Projet réalisé par

CORLAIS Florian, JACQUEMIN Anthony

I. Présentation du logiciel

Le projet ***Physics*** a été développé dans le cadre de l'option *Programmation Fonctionnelle II* de la licence Maths de l'université de Nice Sophia-Antipolis, au cours du second semestre de l'année universitaire.

Il a pour principale vocation de faire découvrir, ou redécouvrir, à son utilisateur les principes de base de mécanique, par le biais d'une documentation détaillée, et d'exemples ludiques, ou plus conventionnels.

Le projet se décompose en deux grandes parties :

- Le logiciel
- Le slideshow explicatif

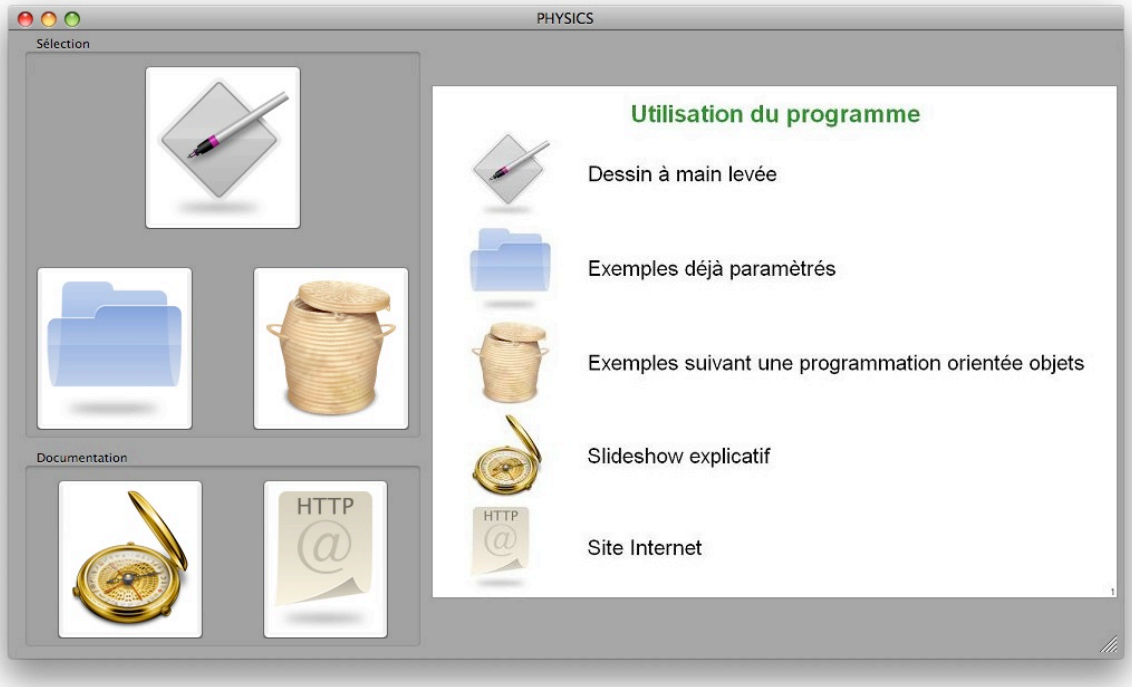
Une version pdf de la présentation accompagne également le projet, rendant une impression du document possible : elle comporte de multiples liens internet illustrant les différentes notions présentées.

