



Département  
du Premier  
Cycle

---

Projet Informatique 2<sup>ème</sup> Année

SIMULATION MÉCANIQUE D'UN SOLIDE  
ET  
MOTEUR GRAPHIQUE 2D EN PASCAL

Tristan POURCELOT  
Victor CLEREN  
Rémi VERSCHELDE  
Théo GARCIA-GUITTON

Responsable : C. WOLF

Groupe 52

Année scolaire 2010-2011

Institut National des Sciences Appliquées de Lyon

# 1 Introduction

Nous sommes partis d'une idée simple, celle de modéliser la chute d'un objet sur un décor, sachant que l'utilisateur pouvait influencer à la fois sur le décor et sur l'objet.

## 2 Spécifications

L'objet principal de notre programme est de modéliser la chute d'un solide dans un fluide quelconque, en interagissant avec son environnement.

### 2.1 IHM - *Interface Homme - Machine*

Il est proposé à l'utilisateur de dessiner à la fois le décor et l'objet. Les deux dessins sont séparés en deux fenêtres, munie chacune de leur interface propre, puisque les outils nécessaires au dessin du décor ne sont pas les mêmes que pour le dessin de l'objet. L'utilisateur peut également ouvrir une fenêtre dédiée aux paramètres physiques, afin d'influer sur la valeur de la gravité notamment.

### 2.2 Problèmes physiques et méthodes de résolution

Le programme doit prendre en compte les différentes forces appliquées à un instant  $t$  sur le solide. Le mouvement de ce solide est donc déterminé par les équations de la mécanique newtonienne. On peut donc définir les différentes forces appliquées sur le solide :

- La gravité : .....  $\vec{P} = -m * g * \vec{y}_0$
- La poussée d'Archimède : .....  $\vec{F}_{Ar} = -\rho_{liquide} * V_{deplace} * \vec{y}_0$
- Les frottements lors du contact : .....  $\vec{F}_{fr} = -f * \vec{V}_{Objet/Décor}$
- La force lors du choc : .....  $\|\vec{V}_r\| = e * \|\vec{V}_i\|$  avec  $e \in [0, 1]$

La majorité des forces sont résolues à l'aide d'algorithmes basiques de calcul, prenant en compte des paramètres telles que le volume de l'objet, sa vitesse ou son poids. Les deux forces nécessitant de gros algorithmes de traitement sont les forces de frottement et le contact entre le solide et le décor. Pour la gestion du contact entre les deux solides, nous avons privilégiés une approche

## 3 Réalisation

### 3.1 Interface

### 3.2 Gestion des chocs

Calcul tangentes - Inversion force avec coeff

### 3.3 Remplissage de l'objet

FloodFill Scan Line

## 4 Bugs - Améliorations

Il subsiste de nombreux bugs et améliorations possibles de notre projet. En particulier, de nombreuses améliorations que nous pensions apporter à notre projet n'ont pas été implémentées par faute de temps, certains problèmes ayant été plus chronophages que prévu.

### 4.1 Améliorations

- Différents fluides (air, eau, vide ...)
- Variation aléatoire et incidence du vent
- Amélioration du modèle physique (élasticité ...)
- Modification du décor et de l'objet en temps réel par l'utilisateur
- Tracé de la trajectoire de l'objet

### 4.2 Bugs

- Lenteur de l'exécution sous Linux (pas de problèmes sous Windows)
- Le décor ne se remet pas à jour si on le redessine
- Calcul des tangentes peu précis (sinon il serait beaucoup trop lent)

## 5 Conclusion

Miam !

## 6 Annexes

### 6.1 Avis personnels et problèmes rencontrés

**Tristan :**

- Communication (qui fait quoi, quand et pourquoi...)
- Conflits SVN
- Implémentation pascal linux = caca

