

# REJ

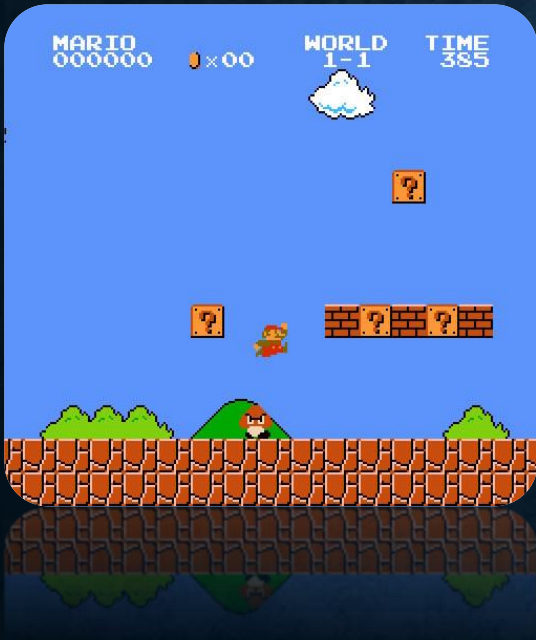
Par Alexis Valois-Adamowicz





# STYLE DE JEU

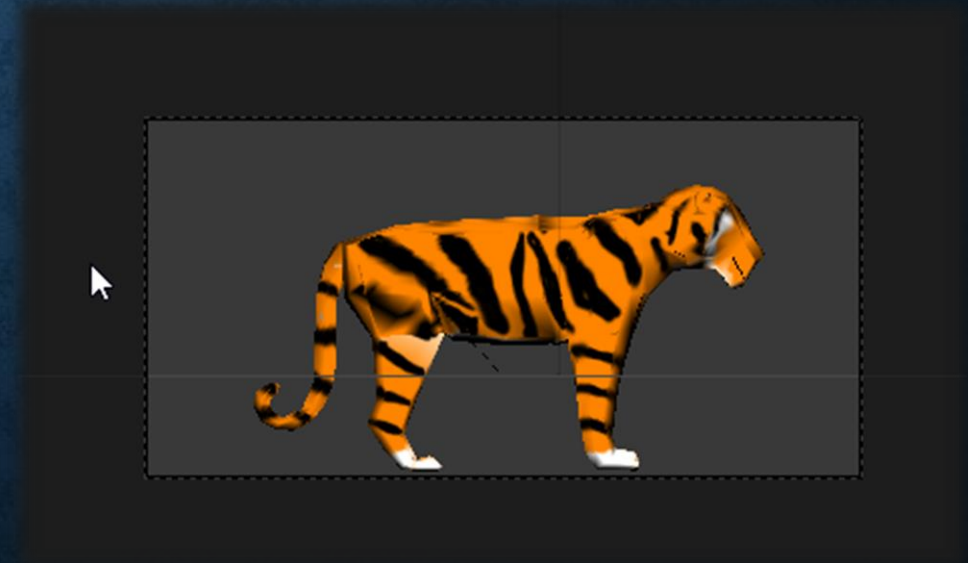
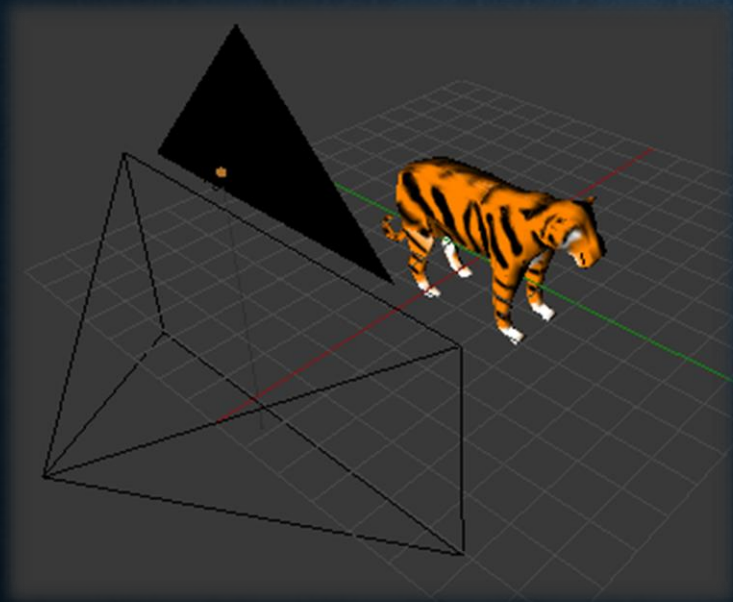
- Jeu de plateforme
- Influences : Super Mario Bros, Donkey Kong Contry, Sonic the Hedgehog





