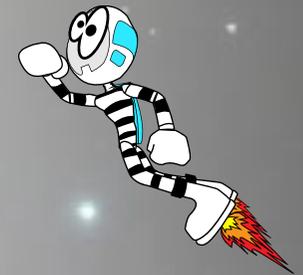
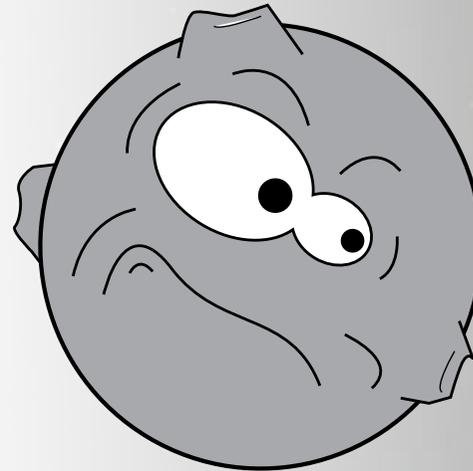


# SPACE JUMPER

## SOMMAIRE

Cadre du projet	p.2
Overview	p.3
Mockup	p.4
Univers	p.5
Histoire et personnage	p.6
Contrôles	p.7
Caméra	p.8
Références	p.9
Gameflow	p.10
Gameplay Loops	p.11
Mécanismes	p.12
Level Design	p.13
Sound Design	p.14
Tuning	p.15