**Rémi :**

- Les autres ont rien foutus

**Victor :**

- Idem

**Théo :**

- niah

## 6.2 Diagrammes de classes

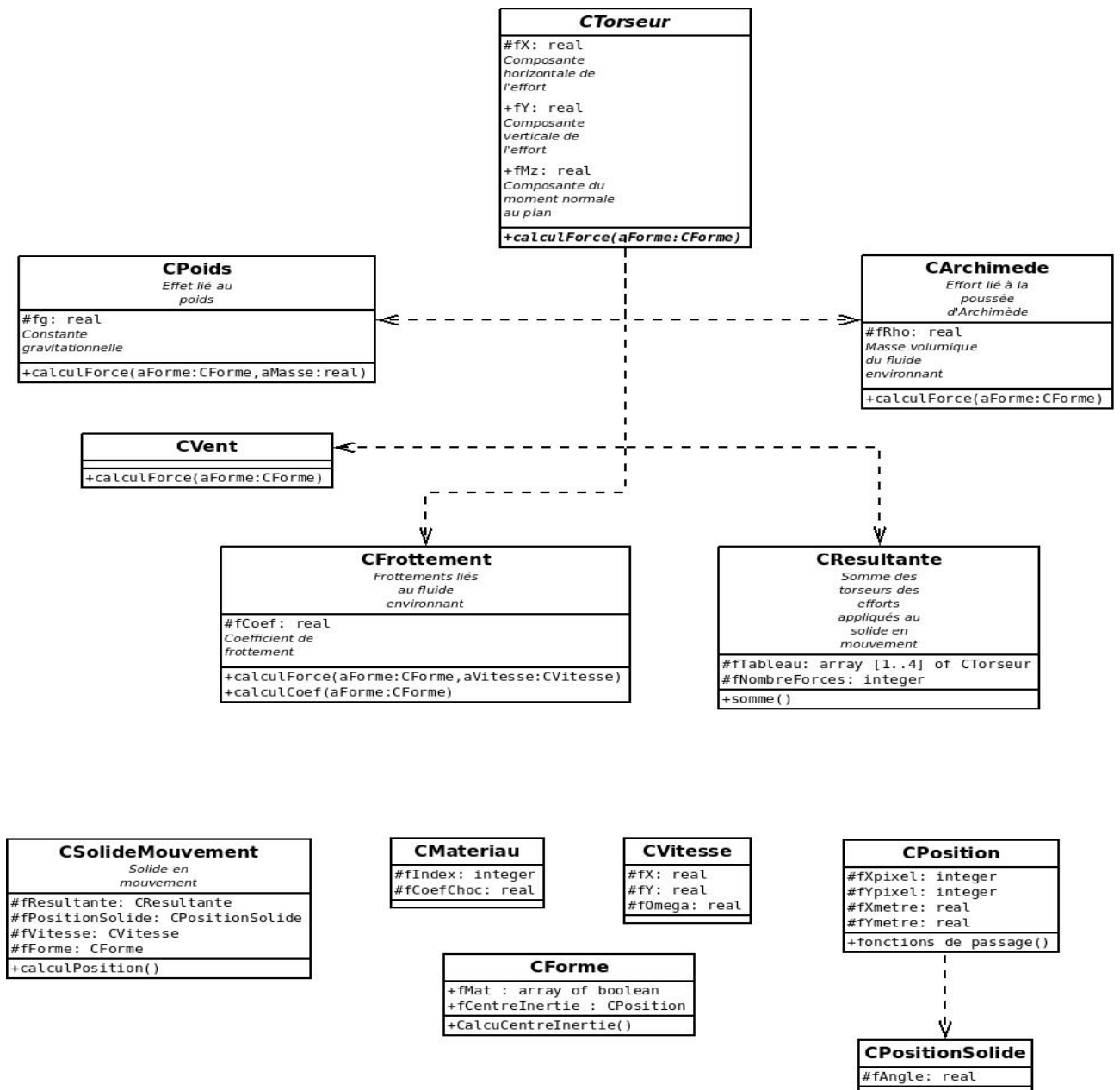


FIGURE 1 – Diagramme de classe utilisé pour notre projet