# PROJECTION 2D UTILISÉE

- Orthographique : caméra orthographique placée latéralement
- Parallaxe : utilisation de 6 calques distincts, dont 2 déroulent indépendamment



# AVANT-PLAN





# OBJETS (CAFÉS)



# SOL





# PARALLAXE



# ARRIÈRE-PLAN





# CIEL





# TECHNIQUE DE MODÉLISATION 3D : REJ

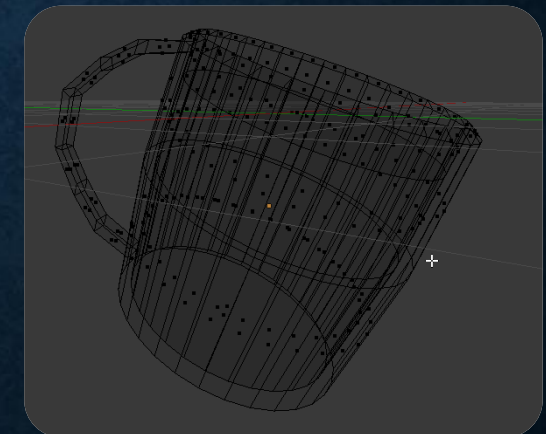
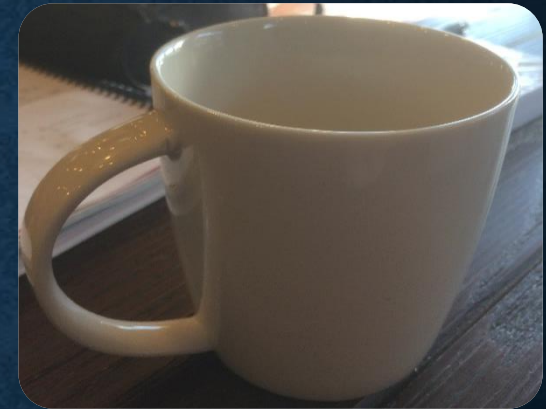
- Utilisation d'une image calque
- Calquage latéral
- Début : cylindre pour faire une patte (arrière)
- Suite d'**extrusions** : faces, arrêtes, sommets
- Créations de faces avec 4 sommets
- Matériel de base : couleur orange, **modèle de Lambert**, diffuse 1, spéculaire 0
- Lumière : Héli, énergie 2.3
- Création d'une map UV + peinture sur le modèle
- Insertion d'une **armature** pour les mouvements
- Source : vidéo YouTube (utilisateur PigArt)