### Demande

*Réaliser un jeu flash jouable avec un seul bouton.*

### One Button Game



*Type : Runner*

*Plateformes : PC/Smartphones*

*Cible : Casual*

### Le Runner à 360°

*Ayez le timing parfait*

*Une vitesse croissante*

*Des planètes de tailles variables*

*Des obstacles à éviter*

### Équipe

#### Graphistes

Sami Daoud  
Anthony Deutsch  
Thomas Boucher

#### Développeur

Grégory Vitte

# SPACE JUMPER

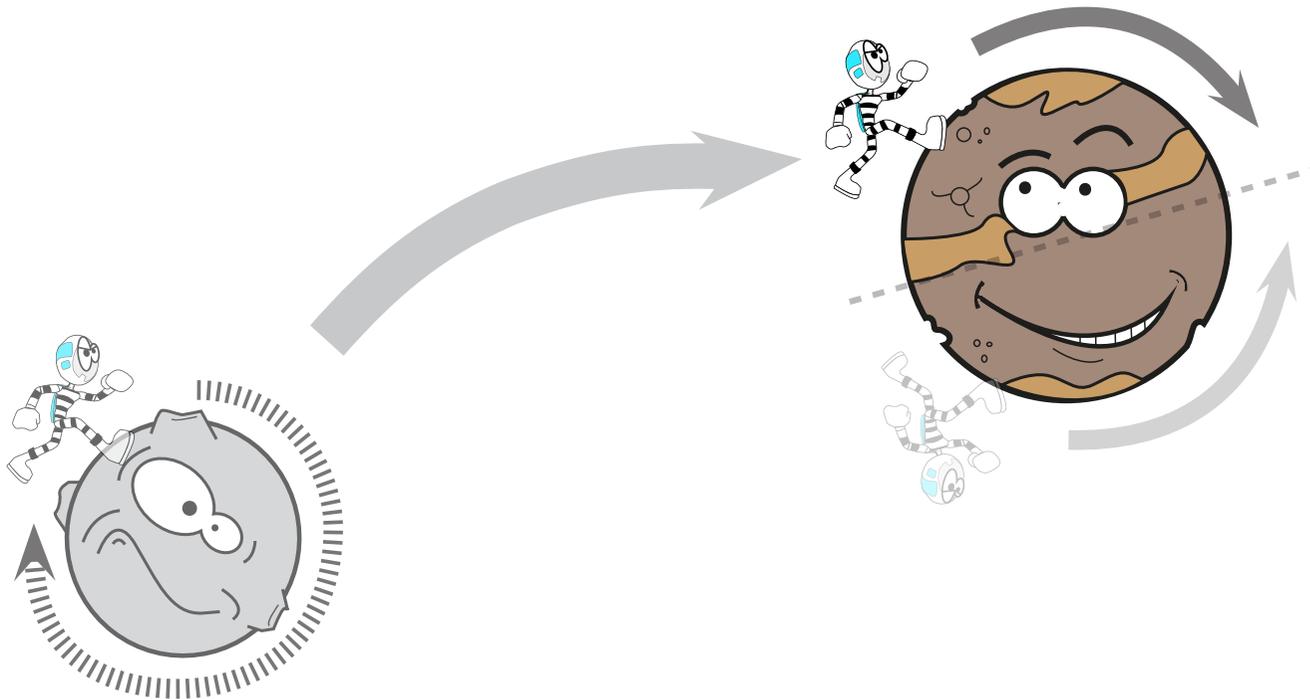
## Overview

**Sautez de planètes en planètes pour aller le plus loin possible !**

*Le joueur peut sauter dans toutes les directions...*

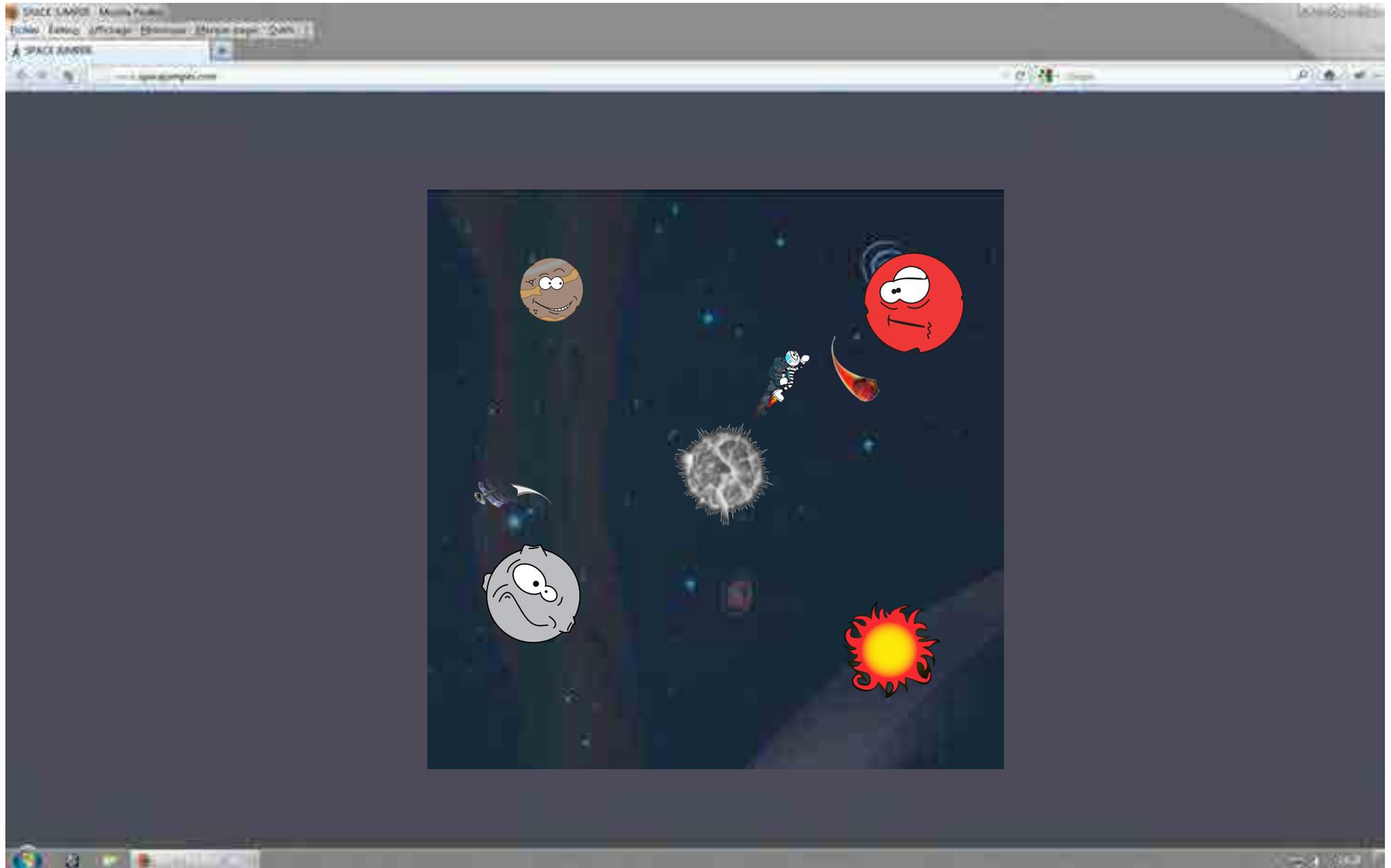
*Mais maîtrisez votre trajectoire, le moindre faux pas et c'est la mort assurée !*

*Mettez vos réflexes à l'épreuve face à une vitesse grandissante.*



# SPACE JUMPER

*Mockup screenshot*



# SPACE JUMPER

## Univers

Lieu : Espace (!)  
Thème : Fantastique, Science fiction  
Univers : Froid (vide spatial), Coloré (planètes)

**L'arrière plan** : composé de deux plan distincts. Un fond plus ou moins uni, sombre, étoilé (avec étoiles générées aléatoirement idéalement). Le deuxième plan est par dessus, composé d'élément spatial et galactique (voie lactée, galaxies lointaines, nébuleuses).