Un site internet est disponible. On y trouvera des exemples vidéos de démonstration du logiciel, des mises à jour régulières, l'intégralité de l'aide en ligne, et il sera accessible via l'url suivante :

http://projetphysics.teria.org/

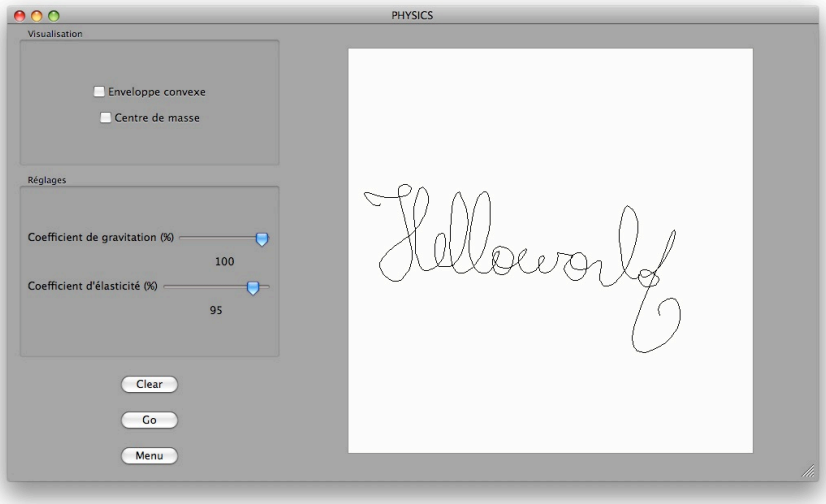
Ce dossier donnera un aperçu des différentes possibilités offertes par notre logiciel, les difficultés rencontrées lors de son développement, ainsi que les limitations imposées pour son bon fonctionnement.

II. Le logiciel



L'accès au logiciel se fait par le fichier ***main.scm***, après exécution du fichier (dans le language *"Assez Gros Scheme"*), la fenêtre principale s'affiche, offrant cinq possibilités :

- **Dessin à main levée** : permet de dessiner un polygone quelconque, qu'il est possible de mettre en mouvement en attrapant son centre de masse. Un clic sur **Go**, et la figure est soumise au champ de pesanteur. Il est possible d'afficher ou non l'enveloppe convexe et le centre de masse du polygone. Pour dessiner une nouvelle figure, une pression sur **Clear** permet de vider la fenêtre d'affichage.

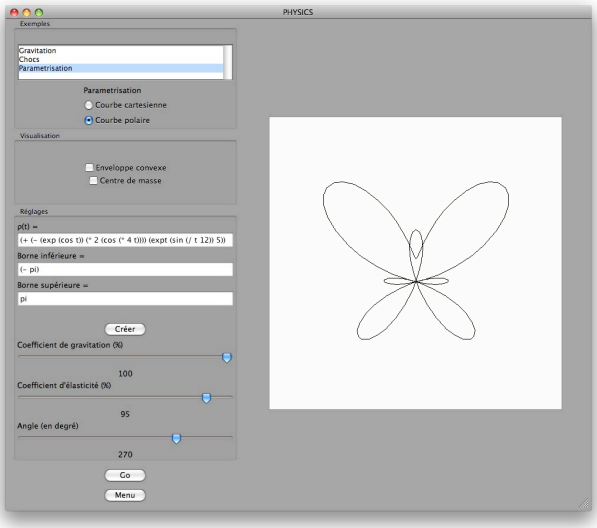


- **Exemples paramétrés** : permet d'observer de nombreux exemples déjà paramétrés, avec des formes plus ou moins exotiques (un escargot, une fleur, un ballon de rugby…). Ils se décomposent en trois parties :

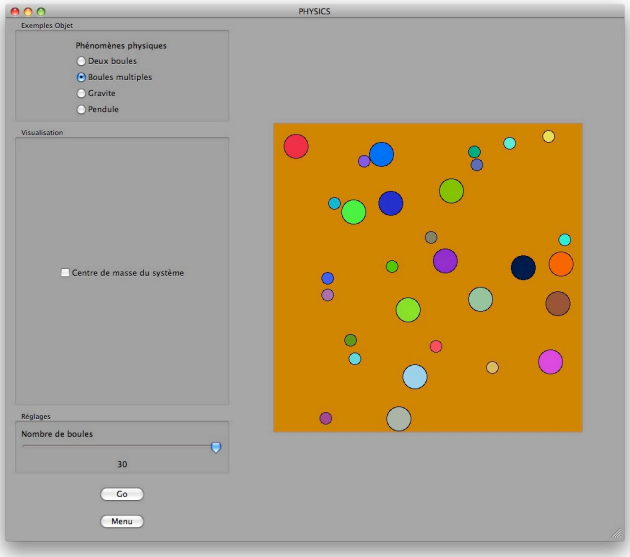
  - Gravitation : un seul polygone dans la fenêtre, soumis uniquement au champ de gravité.

