

# Projet TRIGOU

## Version 1.0 – révision 03112008-1105

Yann Régis-Gianas ([yrg@pps.jussieu.fr](mailto:yrg@pps.jussieu.fr))  
POCA - Master II Pro - Université Paris 7

3 novembre 2008

## 1 Principe

### Objectifs

1. Réfléchir à l'organisation d'un logiciel en vue de ses extensions futures.
2. Développer une (première?) application en C++.
3. Comparer deux langages de programmation objet.

**Description** Le principe du projet est de vous conduire à concevoir et à développer un programme [à l'aide d'objets](#). Nous avons vu que la modularisation permet de prévoir l'extensibilité d'un système. Pour mettre en avant cette caractéristique, le sujet de ce projet vous sera donné en [deux temps](#).

La première version du sujet vous conduira à développer un programme [à l'aide d'objets en C++](#). La seconde version du sujet sera constituée d'une série d'[extensions](#) du premier sujet. En fonction de la conception de la première version de votre programme, ces rajouts seront plus ou moins faciles à effectuer.

Enfin, vous re-programmerez entièrement votre projet dans [un autre langage](#) de votre choix.

## 2 Description du sujet

Ce projet consiste en la conception et l'implémentation d'un simulateur de casse-tête mécanique. Le sujet est sous-spécifié intentionnellement : vous êtes dans la situation d'un chef de projet devant extraire une spécification à partir d'une description informelle donnée par un client.

**Quelques dessins** Cette description informelle est donnée par la figure 1. La résolution du casse-tête se déroule comme suit :

- le joueur prend note des objectifs (ici, il n'y en a qu'un : conduire la boule numéro 1 dans le demi-cercle) ;
- le joueur place des **déclencheurs** (qu'il possède en nombre limité) ;
- la simulation physique est ensuite lancée :
  - si les objectifs sont remplis dans le temps imparti, le joueur a gagné ;
  - sinon, il a perdu.

Un déclencheur est formé de trois composantes :

- une zone de capteur Z ;
- un objet physique O ;
- une force F et son point d'application P.

Durant la simulation, la sémantique d'un déclencheur est :

« Tant que l'objet O est dans la zone Z, la force F est exercée sur P ».

On suppose que les objets formant le casse-tête ont une masse, un contour solide et qu'il existe une force de gravitation verticale dirigée vers le bas. Par simulation physique, on entend donc le calcul des trajectoires de ces objets soumis aux forces classiques de la mécanique des solides. Il va s'en dire qu'on ne demande pas au concepteur d'avoir une connaissance dans ce domaine mais d'utiliser un **moteur physique** existant !

**Interface graphique** La résolution d'un casse-tête doit être une activité ludique ! On fournira donc une interface graphique pour visualiser le casse-tête, définir ses déclencheurs et simuler sa résolution.

**Spécification des entrées/sorties textuelles** Pour stocker les casses-têtes et de façon à automatiser la validation fonctionnelle de votre simulateur, on se donnera la possibilité de spécifier textuellement :

- la description d'un casse-tête ;
- une proposition de résolution (la liste des déclencheurs) ;
- une trace de la simulation.

Ces fichiers seront stockés en XML suivant des DTDs qui seront fournis par le client dans le futur.

### 3 Instructions

**Calendrier** Les grandes étapes de votre projet sont synthétisées dans le tableau suivant :

Date	Description du délivrable
3 novembre	Fin de la définition des groupes
7 novembre – délivrable 1	Un premier document de modélisation
5 décembre – délivrable 2	Version 1.0
6 décembre	Réception de la version 2.0 du sujet
19 décembre – délivrable 3	Version 2.0 (en C++ et dans un autre langage)
15 janvier – délivrable 4	Rapport de modélisation et de retour d'expérience

**Délivrable 1 : un premier document de modélisation** Lorsque nous avons parlé de modélisation objet en cours, nous avons vu qu'une des premières phases est un raffinement des besoins exprimés. Pour mener à bien cette étape, vous devrez m'envoyer par email des diagrammes de cas d'utilisation à valider (utiliser le sujet [POCA-CLIENT] Cas d'utilisation). Ces diagrammes devront suivre la notation UML. Ils peuvent être manuscrits (scannés) ou produits à l'aide de logiciels prévus à cet effet.