**Les planètes** : On opterait pour des planètes empathiques personnifiées (elles ont des yeux quoi) pour faire varier les humeur/expressions en fonction d'à quelle point elles sont proches d'exploser.

Le fait que le personnage soit un bagnard renommé justifie les sentiments des planètes qui peuvent avoir, à ce moment là, différentes réactions à la vue du prisonnier. Elles peuvent trembler de peur au point d'exploser et d'autres peuvent l'encourager.

Chromatiquement on pensait pouvoir s'inspirer de la bille (de quand on était gosse).

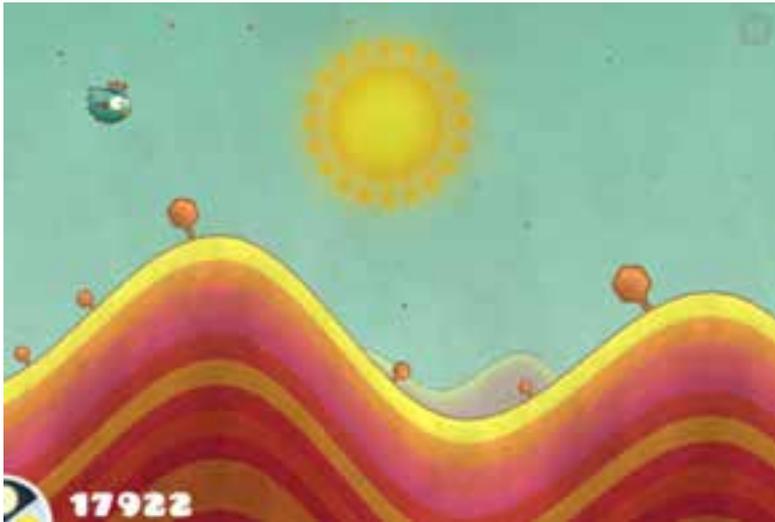
Bruitages indispensables (pas, choc...) pour mettre en avant les interactions.

# SPACE JUMPER

Univers

## Références graphiques

Tiny Wings



Aspect haut en couleurs, fantaisie, clarté

Plants vs Zombies



Couleurs, simplicité

# SPACE JUMPER

## *Histoire et personnage*

John Doe, mieux connu sous le nom de Space Jumper est un explorateur et sportif acharné pour qui l'univers n'est qu'un vaste terrain de jeu.

Grace à ses désormais célèbres chaussures : les Air Strike 2600 lui permettant d'effectuer des bonds extraordinaires de planètes en planètes sur lesquelles il ne reste jamais bien longtemps, il parcourt les moindres recoins de la galaxie.

Accusé d'avoir terrorisé plus d'une centaine de pauvres planètes lors de son passage dans de nombreuses galaxies, il est Jugé coupable de ses crimes et emprisonné dans la plus grande prison interstellaire.

Seulement voilà, rien ne peut arrêter cet explorateur invétéré, pas même les murs d'une prison !

Parvenant à s'échapper, il est contraint de fuir, pourchassé par les polices interstellaires. ? ne voit alors plus aucune limite dans son exploration et est plus déterminé que jamais à aller là où personne n'est jamais allé !

Mais les chemins qui mènent aux confins de la galaxie ne manquent pas d'obstacles...

**Caractéristiques :**

- Maîtrise l'art du saut galactique
- Ne quitte jamais ses Air Strike 2600
- Porte toujours sur lui sa combinaison de prisonnier (il n'a pas le temps de se changer...)
- Personne ne connaît son véritable visage (lui inclut...)
- N'aime pas les vaches galactiques
- Déteste être arrêté en plein élan

