : [keyboard](#), qui est un objet destiné au clavier, et [gamepad](#), pour les manettes.

`this.input.keyboard` est une instance de [KeyboardPlugin](#). La documentation nous confirme que nous n'avons pas besoin de la créer, elle est instanciée par défaut. "You do not need to create this class directly, the Input system will create an instance of it automatically."

Méthodes utiles pour le contrôle clavier

[CreateCursorKeys](#) (utilisé dans le tutoriel) est vraisemblablement un cas particulier de [AddKeys](#), qui permet de gérer davantage de touches du clavier. Il existe aussi [AddKey](#), destiné à contrôler une touche en particulier.

Fonctionnement du gamepad

Il nous faut activer sa prise en charge par Phaser en ajoutant dans les paramètres de configuration l'entrée `input:{gamepad:true}`,

Ensuite, le gamepad doit s'activer une fois le jeu lancé, en appuyant sur une touche. Cette activation est récupérée dans la fonction `update()` par l'instruction suivante :

```
this.input.gamepad.once('connected', function (pad) {
    // 'pad' is a reference to the gamepad that was just connected
});
```

C'est à l'intérieur de ce bloc que l'on peut attribuer [pad](#) à une variable que l'on pourra réutiliser afin d'écouter les contrôles manette. On trouvera pertinent et pratique de se souvenir grâce à un booléen qu'une manette a bien été activée, afin de s'assurer ne pas générer de bug dans le cas où aucune manette n'est connectée ou activée.