Il va sans dire que ces diagrammes sont des documents de travail, sujets à d'inévitables évolutions. La question n'est donc pas de savoir si ces diagrammes correspondent à ce qui est attendu (il n'y a pas de réponse toute faite à un problème de conception). L'objectif de cette étape est de vous faire cerner les contours de votre projet et d'aboutir à une première architecture.

Le document à rendre sera formé (au minimum) de :

- Un paragraphe expliquant ce que vous avez compris des besoins du client.
- Un paragraphe décrivant les difficultés que vous pensez rencontrer.
- L'ensemble des cas d'utilisation que vous avez réussi à valider.
- L'ensemble des cas d'utilisation que vos n'avez pas réussi à valider.
- Une description de l'architecture préliminaire obtenue jusqu'à maintenant.

**Date limite** : 7 novembre – 23h59

**Contraintes techniques** Seuls les documents enregistrés dans des formats portables (PDF, PS, PNG, ...) seront acceptés.

### 4 Critères d'évaluation

- Gestion du projet
  - Conformité au sujet.
  - Respect des deadlines.
  - Effort de modélisation.
  - Organisation temporelle.
  - Répartition du travail.
  - ...
- Qualité du code
  - Code lisible, documenté.
  - Bon choix d'abstraction.
  - Utilisation des mécanismes du langage.
  - Pas d'horreur algorithmique.
  - ...
- Fonctionnalité
  - Passe une batterie de tests.

## 5 Coup de pouce

### Bibliothèques C++ potentiellement utiles

- Moteur physique :
  - BOX2D
  - CHIPMUNK
  - ...
- Visualisation graphique :
  - OPENGL
  - SDL
  - CAIRO
  - ...
- Manipulation de document XML :
  - XERCES
  - EXPAT
  - ...