**Esthétique du personnage :**

- Athlétique
- Futuriste

# SPACE JUMPER

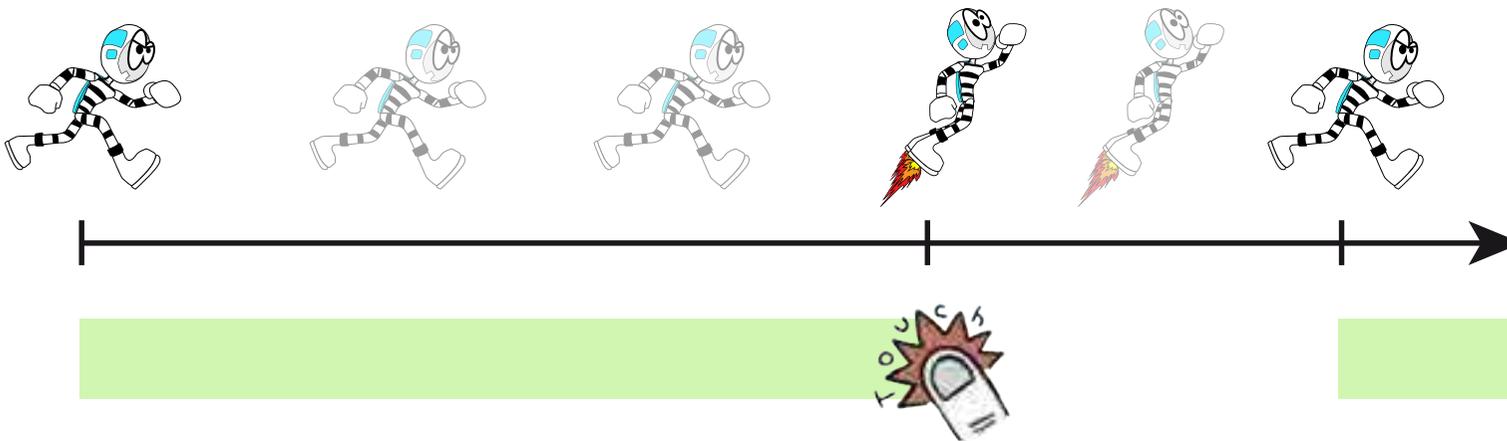
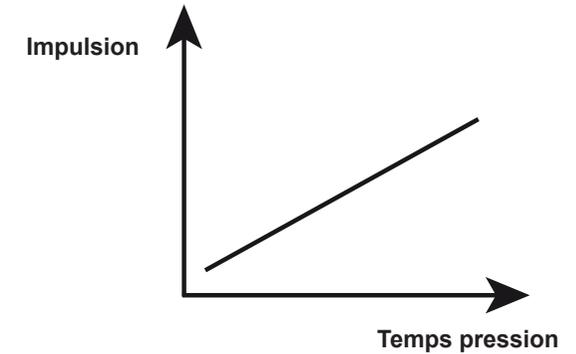
## Contrôles

**Une seule action possible : Sauter !**

Contrainte : être sur une planète

Difficulté : Timing : sauter au bon moment et anticiper la trajectoire de saut.

Temps de pression : en appuyant plus longtemps, le perso saute plus haut.



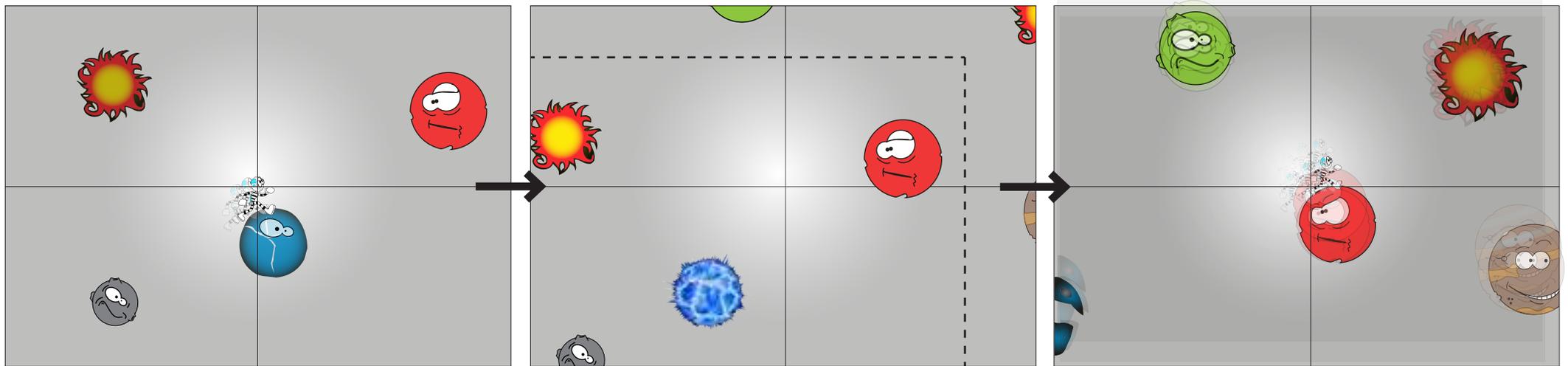
*Feedbacks : sonore + animation du personnage*

# SPACE JUMPER

## Caméra

Lors d'une rotation, la caméra est **centrée** sur le héros (zoom fixe).

