Émile Grégoire  
Simon-Pierre Deschênes  
Jérémie Bolduc  
Jonathan Samson

Projet d’intégration en sciences  
informatiques et mathématiques

420-204-RE

**Dossier de conception**

Travail présenté à

M. Jocelyn Goulet

Département d’informatique Cégep Limoilou – Campus de Québec

Le 11 février 2015

**Table des matières**

[Description du projet 2](#_Toc410719825)

[Objectifs 2](#_Toc410719826)

[Description détaillée 2](#_Toc410719827)

[Rôles et justifications 2](#_Toc410719828)

[Type d’application 2](#_Toc410719829)

[Technologies impliquées 2](#_Toc410719830)

[Prototypes de l’application 3](#_Toc410719831)

[Scénarios 4](#_Toc410719832)

[Diagramme de classes 5](#_Toc410719833)

[Échéancier 5](#_Toc410719834)

# Description du projet

## Objectifs

Notre projet illustrera certains phénomènes physiques observés en état d’apesanteur. Les constituants principaux seront les planètes, les astéroïdes et un vaisseau spatial. Afin de rendre l’expérience plus ludique, le programme se présentera comme étant un mini-jeu. Ainsi, l’utilisateur aura à atteindre certains objectifs pour progresser.

## Description détaillée

Puisque notre application simulera les déplacements spatiaux, de nombreuses lois de la physique mécanique seront mises de l’avant. Notamment, la gravitation newtonienne sera nécessaire. En ce sens, nous utiliserons la formule suivante pour calculer la force exercée par les astres sur le vaisseau :

De plus, il sera question des sommes des forces influençant le vaisseau provenant des différentes planètes du système pour que sa direction soit jugée comme réelle. Le mouvement du vaisseau sera calculé grâce à la cinématique. Pour ce qui est de la réussite des niveaux (scénario), il y aura une petite jauge à essence qui affichera la quantité d’essence restante dans le vaisseau. Préférablement, le trajet se voudra assez court pour que la difficulté du niveau soit acceptable. Ainsi, il ne sera pas question d’avoir des trajets trop faciles ou impossibles.

## Rôles et justifications

|  |  |  |
| --- | --- | --- |
| **Nom** | **Rôle** | **Justification** |
| Émile Grégoire | Chef d’équipe | Leader positif |
| Simon-Pierre Deschênes | Responsable de la qualité | Perfectionniste |
| Jonathan Samson | Responsable des livrables | Autonome et ponctuel |
| Jérémie Bolduc | Responsable des réunions | Esprit de synthèse et responsable |

## Type d’application

Logiciel récréatif disponible sur toutes les plateformes compatibles avec Java. Peut s’exécuter en mode fenêtré et en plein écran.

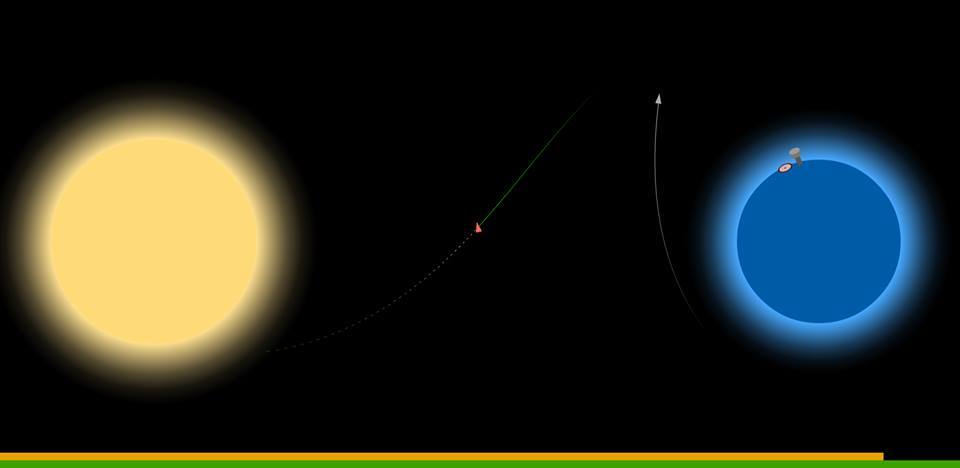
## Technologies impliquées

* Java 8
* FXML
* CSS
* XML
* Git (BitBucket)
* Eclipse
  + Plugin ObjectAid
  + Plugin e(fx)clipse
  + Plugin JUnit
  + Plugin EclEmma
* SceneBuilder 2.0
* Microsoft Project 2013

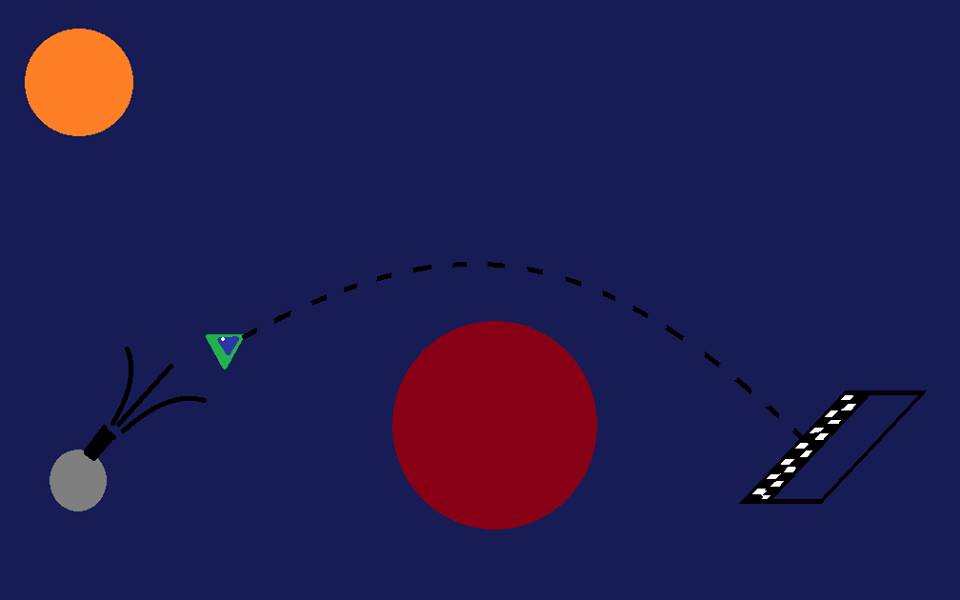
# Prototypes de l’application

L’image suivante est tirée d’une application existante qui a nourri notre idée originale. Elle est disponible ici :