  - Chocs : traite le cas de la collision entre deux convexes.

  - Parametrisation : permet de tracer des courbes paramétrées (cartésiennes et polaires) en entrant les informations les caractérisant par des formules scheme (préfixées).

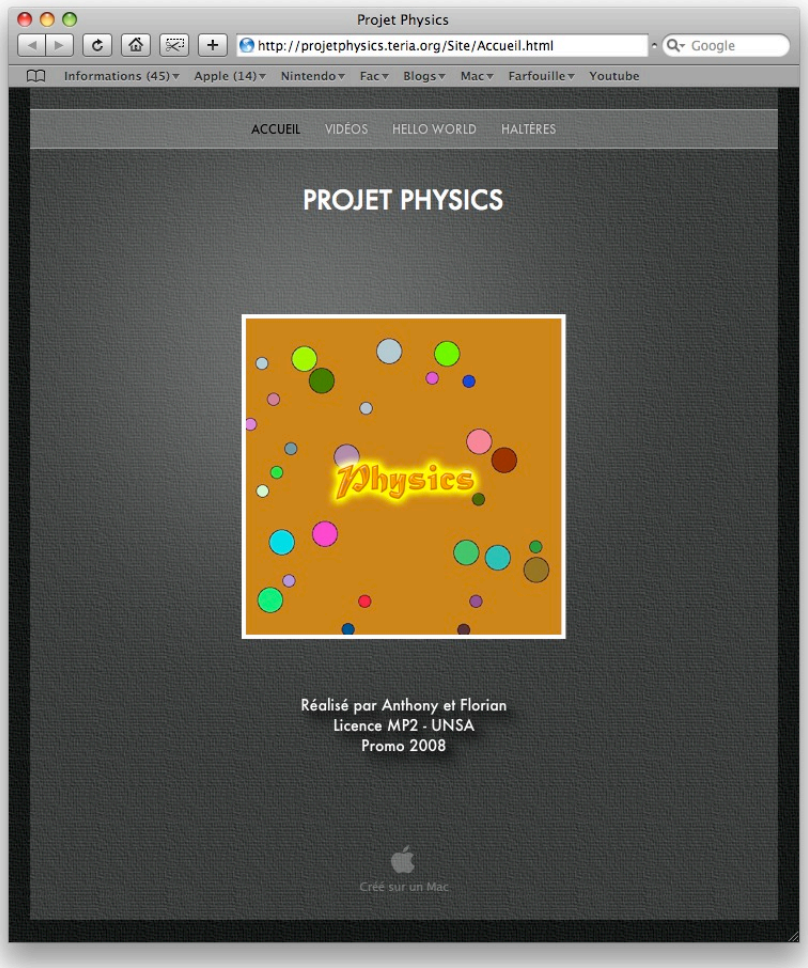


- **Exemples suivant une POO** : cette partie regroupe de nombreux exemples de cours implémentés avec des objets : on y retrouve les classiques *Collision entre deux boules*, *Gravité, Boules multiples*, *Pendule de Newton.*



- **Slideshow** : invite l'utilisateur à regarder la documentation complète du projet via la présentation écrite en Slideshow.

- **Site internet**: ouvre notre site internet dans le navigateur par défaut de l'ordinateur de l'utilisateur.



A tout moment, il est possible de revenir à la fenêtre principale par simple clic sur le bouton **Menu**, de naviguer entre les différentes parties via le menu ***Fichier*** de la barre d'outils, et de modifier les couleurs de fond et de tracé des objets via ***Affichage***, toujours dans la barre d'outils. Enfin, l'utilisateur peut requérir de l'aide par le menu ***Aide*** dédié.

