Jérémie Bolduc  
Simon-Pierre Deschênes  
Émile Grégoire  
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](#_Toc410904089)

[Objectifs 2](#_Toc410904090)

[Description détaillée 2](#_Toc410904091)

[Rôles et justifications 3](#_Toc410904092)

[Type d’application 3](#_Toc410904093)

[Technologies impliquées et langages utilisés 3](#_Toc410904094)

[Prototypes de l’application 4](#_Toc410904095)

[Scénarios 6](#_Toc410904096)

[Diagramme de classes 12](#_Toc410904097)

[Échéancier 13](#_Toc410904098)

# Description du projet

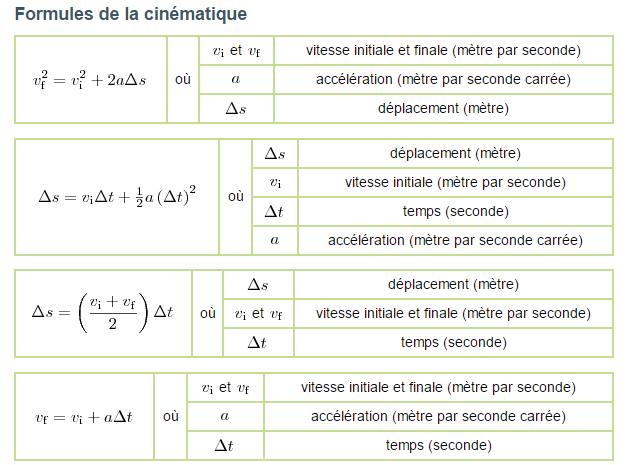
## Objectifs

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

* 1. La Physique

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[[1]](#footnote-1) :

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[[2]](#footnote-2). La cinématique est l’étude des mouvements qui incluent la notion du temps. En ce sens, les formules de la cinématique comprenant le temps, la vitesse, la position de l’objet et son accélération seront indispensables. Elles seront au cœur de notre moteur physique.

[[3]](#footnote-3)

1.2 Conception des niveaux

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, voir la section des schémas dans ce même dossier de conception. Le but des concepteurs est de faire en sorte que les niveaux soient réalisables selon une trajectoire restreinte et calculée. En ce sens, l’utilisateur devra gérer sérieusement ses effectifs de carburant et devra aussi regarder par quel chemin il n’aura pas le choix de passer. 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. Par ailleurs, une jauge de santé sera utilisée pour faire en sorte que les collisions avec les autres objets du système soient justifiables. Donc, plus le nombre de collisions sera élevé, plus la jauge de santé s’affaiblira.

* 1. Interface et contrôles

Les contrôles seront situés sur le clavier de l’utilisateur. Ce seront probablement les flèches du pavé de jeu qui seront utilisées pour définir la direction du vaisseau. D’autres touches comme la lettre R pourront être pressées pour recommencer la partie. Pour ce qui est de l’interface, un écran d’accueil s’occupera de guider l’utilisateur à travers l’application. De cet écran, il pourra choisir de commencer au niveau 1, commencer à un niveau x, faire son propre niveau et quitter le jeu.

## 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. L’application est un jeu de type arcade, comportant de nombreux petits jeux.

## Technologies impliquées et langages utilisés

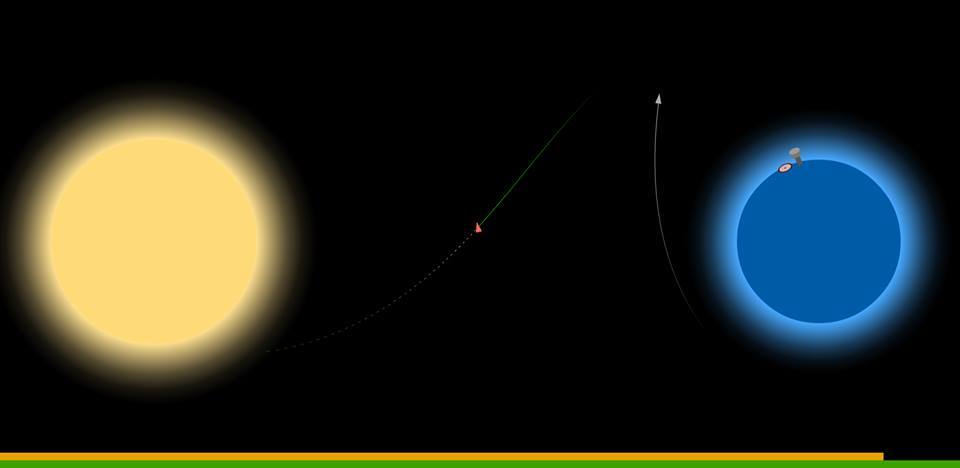
* Java 8
* FXML
* CSS
* XML
* Git (BitBucket)
* Eclipse
  + Plugin ObjectAid
  + Plugin e(fx)clipse
  + Plugin JUnit
  + Plugin EclEmma
* SceneBuilder 2.0
* Microsoft Project 2013