Lors d'un saut, la caméra suit le personnage jusqu'à un seuil limite : la taille de l'écran. Passé ce seuil, la caméra ralentit jusqu'à s'immobiliser : les bords de l'écran forment alors une limite infranchissable (Game Over). Un facteur chance entre donc en jeu lors d'un mauvais saut (atteindre une planète sans la voir).



*Cam Shake lors d'une collision*

# SPACE JUMPER

## Références gameplay

Ce type de gameplay existe déjà dans plusieurs jeux iPhone (images cliquables).



**Captain Ludwig**

Référence clé : Gameplay et univers identiques.

*Divergences* : Level Design, objectifs, rythme



**Rocket Bunnies**

Référence clé : Gameplay et univers identiques.

*Divergences* : Level Design, rythme, aide à la visée (le perso atteindra toujours une planète).



**Jump O'Clock**

**Référence clé** : Objectif et gameplay identiques.

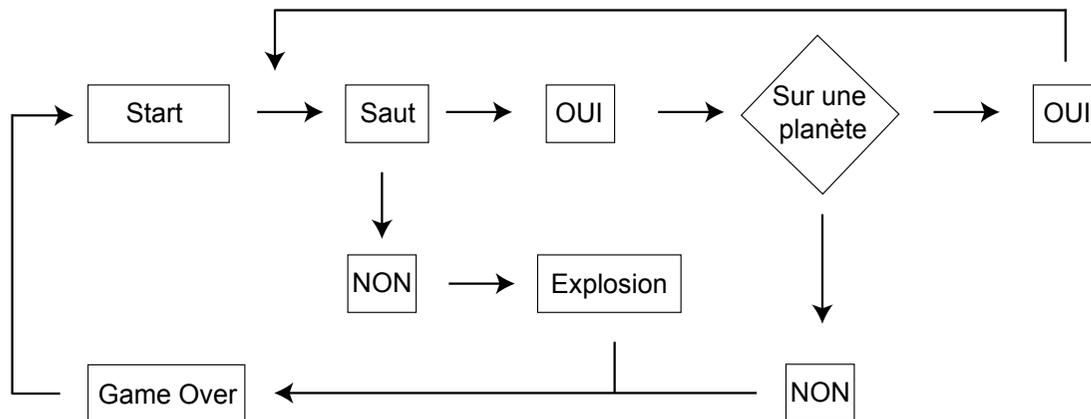
*Divergences* : Verticalité, rythme.

# SPACE JUMPER

## Gameflow

**Space Jumper est basé sur un niveau unique et infini :**

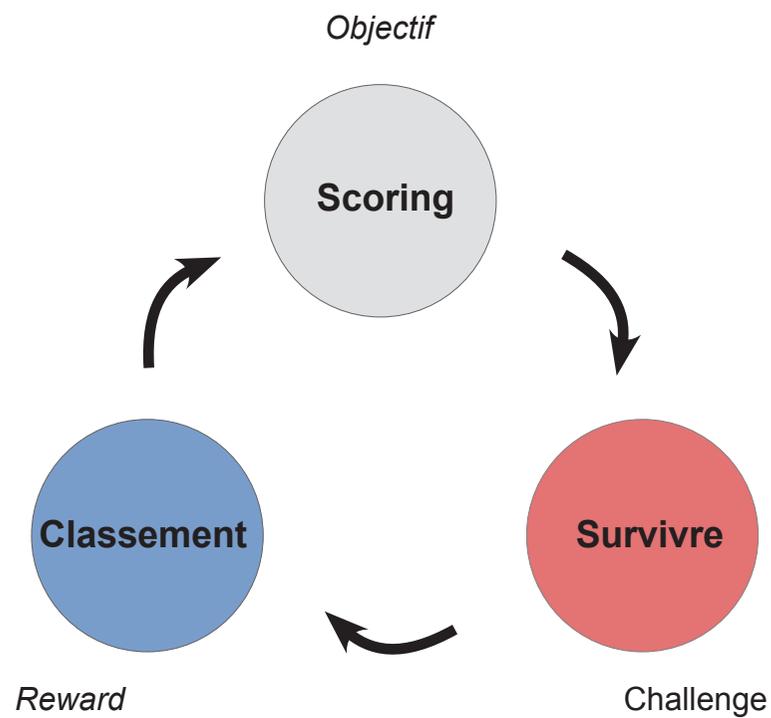
Le joueur ne peut donc pas gagner, pas de sauvegarde possible.  
Par contre, il peut perdre, et vite !