III. Le slideshow

Pour ouvrir la présentation slideshow, il suffit d'exécuter le fichier ***slideshow.scm*** dans le language "*module…*". La navigation à travers les slides se fait par les touches multi-directionnelles du clavier, ou par simple clic au cours de la présentation.

La présentation est interactive :

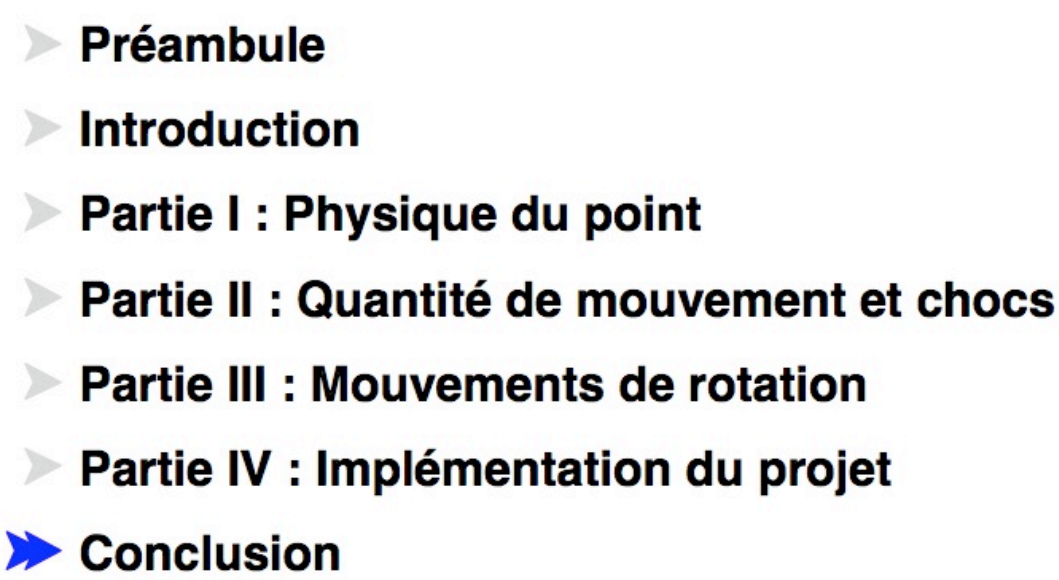
- **Liens scheme** : ces liens sont bleutés et soulignés. Ils permettent, lorsqu'ils sont activés par le lecteur, de lancer une démonstration (un programme scheme) illustrant le point développé au cours du slideshow. Ils sont de la forme :

celui-ci

- **Liens internet** : ces liens pointent vers une page internet : un clic, et celle-ci s'affiche dans le navigateur de l'utilisateur. Ils sont de la forme :