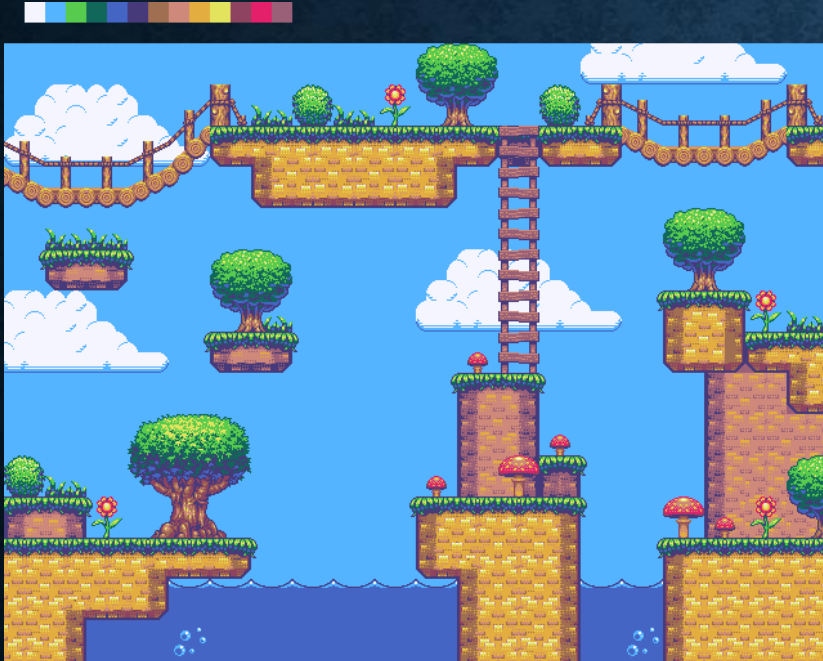


# TECHNIQUE DE MODÉLISATION : TASSE

- Modèle : observation
- Début : cylindre
- **Extrusion** + **mise à l'échelle** pour l'épaisseur
- Ajout d'arrêtes (révolution) pour l'élargissement central
- Utilisation de l'outil « **Knife** » pour créer des faces de contrôle pour les extrusions de la poignée
- Création d'une face pour lui assigner la couleur du café
- Utilisation des « **keyframe** » et de la vue « **dopesheet** » pour l'animation
- Matériaux
  - Poterie : **Modèle de Phong**, Diffuse 1, Spéculaire 1
  - Café : **Modèle de Phong** : Diffuse 0.8, spéculaire 0.5, effet « Distorted Noise »