## 6.3 Impressions écrans

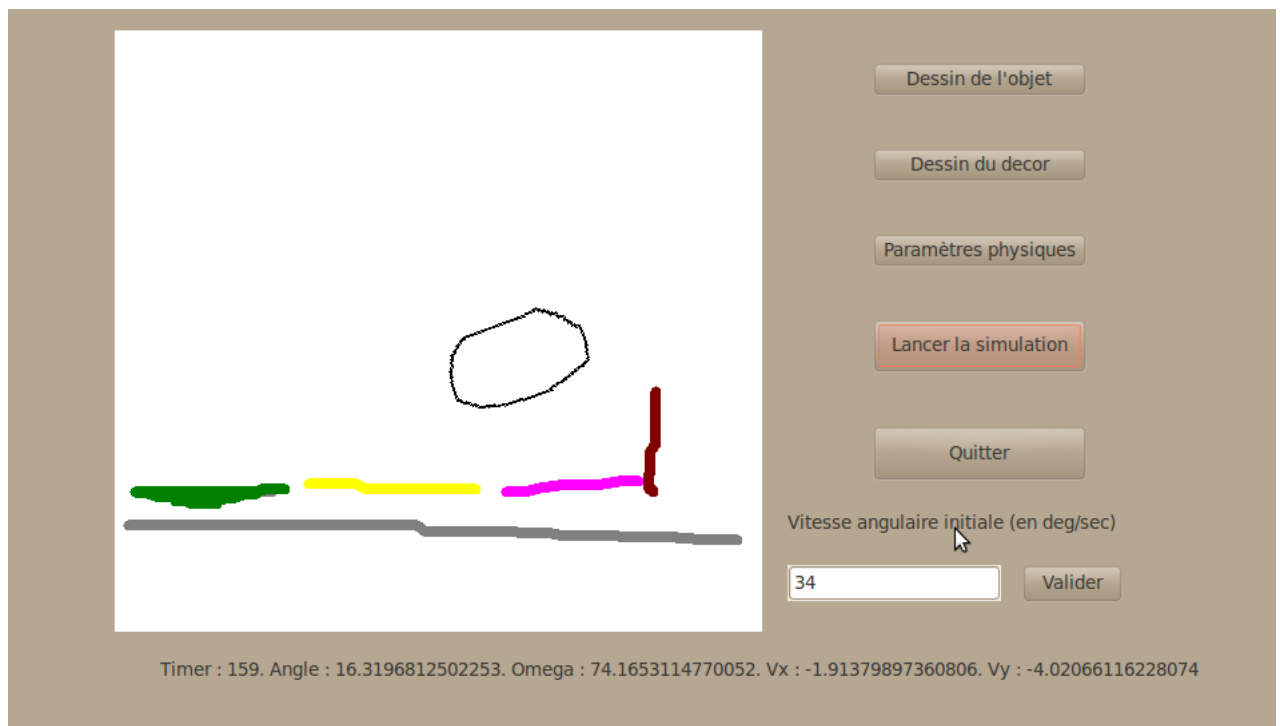


FIGURE 2 – Fenêtre principale

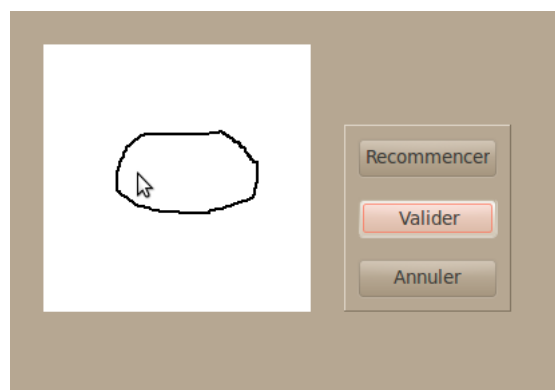


FIGURE 3 – Fenêtre gérant le dessin de l'objet

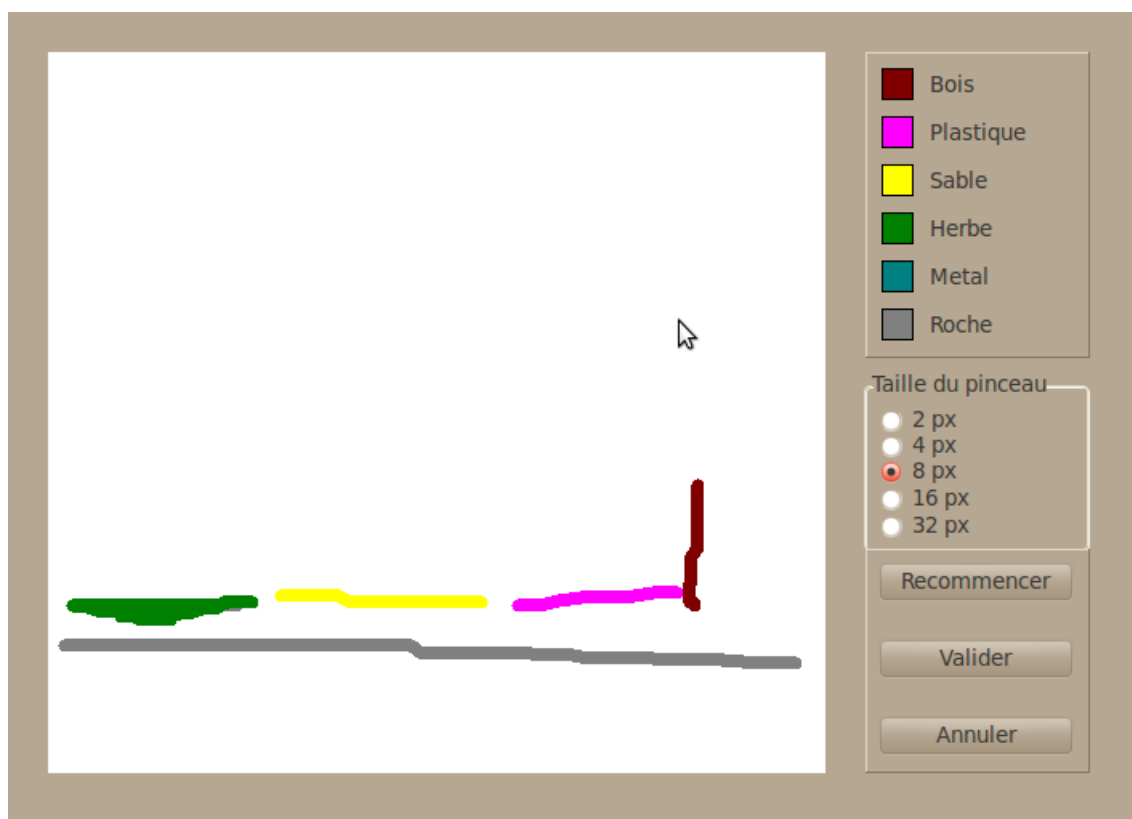


FIGURE 4 – Fenêtre gérant le dessin du décor