# 

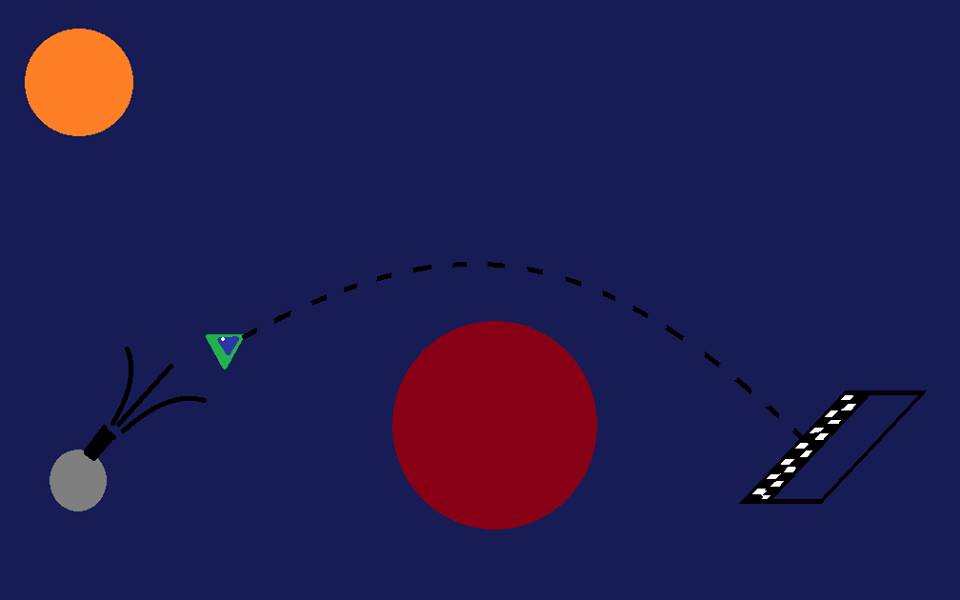
# Prototypes de l’application

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

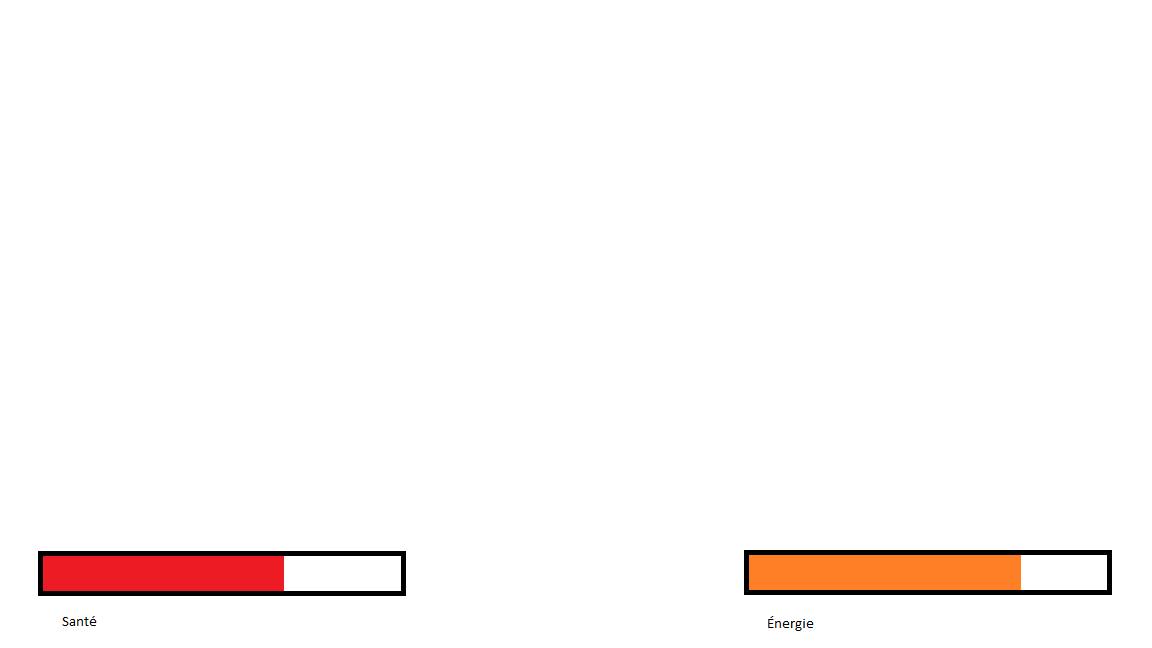
<http://zhaop.me/grav/>