# UTILISATION DE RESSOURCES LIBRES

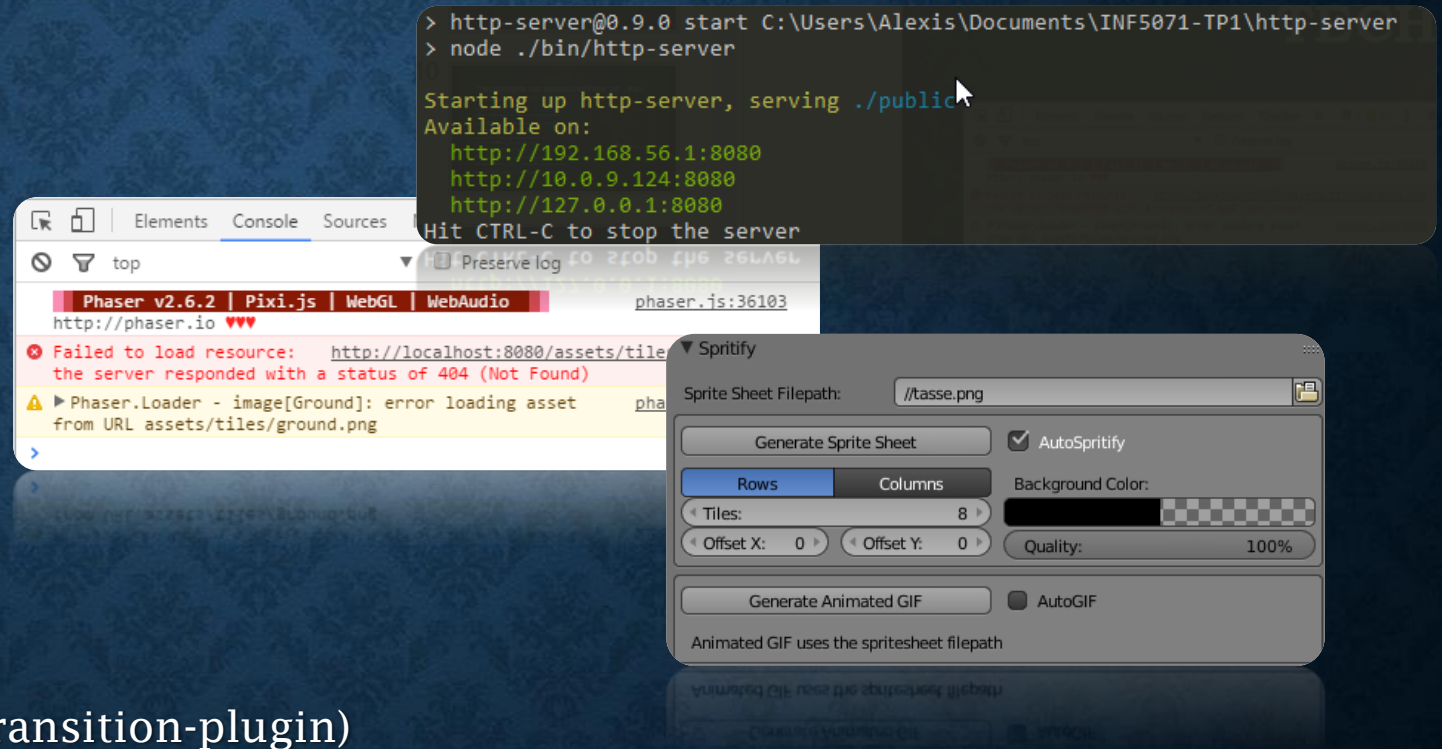


- GenericPlatformer : éléments du décor
- Source : <http://opengameart.org/>
- Licence : Public Domain



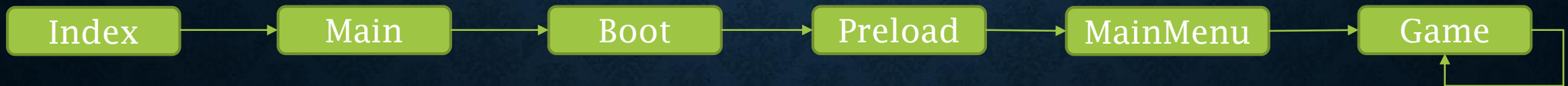
# TECHNOLOGIES UTILISÉES

- Javascript
- Phaser
- Tiled
- Node.JS
- Sublime Text 3
- Google Chrome DevTools
- Git / SourceTree / GitHub
- http-server (JavaScript)
- Npm
- Bower (underscore et phaser-state-transition-plugin)
- Spritify



