# Projet

## Modalités

Le but de ce projet est de concevoir un *jeu de catapulte* (de type *AngryBirds* , mais plus modeste) en HTML5 + Javascript. Le projet est décomposé en une partie basique, notée sur 14 points et des fonctionnalités supplémentaires permettant d'arriver jusqu'à 20 points. Une soutenance, elle aussi notée sur 20 points sera faite à l'issue du projet. La note finale sera calculée avec un coefficient de 60% pour le rendu et 40% pour la soutenance.

Le projet (code + rapport) doit être rendu au plus tard le :

dimanche 29 avril 2018, 20h00

Le rapport (format PDF) et le code doivent être regroupé dans une archive (.tar.gz uniquement) et déposée via un formulaire de soumission sur la page du cours. Les soutenances auront lieu le 3 mai 2018.

Le projet est à faire en binôme, merci de contacter le responsable du cours si vous n'avez pas de binôme au préférez faire le projet en monôme.

## Spécifications

On souhaite concevoir un jeu de catapulte similaire au célèbre jeu *AngryBirds*. Dans un tel jeu, un dispositif de tir, contrôlé par le joueur au moyen du clavier ou de la souris permet de projeter des objets. Ces derniers suivent une trajectoire parabolique par défaut. Le but est d'utiliser les projectils pour toucher une ou plusieurs cibles. Cette dernière peut être abritée derrière des éléments de reliefs. Ces derniers peuvent êtres indestructibles ou friables.

**Représentation du monde** Le plus simple est de représenter le « monde » comme un ensemble d'objets physique régis par des lois simples (collisions, gravité, elasticité, cf. TP2).

## Fonctionnalités de base (14 points)

On demande que les aspects suivants soient respectés :

- le joueur doit pouvoir trier des projectils avec une force variable (en utilisant soit la souris, soit le temps de pression au clavier, ...)
- le niveau doit comporter au moins une cible
- la cible doit être protégée par des objets
- le niveau ne doit pas être hard-codé mais doit être représenté de manière *générique* au moyen d'un fichier JSON dont vous spécifierez proprement le format, et chargé dynamiquement
- votre jeu doit implémenter des mouvements réalistes (trajectoire paraboliques par exemple)
- le rendu doit être fait dans un élément *canvas*
- le rendu doit au minimum être de 30 FPS (images par secondes), idéalement 60 (vous devez afficher ces informations de débuggage dans la page)

## Fonctionnalités avancées (2 points chacune)

Attention, les fonctionnalités avancées ne **rapportent des points que si tous les aspects des fonctionnalité de bases sont implémentés**. Exemple un projet qui ne charge pas ses niveaux dynamiquement obtiendra une note maximale de 14 (et probablement moins) même si toutes les fonctionnalités ci-dessous sont implémentées.

**Objets friables** On souhaites que vous implémentiez des objets *friables*, c'est à dire qui disparaissent après un nombre fini de collision avec un projectil ou un autre objet.

**Blessures** les dégats occasionnés à la cible ne doivent pas être du « tout ou rien » mais être proportionnel à la force d'impact

**Cibles ou obstacles mobiles** le niveau possède des cibles ou obstacles mobiles (selon un trajet prédéfini)

**Jeu complet** le jeu doit être composé de plusieurs niveaux, s'enchaînant les uns avec les autres et d'un écran d'accueil. Une logique de jeu doit aussi être présente (score pour chaque niveau, défit à remplir par niveau, par exmple détruire toutes les cibles avec moins de $x$ projectiles)

**Éléments de reliefs** certains éléments du reliefs doivent avoir des propriétés physiques particulières (par exemple agir comme des ressors pour augmenter la vitesse des projectiles lors de rebonds ou au contraire absorber de l'énergie)

**Projectiles multiples** Implémenter des projectiles aux propriétés diverses (qui se séparent, plus lours, plus légers, avec une trajectoire particulière, ...)

Tout autre suggestion est la bienvenue **mais** doit être validée avant par le responsable de cours. Implémenter plus de 3 fonctionnalités ne rapporte pas plus de 6 points.

## Conseils

Il est fortement conseillé de reprendre le TP2 du cours (moteur physique) et de l'adapter plutôt que de repartir de zéro. On prendra garde à rajouter de *l'hystérésis*. En effet, si le mouvement d'un objet descends en dessous d'une certaine constante, on le considère comme nul. Sans cela des objets censés êtres immobiles peuvent vibrer et le mouvement s'amplifier.

Afin d'obtenir des bonnes performances on peut séparer les éléments mobiles de l'image de fond (ou des animations de fond). On pourra donc avoir deux canvas superposés, l'un contenant le joueur (et les autres éléments mobiles le cas échéant) et un fond transparent et l'autre les éléments de décors.

Le lien ci-dessous sera probablement utiles :

— `http://opengameart.org` (un site contenant de graphiques réutilisables librement pour les blocs et les personnages)

## Rendu

Vous devez rendre en fin de projet :

— un rapport succinct mais complet décrivant la répartition du travail, les choix d'implémentation, la documentation utilisée, le code réutilisé et la manière dont il a été modifié, la spécification de vos formats de niveaux et des exemples judicieux de code complexe que vous auriez pu écrire. L'organisation du code (quel fichier/objet fait quoi) doit aussi y être décrite

— votre code judicieusement commenté

Vous aurez aussi une soutenance de 10 minutes pour faire la démonstration de votre code. Il vous est rappelé que le plagiat est formellement interdit, tout code qui ne serait pas de vous doit donc être explicitement marqué comme tel. Si une trop grosse proportion de code n'est pas de vous, la note s'en ressentira évidemment (avec une exception pour le moteur de rendu, dont vous avez écrit une partie en TP et dont le corrigé est disponible).