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



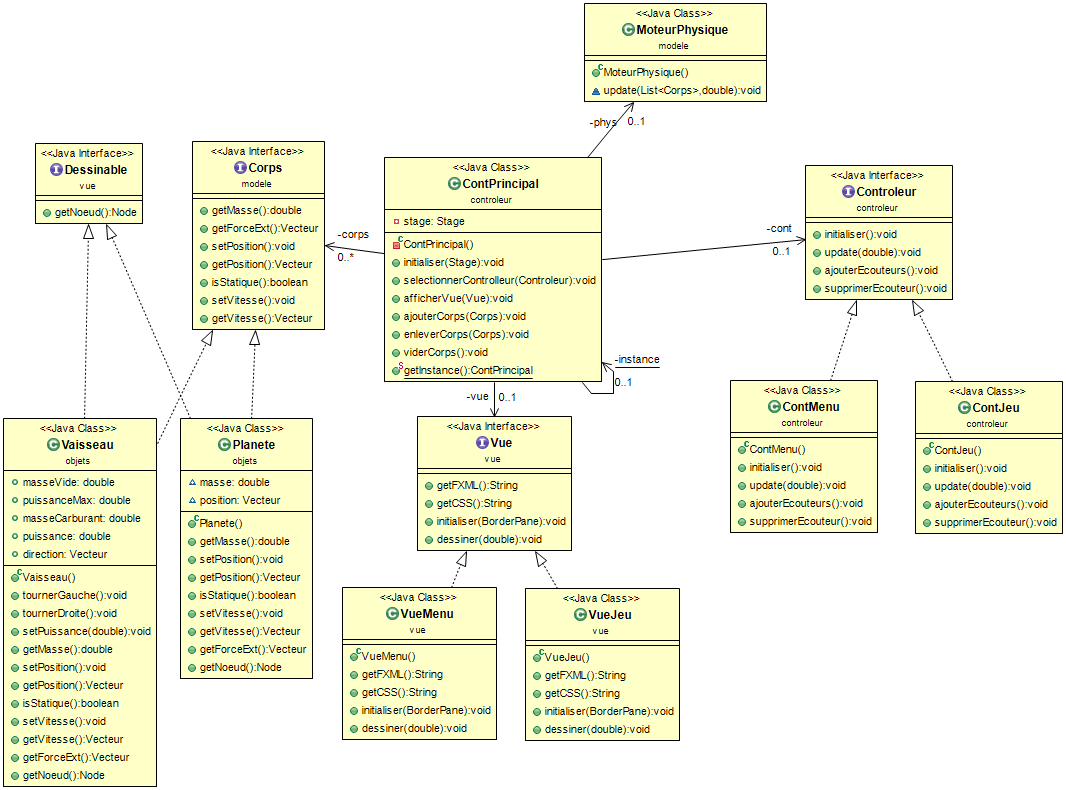
L’image suivante illustre l’interface utilisateur qui sera présentée au joueur au bas de l’écran.