url

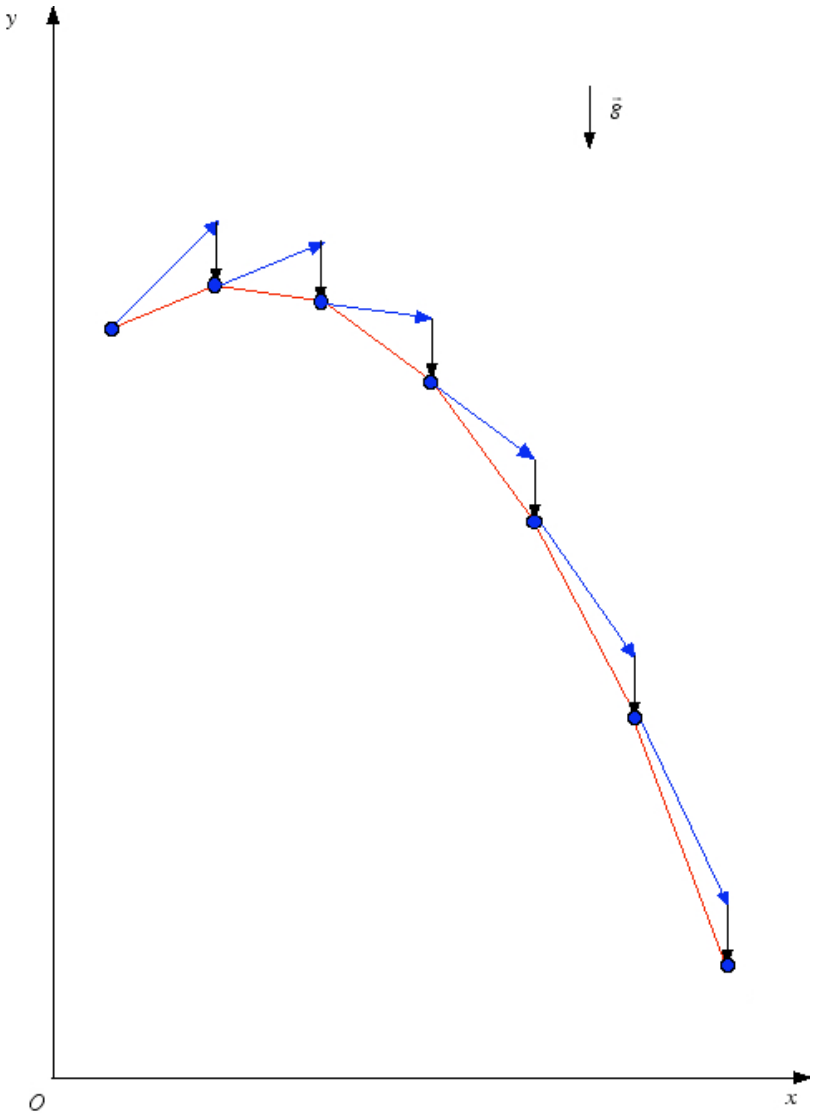
Ce document se décompose en 7 parties :



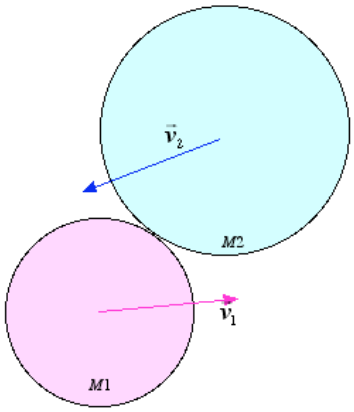
- **<u>Préambule</u>** : cette première partie donne quelques conseils d'utilisation pour la lecture du document, et fait une brève présentation du projet ***Physics***.

- **<u>Introduction</u>** : cette seconde partie est une introduction à la mécanique : une rétrospective des outils de mesure du temps et des longueurs permet de saisir l'importance qu'a eu cette science au cours de l'Histoire de l'Homme.

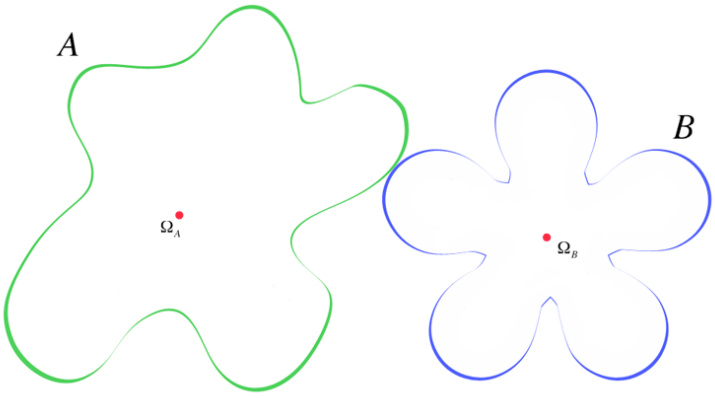
- **<u>Physique du point</u>** : Cette partie pose les définitions élémentaires utiles à l'étude du mouvement de masses ponctuelles : on définit la vitesse, la quantité de mouvement, et le théorème fondamental de la mécanique. On en déduit les trois lois de Newton, le théorème de composition des vitesses, et l'on arrive finalement à *construire* le trajet d'un point en chute libre par l'algorithme de *Hook-Newton*.



