INCOMING!!!

Par

Samuel DALLAIRE

Vincent BOILY

Travail présenté à

M. Michel Payette

Dans le cadre du cours Projet intégrateur

420-KMA-JQ

Session hiver 2017



**Intro**

Dans l’optique du cours de projet de quatrième session où il est question d’accomplir un certain travail de recherche/développement sur sujet reliée à un des divers domaines de science. Parmi ceux-ci et après mûre réflexion, nous avons choisi le domaine de physique. La raison est que nos connaissances dans les autres domaines soit la biologie et la chimie sont soit nulles ou bien très minime pour l’instant. De plus, nous nous sentons tout simplement plus alaise à travailler avec les formules et théorèmes de physique que nous avons côtoyés pendant 3 sessions.

**Sujet**

Parmi les milliers de cas où les principes de physiques s’appliquent et qui sont d’une certaine façon « humainement » programmable, la balistique s’est démarqué. En effet, les calculs et réactions réalisés sur divers objets projetés par divers engins comme la catapulte sont facilement observables lors de test. De plus, dans le cas où nous aurions terminé d’implémenter ce principe, nous serions pleinement en mesure d’explorer la possibilité d’ajouter d’autres notions tel que : le magnétisme, les contacts élastiques/non-élastique, la force hydraulique, etc.

Moi et mon coéquipier tout de même une tout autre raison pour choisir ce sujet spécifique. Effectivement, bien qu’étudiant de science, nous sommes des programmeurs avides d’améliorer nos compétences par rapport au code et nous allons utiliser ce projet à notre avantage. En addition de faire recherche des recherches sur ce sujet de physique nous allons produire un programme qui nous permettras de nous dépasser et d’assouvir notre soif de défi de codage.

**Projet**

Nous souhaitons réaliser un logiciel de combat entre deux engins de sièges se déroulant sur plusieurs ordinateurs.

**Variables et notions**

Commençons par regarder les détails physiques. Parmi les divers engins de sièges on associe les principes généraux de tension, torsion, contrepoids, rotation et conception qui donnerons l’angle et l’énergie de départ d’un projectile. Ensuite ce projectile seras constamment soumis à la force gravitationnelle de l’astre et affronteras différents facteurs comme l’aérodynamique et la force ce du vent. Ceux-ci viendra jouer sur ces paramètres de vitesse, d’énergie potentiel et cinétiques pour arriver à un impact final où il y aura collision et peut-être rebond.

Du côté du code informatique nous devrons comprendre certaines notions de serveurs et d’inter-connectivités.