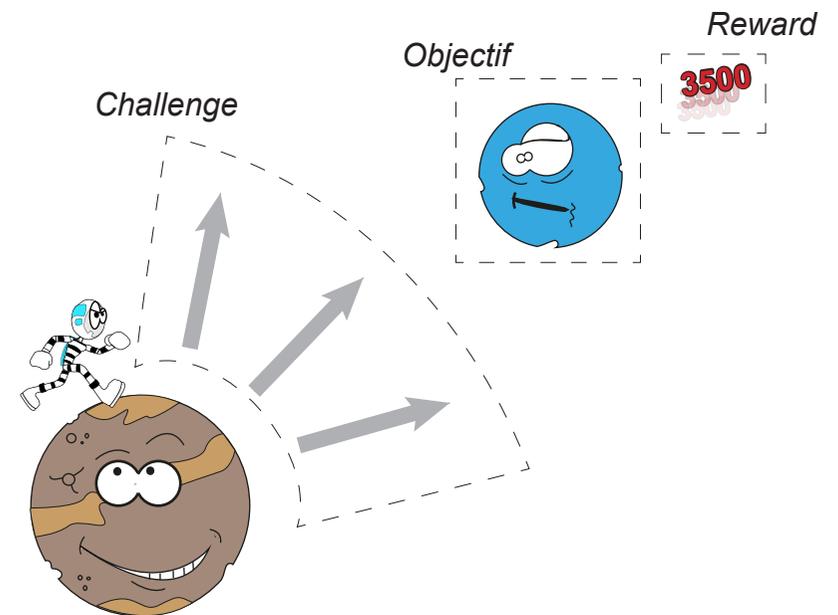
# SPACE JUMPER

## Gameplay Loops

À l'échelle du jeu



À l'échelle d'un input



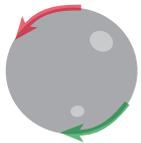
# SPACE JUMPER

## Mécanismes

### 1/ LA VITESSE

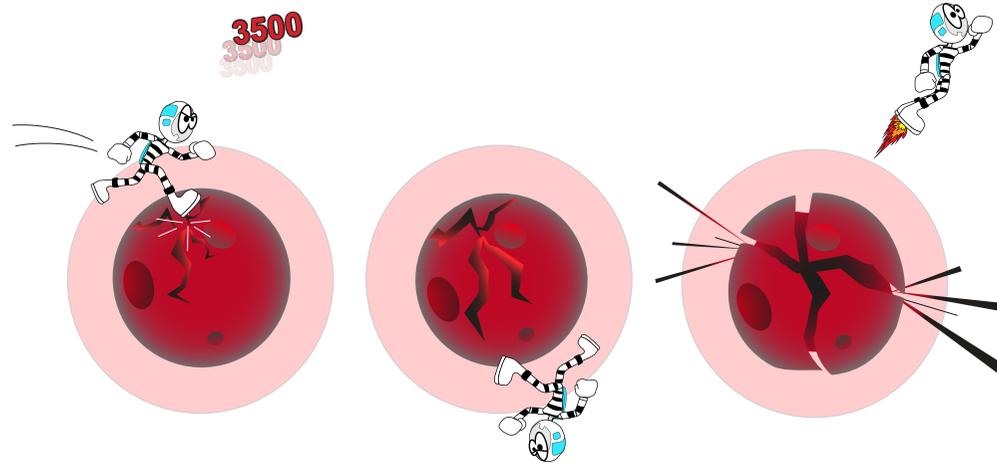


La vitesse est proportionnelle au temps de course.



*Si rotation :  
Ralentissement / Accélération  
du joueur*

### 2/ LES PLANÈTES

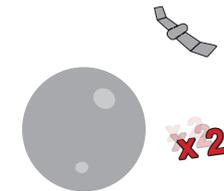


La force engendrée lorsque le joueur arrive sur une planète la rend instable : son explosion à court terme est irremédiable, il ne doit donc pas trop tarder !

Les planètes possèdent un champ de force qui attire le joueur, tel un aimant, avec plus ou moins d'effet.