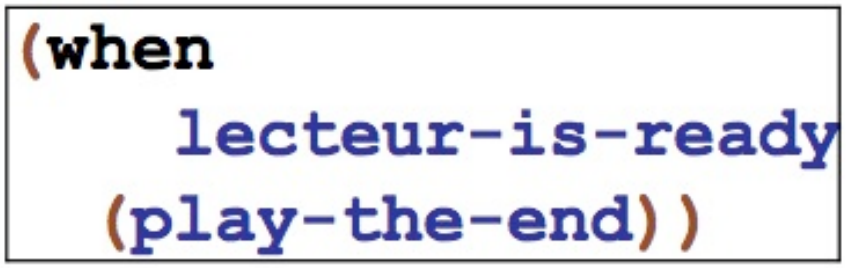
- **<u>Quantité de mouvement et chocs</u>** : On souhaite résoudre le problème à deux dimensions de la percussion de deux masses. Pour cela, on introduit la notion de travail d'une force, d'énergie cinétique, et l'on décompose le problème en deux sous-problèmes : les chocs frontaux et désaxés.



- **<u>Mouvements de rotation</u>** : On traite ensuite la partie la plus délicate de notre projet : les chocs entre deux objets de formes quelconques. Une fois ces slides lus, le lecteur sera plus à même de comprendre la théorie physique intervenant dans notre projet, et saura (peut-être) décrire le mouvement de ces deux objets après leur collision :



- **Implémentation du projet** : Cette avant-dernière partie s'adresse avant tout aux lecteurs curieux, souhaitant savoir comment nous avons implémenté toute cette théorie en Scheme. Certaines fonctions capitales sont entièrement commentées, et les grands algorithmes que nous avons utilisés sont détaillés : la recherche de l'enveloppe convexe d'un polygone, la détection de collision entre deux convexes. Des près-requis en scheme sont indispensables pour comprendre les morceaux de codes présentés, mais la compréhension des algorithmes ne pré-suppose aucune connaissance dans ce language.



- **Conclusion** : Cette ultime partie du slideshow est l'occasion de faire un bilan du projet, de proposer quelques pistes pour lever ses limitations, et de remercier les enseignants nous ayant apporté leur aide au cours de ce semestre.

IV. Version pdf du slideshow

Dans le dossier principal se trouve l'exacte copie de la présentation slideshow au format pdf sous le nom de ***slideshow.pdf***

L'interactivité y est moins poussée (les liens scheme n'étant pas actifs), mais ce fichier, consultable sur Internet, présente l'intérêt d'être consultable sur n'importe quel poste relié au réseau, et de pouvoir être imprimé.

Sa numérotation tient compte de l'indexation des slides, et ce fichier est exporté au format **condense** : une page correspond au slide affiché après toute les transitions possibles depuis celui-ci (dans le cas d'affichages multiples, tous apparaissent sur des pages différentes).

Le fichier est téléchargeable depuis l'adresse :

http://projetphysics.teria.org/Site/Slideshow_files/slideshow.pdf

V. Configuration requise

Ce projet a été réalisé et testé :

- Sur la version **371** de **DrScheme**
- Le langage de développement est "*Assez gros scheme"* pour le logiciel (pour changer de langage, dans la barre d'outils DrScheme : Langages > Sélectionner le langage > PLT > Assez Gros Scheme), "*module…"* pour le slideshow
- Depuis un **mac**, version **10.5.2**, avec, pour le slideshow, une définition d'écran de **1920x1200 px**
- Depuis un **pc** tournant sur **windows XP**