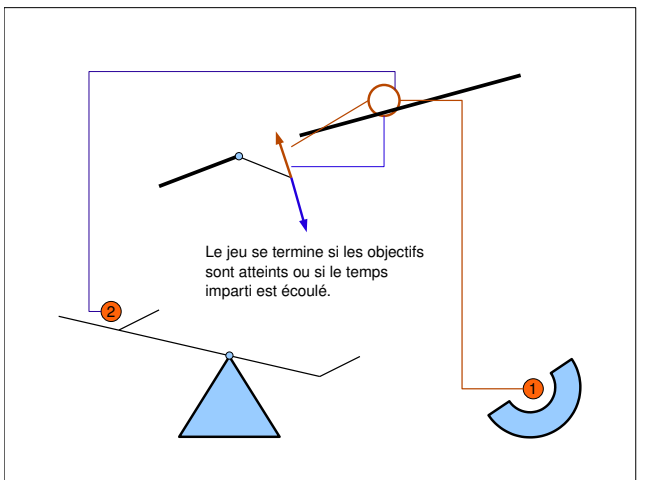
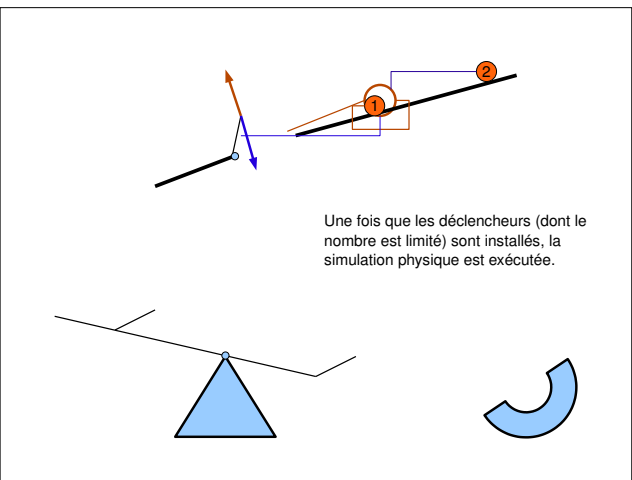
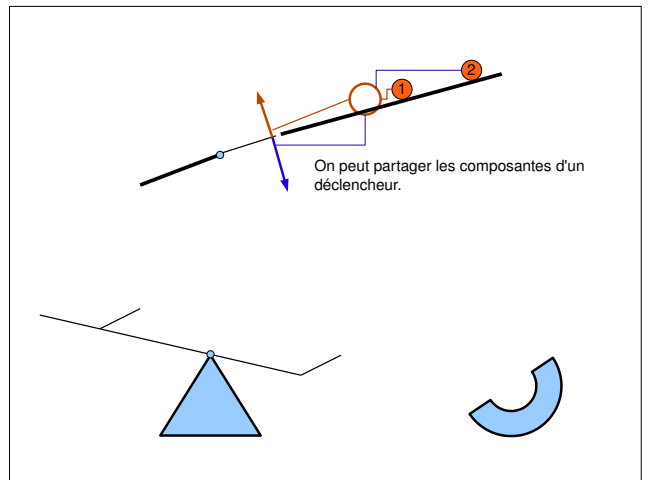
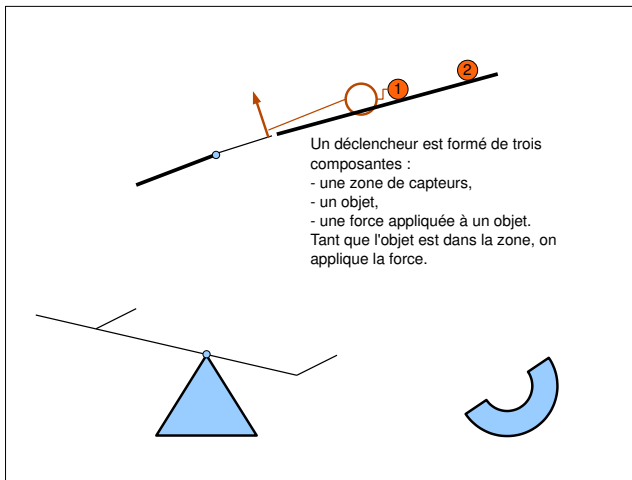
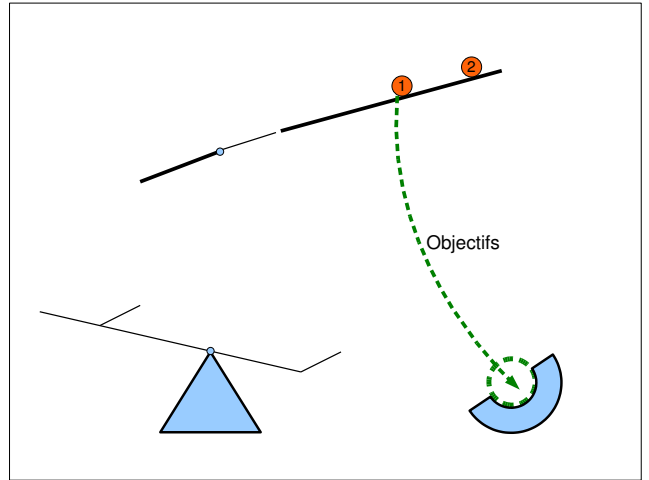
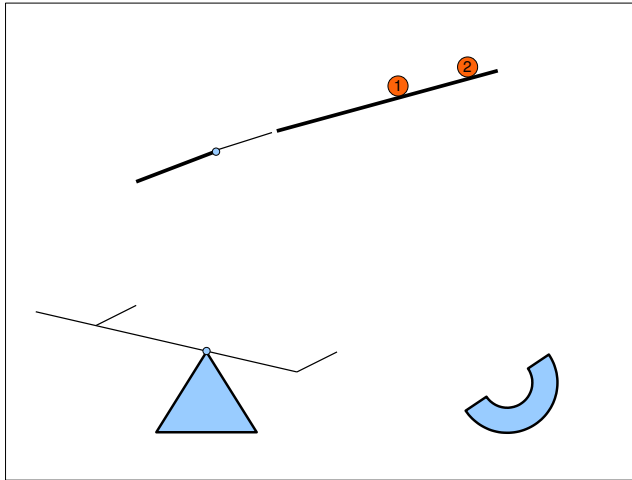


FIG. 1 – Une session de jeu.