Par ailleurs, dès qu'il quitte le **champ de gravité** de l'astre, celui-ci explose immédiatement !

Compte tenu de la difficulté d'accéder à une planète qui présente un satellite, un bonus de points est accordé au joueur : le score est doublé, pour ne pas rendre le bonus dérisoire et inciter le joueur à prendre des risques (cf. jeu du lapin qui saute sur des cloches).

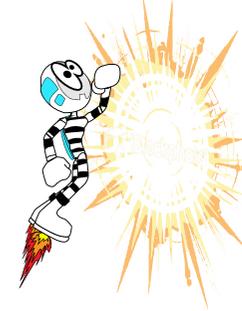
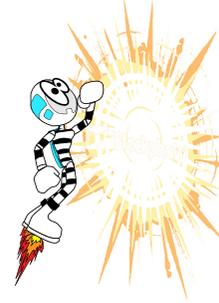
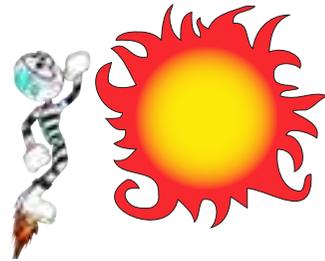


*Planète avec obstacles :*  
**score doublés**

# SPACE JUMPER

## Mécanismes

### 3/ LES OBSTACLES



Toucher un obstacle est synonyme de game over, aucune erreur n'est permise.

Il sont de diverses formes :

- les soleils réduisent en poussière le héro

- les obstacles explosent

Proposition de mécanismes :

Des trous noirs. Qui ressemblent à des planètes mais qui sont en fait piège. Des planètes pièges.

Des planètes géantes?

# SPACE JUMPER

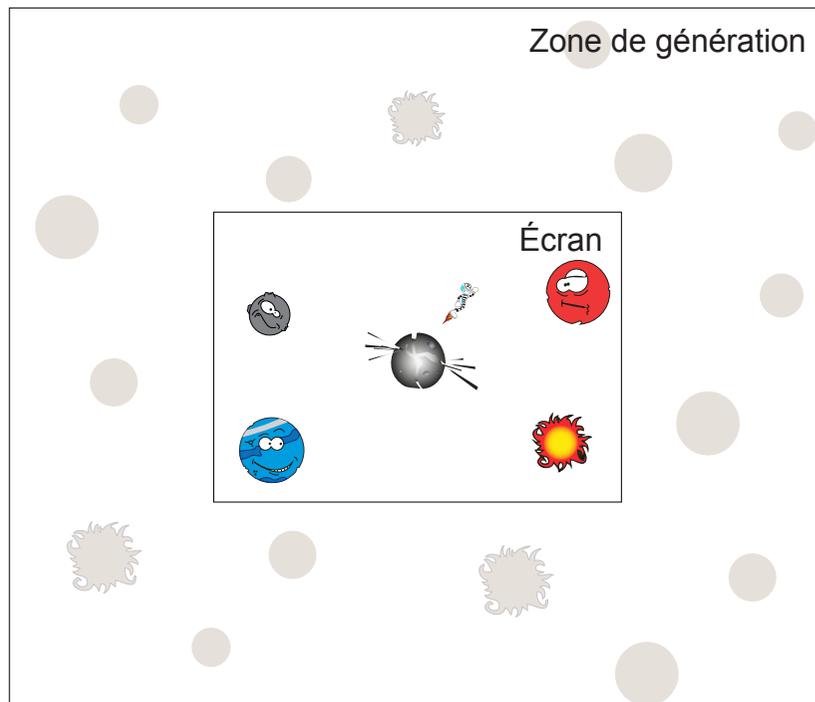
## Level Design

Space Jumper offre **un seul niveau** : il est infini. Une partie peut donc durer de quelques secondes à plusieurs minutes.

Les éléments sont **générés aléatoirement** en dehors de la zone visible (dite zone de génération).

Une planète détruite est remplacée (en tenant compte de la progression du joueur et de l'emplacement des autres planètes proches) dès lors que son emplacement se trouve dans la zone de génération.

Le jeu est basé sur un **apprentissage par erreurs**. Le joueur qui débute aura donc tendance à mourir très vite avant d'appréhender tous les mécanismes.



**Rythme** imposé par la disposition aléatoire des planètes et obstacles.

**Contraintes** de création liées à la caméra :

Les planètes ne doivent pas être trop grosses ou trop petites.  
Elles ne doivent pas être trop proches ou trop éloignées.