<http://zhaop.me/grav/>



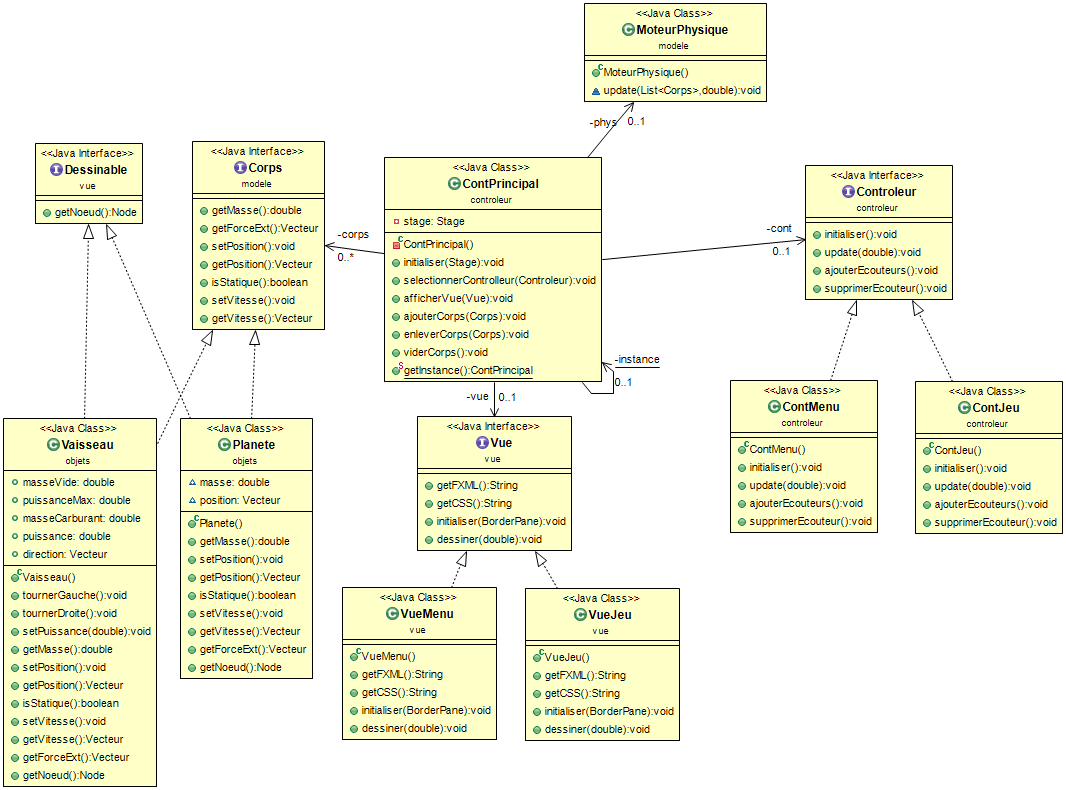
L’image suivante est un prototype de notre application. Elle illustre la poussée initiale et l’objectif final qui seront présents dans notre application.



# Scénarios

|  |  |
| --- | --- |
| **Scénario #1** | |
| Acteur | Équipe |
| Scénario | En tant qu’équipe, nous devons uniformiser l’environnement de travail afin de s’assurer de la compatibilité entre les membres de l’équipe. |
| Description | 1. S’assurer de l’utilisation d’Eclipse 2. Installer tous les plug-ins nécessaires (énumérés ci-dessus) 3. Installer SceneBuilder 2.0 4. Installer TortoiseGit 5. Créer le référentiel sur BitBucket 6. S’assurer que tous les membres de l’équipe comprennent l’utilisation de TortoiseGit |
| Tests d’acceptation | Vérifier l’usage des différents outils.  Tester à l’aide d’un commit et d’un push. |
| Complexité | 1 |
| Effort | 1 |
| Commentaires |  |
| **Scénario #2** | |
| Acteur | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que les éléments du jeu interagissent ensembles comme le dicte les lois de la physique afin que le jeu soit réaliste. |
| Description | 1. Définir les caractéristiques physiques des éléments du jeu 2. Implémenter les lois de la physique |
| Tests d’acceptation | Placer plusieurs éléments près les uns des autres et vérifier que les lois de la physique sont respectées. |
| Complexité | 5 |
| Effort | 3 |
| Commentaires |  |
| **Scénario #3** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux qu’une fenêtre et les éléments du jeu s’affichent lorsque j’exécute le programme afin de pouvoir voir ce qui se passe. |
| Description | 1. Générer les fichiers FXML à l’aide de SceneBuilder 2. À chaque rafraîchissement de l’écran, récupérer les éléments qui s’affichent à l’écran 3. Dessiner les éléments dans la zone de dessin |
| Tests d’acceptation | Ouvrir le programme et vérifier que tout s’affiche correctement. |
| Complexité | 5 |
| Effort | 3 |
| Commentaires |  |

# Diagramme de classes



# Échéancier