VI. Développement

Physics a pris forme en parallèle des cours de l'option informatique du semestre 2 : notre façon de programmer a donc changé au fur et à mesure que nos connaissances se renforçaient, et nous avons connu deux changements d'orientation assez nets. Notre principale difficulté a donc été de rendre le plus cohérent possible ces différentes parties, écrites dans des logiques parfois complètement différentes :

- Dès la semaine 3, nous avons développé de petits modules graphiques (traitant des collisions) via la librairie world.ss
- Lorsque le cours sur la programmation orienté objet a été fait, et que la GUI a été introduite, nous nous sommes lancés dans le développement d'exemples plus complexes (comme les collisions multiples pour modéliser un gaz)
- Enfin, le traitement de la collision de deux convexes a été implémenté en utilisant seulement des structures

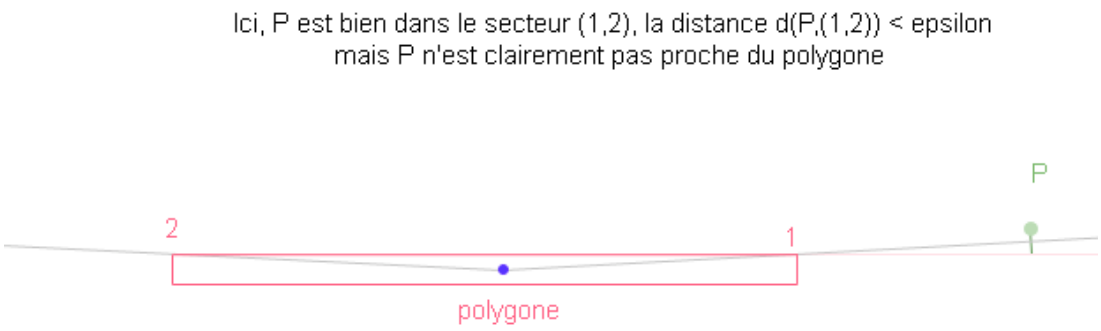
Plutôt que de réécrire l'ensemble des fichiers produits au cours des trois mois impartis, nous avons pris la décision de conserver chacune des trois parties, pour illustrer les différentes approches qu'offre scheme aux programmeurs.

VII. Limites du projet

Le projet, pour bien fonctionner, nécessite quelques concessions (abordées dans le slideshow) :

- Pour traiter la collision de deux convexes, il faut que leur vitesse soit faible, pour éviter d'avoir des cas d'interpénétrations non détectées : ceci est la conséquence directe de notre algorithme de détection (se reporter à la présentation).

- Pour que cet algorithme fonctionne bien, il faut que le convexe ne soit pas trop aplati, pour ne pas détecter des collisions là ou elles n'ont pas lieu, comme l'illustre le graphique suivant :



- On ne gère que la collision de convexes : pour traiter le cas des objets quelconques, il faudrait trianguliariser l'objet, et appliquer notre algorithme à toutes les oreilles périphériques (celles dont l'un des côtés coïncide avec une arrête du polygone).

- On accepte le fait qu'au contact des parois de la fenêtre d'affichage, un segment délimitant le polygone soit exactement à la coordonnée de la limite de la fenêtre : auquel cas il n'apparaît pas.

- On limite la zone de dessin à un cadre de dimensions inférieures à celle de la fenêtre pour éviter à l'utilisateur de créer des polygones dégénérés (entrant directement en collision avec la fenêtre).

VIII. Conclusion

Malgré ces restrictions, nous espérons que ce projet remplira ses principaux objectifs : rendre la mécanique un peu plus attractive, et vous faire passer de bons moments.

Pour nous aider dans la poursuite du développement de l'application, vous pouvez nous faire part de vos impressions, commentaires, ou rapports de bug à l'adresse :

projetphysics@teria.org

Nous vous invitons également à vous rendre régulièrement sur notre site pour les futures mises à jour du projet…

(the-end)