# SPACE JUMPER

## Sound Design

### Intentions

L'idée serait d'amplifier la progression du gameplay (le runner va de plus en plus vite) grâce au son.

La musique électro synthétique va en crescendo tout au long de la partie

Les bruitages organiques (souffle pour le saut par exemple) composent et rythment principalement l'environnement sonore en début de partie. Au début la musique les accompagne discrètement et s'intensifie en rythme et en éléments sonores, toujours en complément des bruitages qui, eux, resteront les mêmes tout le long de la partie.

Le jeu pourrait être un jeu musical (façon Rez/ Child Of Eden) ? Plus on joue, plus on compose ?  
Le son pourrait remplacer les éléments de bonus un peu trop classiques dans le jeu vidéo ?

### Pôles d'assets sonores

- Sound Design : Sons de feedback
- Bruit de pas : L'idée est d'éviter des bruits de pas classiques.  
Ces bruits sont illustrés par des éléments musicaux. Ces éléments s'accordent à l'humeur de la planète "traversée".  
ex :
  - Dans Rayman Origins, niveau "Désert des Didgeridoos" : musique créée quand Rayman marche sur un serpent géant
  - bruits de tambour
  - voix "Ah", "ah", "AAh" (demander à Anthony!)
- Bruit de saut : Idée d'une respiration, voire d'une résonance, d'un élan.
- Bruit d'atterrissage
- Bruit de collision
- Sons d'interface et HUD : Bouton start = cri de joie (SJ s'évade) ? criant "LIBRE !!"  
= bruit métallique (les barreaux) ?  
= bip bip tout simplement ?



# SPACE JUMPER

## Tuning

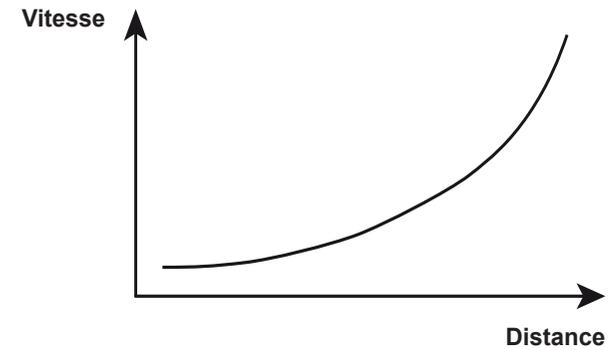
### I/ LA VITESSE

La vitesse est proportionnelle au temps de course.

$$\text{VitesseMin} \leq \text{Vitesse} \leq \text{VitesseMax}$$

Le joueur n'accélère pas en plein saut.

Le joueur retrouve sa vitesse initiale après un passage sur une planète qui tourne ?



# SPACE JUMPER

## Tuning

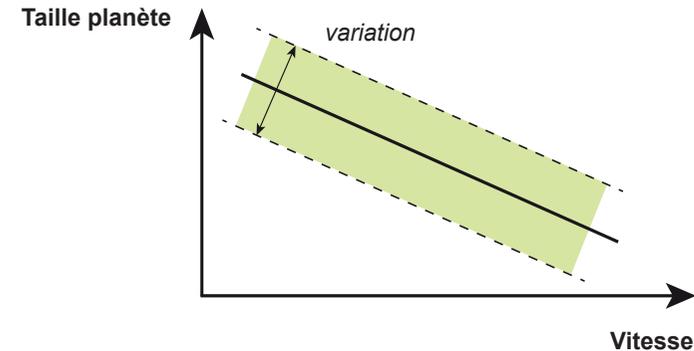
### 2/ LES PLANETES

La résistance d'une planète est proportionnelle à sa taille.  
Une grosse planète mettra donc plus de temps avant d'exploser.

Minimum 3 planètes doivent être présentes à l'écran, pour ne pas forcer le joueur à suivre un itinéraire.

Leur distance les unes par rapport aux autres augmente, mais ne peut dépasser une certaines limites (taille de l'écran divisée par 2 max).

Le nombre de planètes à rotation augmente.



# SPACE JUMPER

## Tuning

### 3/ LES OBSTACLES

Le nombre d'obstacles  $X$  est compris entre deux valeurs :

$$X_{\min} = 0$$

$$X_{\max} = (\text{NbPlanètesMin}) - 1$$

$$X_{\min} \leq X \leq X_{\max}$$

Il ne peut donc pas y avoir d'obstacles sur toutes les planètes visibles à l'écran, mais il peut y en avoir aucun.

Cette probabilité baisse au cours du temps.