# 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. Configurer le plug-in Git sur Eclipse 5. Créer le référentiel sur BitBucket 6. S’assurer que tous les membres de l’équipe comprennent l’utilisation de Git |
| 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 | Le SVN sera mis à jour à chaque fin de sprint. |
| **Scénario #2** | |
| Acteur | Développeurs |
| Scénario | En tant que développeur, je veux pouvoir changer de contrôleur et de vue facilement afin de simplifier les tests sur le modèle physique. |
| Description | 1. Créer un contrôleur principal qui contiendra les vues et les contrôleurs. 2. Implémenter les changements entre les contrôleurs. 3. Implémenter le chargement des vues. |
| Tests d’acceptation | Créer deux contrôleurs de test et de vue pour s’assurer que les changements s’opèrent bien. |
| Complexité | 3 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #3** | |
| Acteurs | Développeurs |
| Scénario | En tant que développeur, je veux pouvoir me servir d’une horloge interne qui me retournera des variations de temps utiles dans les formules de physique. |
| Description | 1. Écrire un thread qui calcule la différence de temps entre un temps x et y. 2. Dans le contrôleur principal, démarrer le thread. 3. Accéder aux valeurs de variations de temps. |
| Tests d’acceptation | Demander d’afficher le nombre de secondes ou minutes qui se sont écoulées entre deux appels de la méthode. |
| Complexité | 2 |
| Effort | 1 |
| Commentaires |  |
| **Scénario #4** | |
| Acteurs | Développeurs |
| Scénario | En tant que développeur, je veux utiliser une classe Vecteur afin d’illustrer les mouvements entre les différents corps. |
| Description | 1. Implémenter l’addition de vecteurs. 2. Implémenter la soustraction de vecteurs. 3. Implémenter la multiplication de vecteurs par des scalaires. 4. Implémenter le produit scalaire de vecteurs. 5. Implémenter le produit vectoriel. 6. Implémenter la normalisation de vecteurs. 7. Implémenter le calcul des modules de vecteurs. |
| Tests d’acceptation | Effectuer différentes opérations vectorielles et vérifier la cohérence des résultats via des tests unitaires. |
| Complexité | 1 |
| Effort | 1 |
| Commentaires |  |
| **Scénario #5** | |
| Acteurs | Développeurs |
| Scénario | En tant que développeur, je veux pouvoir travailler avec des corps physiques qui seront soit statiques, soit en mouvement. |
| Description | 1. Créer l’interface Corps qui sera utilisée par les vaisseaux et les planètes. 2. Créer la classe Planète qui implémente l’interface Corps. 3. Créer la classe Vaisseau qui implémente l’interface Corps. |
| Tests d’acceptation | Vérifier que les valeurs retournées ont du sens. |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #6** | |
| Acteurs | Développeurs |
| Scénario | En tant que développeur, je veux que les différents corps puissent s’attirer entre eux afin que le jeu soit réaliste physiquement. |
| Description | 1. Récupérer tous les corps. 2. Analyser les propriétés des différents corps. 3. Pour chaque corps, calculer la force résultante exercée par les autres corps. |
| Tests d’acceptation | Mettre plusieurs corps à proximité et vérifier que leur comportement est réaliste grâce à des calculs fait à la main. |
| Complexité | 3 |
| Effort | 3 |
| Commentaires |  |
| **Scénario #7** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir voir les différents corps afin de pouvoir voir ce qui se passe. |
| Description | 1. Charger la vue. 2. Récupérer les différents nœuds à afficher. 3. Afficher les nœuds. |
| Tests d’acceptation | Démarrer le programme et vérifier que la fenêtre s’affiche. |
| Complexité | 2 |
| Effort | 1 |
| Commentaires |  |
| **Scénario #8** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir mettre le jeu en pause afin de pouvoir réfléchir. |
| Description | 1. Mettre l’horloge interne sur pause. 2. Créer la classe ContrôleurMenu. 3. Créer la classe VueMenu. 4. Charger le contrôleur. 5. Charger la vue. |
| Tests d’acceptation | Démarrer le programme, appuyer sur pause et s’assurer que le menu pause fonctionne. |
| Complexité | 2 |
| Effort | 1 |
| Commentaires |  |
| **Scénario #9** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que le vaisseau ne puisse pas passer au travers des planètes. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 2 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #10** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir créer mes propres niveaux. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 3 |
| Effort | 3 |
| Commentaires |  |
| **Scénario #11** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que la caméra suive mon vaisseau. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 2 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #12** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux diriger mon vaisseau avec la souris |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #13** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir recommencer le jeu à n’importe quel moment. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #14** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir sauvegarder mon progrès au jeu. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #15** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que les graphiques de fonds soient esthétiquement beaux. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #16** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que les graphiques des vaisseaux soient esthétiquement beeaux. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #17** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que les graphiques des planètes soient esthétiquement beaux. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #18** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux diriger mon vaisseau avec la souris |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #19** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que mon vaisseau se déplace avec une jauge d’énergie. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #20** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que mon vaisseau ait une santé que l’on peut affaiblir. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #21** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux que le tableau se finisse par une ligne d’arrivée. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #22** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux diriger mon vaisseau ait une vitesse initiale et un canon pour le projeter. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |
| **Scénario #23** | |
| Acteurs | Utilisateur |
| Scénario | En tant qu’utilisateur, je veux pouvoir changer de niveau à l’aide d’une interface différente à celle du départ. |
| Description |  |
| Tests d’acceptation |  |
| Complexité | 1 |
| Effort | 2 |
| Commentaires |  |

# Diagramme de classes



# Échéancier



Pour une meilleure vue de l’échéancier, le fichier Microsoft Project est disponible dans le SVN du projet.

1. Wikipédia, l’encyclopédie libre. (Page consultée le 7 février 2015), *Gravitation*, [En Ligne]. Adresse URL : http://fr.wikipedia.org/wiki/Gravitation [↑](#footnote-ref-1)
2. Wikipédia, l’encyclopédie libre. (Page consultée le 7 février 2015), *Cinématique*, [En Ligne]. Adresse URL : http://fr.wikipedia.org/wiki/Cinématique [↑](#footnote-ref-2)
3. Edusofad. (Page consultée le 7 février 2015), *Étude du mouvement*, [En Ligne]. Adresse URL : http://edusofad.com/www/demo/wphy-534/demo/form1\_1a.php [↑](#footnote-ref-3)