# ORGANISATION DU TRAVAIL

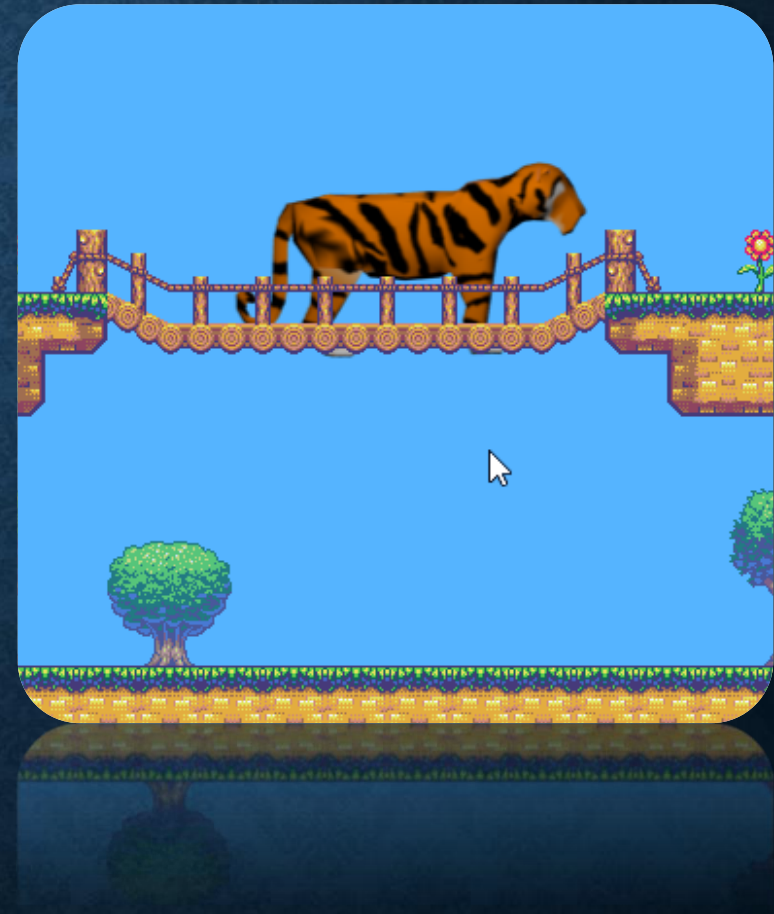
- Un fichier blender par élément modélisé (tasse et Rej)
- Une [spritesheet](#) par éléments animable contenant toutes les animations possibles
- Une map Tiled exportée en JSON
- Code source (Organisation selon tutoriel [gamedevacademy.org](http://gamedevacademy.org))
  - [Index.html](#) : Chargement des dépendances JavaScript et appel du [main.js](#)
  - [Main.js](#) : Chargement de Phaser et définition des « Game State »
  - [Boot.js](#) : Configuration de la physique et des transitions d'états ([Phaser Transition Plugin](#))
  - [Preload.js](#) : Chargement des [images](#), [spritesheet](#) et [map](#) du niveau 1
  - [MainMenu.js](#) : Affichage du menu pour démarrer le niveau 1
  - [Game.js](#) : Logique du niveau 1





# POINTS FORTS

- Projection [parallaxe](#) bien réussie
- Modèle de [Rej](#) respecte bien l'image calque
- Effet d'avant-plan / arrière-plan
- Fonctionnalité d'accélération des cafés
- Réutilisation des images du [spritesheet](#) pour les mouvements vers la droite afin de produire ceux vers la gauche programmatiquement.



## POINTS FORTS (SUITE)

- Technique de « l'escalier » : ne pas bloquer le joueur s'il est en collision avec un palier qui monte légèrement.
- Inspiration : [emanueleferonato.com](http://emanueleferonato.com)
- **getTileWorldXY** : vérifie la présence d'une tuile sur un calque à une position donnée

```
/*
Code simplifié pour illustration
*/
if(player.body.blocked.right &&
    player.body.velocity.x > 0){
    if((
        !map.getTileWorldXY(
            player.x+tileSize,
            rej.player.y-tileSize,
            layer.ground)

        &&
        !map.getTileWorldXY(
            player.x,
            player.y-tileSize,
            layer.ground)
    ))
    {
        rej.player.x += tileSize + 1;
        rej.player.y -= tileSize;
    }
}
/*
Mécanisme similaire pour le côté gauche
*/
}
```



# ILLUSTRATION TECHNIQUE DE « L'ESCALIER »



# POINTS FAIBLES

- Général
  - Il n'y a pas de son ou de musique
  - Il n'y a pas de système de pointage ou d'énergie
  - Il n'y a pas d'ennemis
  - Niveau de difficulté assez faible
- Programmation
  - Boîte de collision des objets pourrait être mieux adaptée
- Modèle (Rej)
  - Pas d'yeux
  - Rayures pourraient être retravaillées ([coutures](#))
  - Pas de système de particules pour créer un effet de fourrure
- Animations (Rej)
  - Manque de réalisme / fluidité
  - Manque de types d'animation