-Trébuchet datant de l’ère médiéval au château de Castelnaud

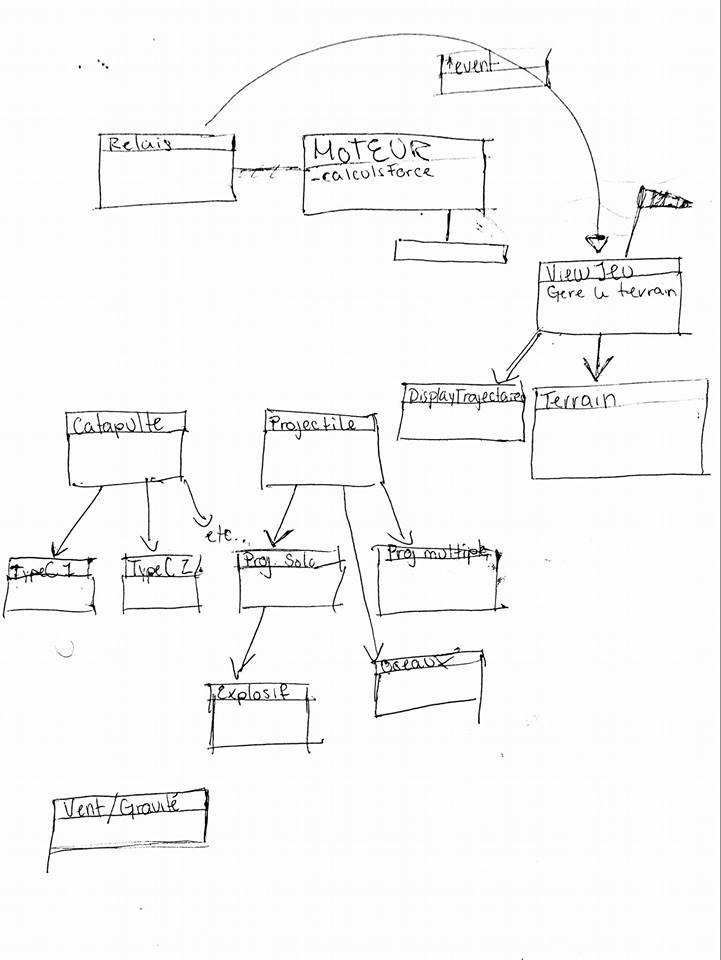
**Résultat Attendu**

Vision du projet :

Pour notre projet final, nous visons à combiner les éléments interactifs du jeu avec les notions théoriques de la physique. En utilisant les différents aspects théoriques de la physique, nous envisageons simuler des trajectoires de projectiles de manière fidèle à la réalité.   
  
Nous nous attendons à un projet simple mais très bien rempli, qui fonctionnera de manière efficace et sans superflu, pour que tous les éléments du programme soient utilisables à leur plein potentiel. Au travers de différents éléments interactifs ajoutés au logiciel, l’utilisateur pourra visualiser de manière précise les différents systèmes en jeu dans la trajectoire d’un projectile.

Le programme sera également, pour nous, une grande étape dans notre progrès en tant que programmeurs. Nous comptons y inclure un aspect de réseautique concret, en faisant fonctionner le programme sur une multitude de machines à la fois, toutes et chacune synchronisées. Pour nous, cet aspect du jeu nous permettra de rendre le tout plus attirant pour l’utilisateur, tout en renforçant nos compétences en programmation.

Nous avons une bonne idée de ce que nous voulons concevoir, sans y être 100% fixés. Nous sommes prêts à adapter notre conception du projet, dans le cas où certains de nos objectifs sont superflus, ou encore irréalisables. Nous espérons par contre y inclure tous les éléments planifiés, et même plus encore, pour pouvoir être vraiment satisfaits de notre effort.

****

**-Brouillon de diagramme de classe**

**Problèmes**

Commençons par la bête noire de notre programme : la gestion de la communication entre les machines. Non seulement est-ce quelque chose de quasi-inconnu pour nous, au niveau programmation, c’est également un immense nid de problèmes éventuels. Évidemment, nous croyons être à la hauteur de la tâche, mais reste que le potentiel problématique de cet aspect est élevé.

Ensuite, la gestion de collisions. Bien que nous ayons de l’expérience dans ce domaine, la gestion de collisions précise reste un défi considérable, sans être effrayant. Nous avons par contre des exemples passés de gestion de collision, ce qui pourra nous aider au cours du projet.

Finalement, le troisième défi de notre projet se trouvera au niveau de l’animation. Nous voulons un projet bien fini et attirant, et cela inclut des animations de bon niveau. Il est important dans ce type de projet de représenter de manière fidèle les différents évènements qui se produisent au cours de la partie, et pour cela, le rendu doit être fluide et sans ambiguïté. Cela se révèlera sans doute être un bon défi, même si nous avons un peu d’expérience dans le domaine.

**Médiagraphie**

# SÉGUIN Marc. *Physique XXI T.A Mécanique*, ERPI, avril 2010

# Java™. *Java™ Platform, Standard Edition 7*. [En ligne]. <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>

# Java™. *Java™ Platform, Standard Edition 8*. [En ligne]. <https://docs.oracle.com/javase/8/javafx/api/overview-summary.html>