

RAPPORT : Projet POO 2

Simulateur de gravité en 2D

LUTTRINGER

Jean – Romain

MULLER

Cyrille

Présentation rapide :

Le programme a été codé en Lisaac et se lance, après compilation via la commande « **./main** ». Une fois lancé, il est possible de rajouter des cercles en faisant soit un clic gauche (pour un rayon de 15) soit un clic droit (pour un rayon de 35) à l'endroit voulu dans la fenêtre.

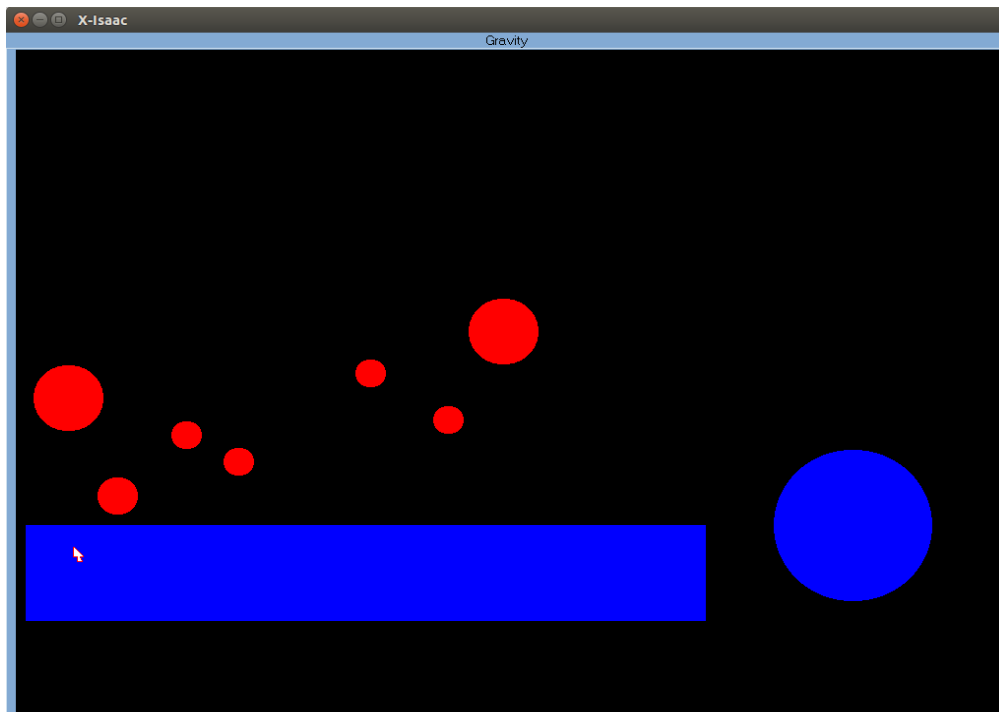
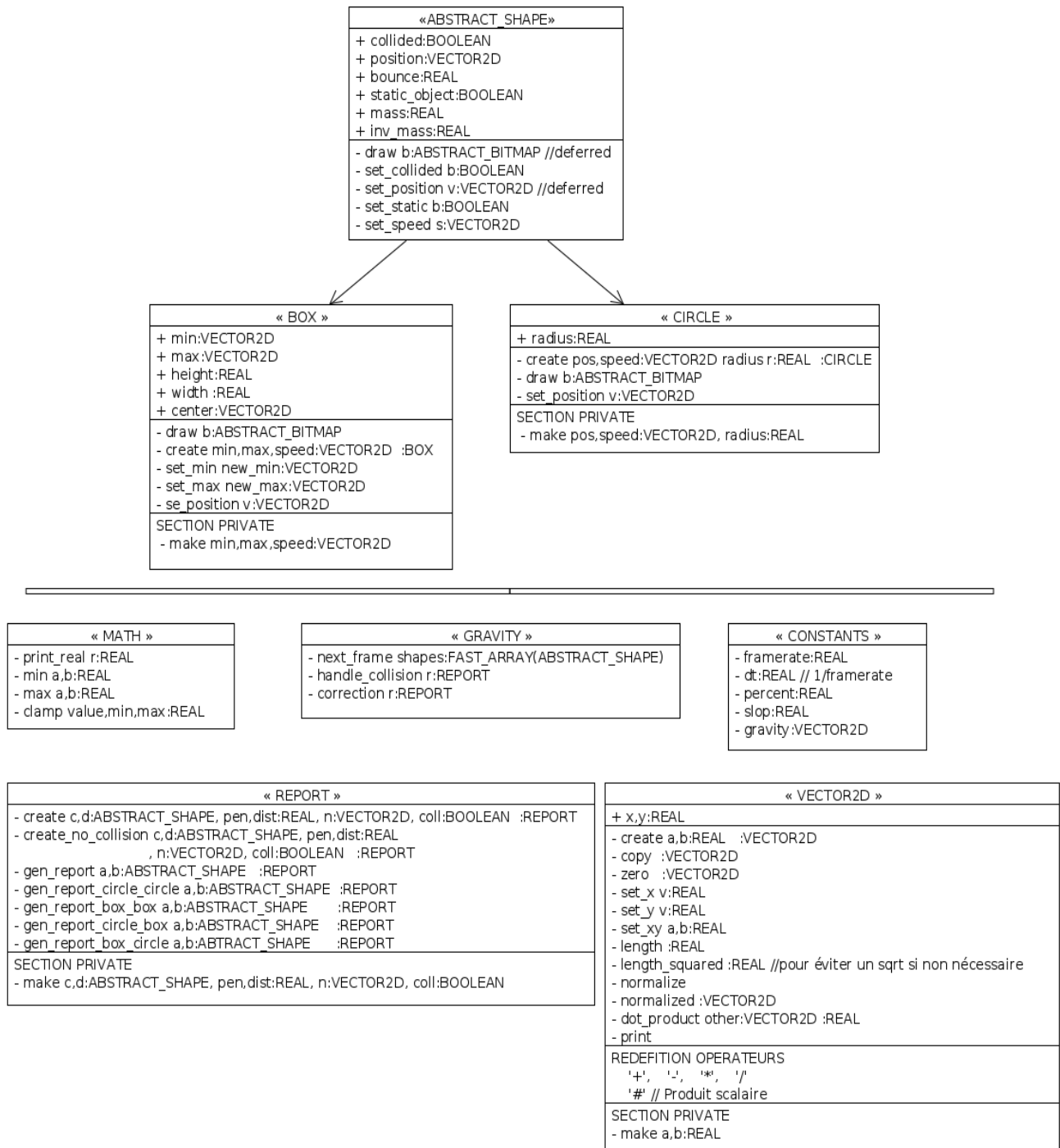


DIAGRAMME DES CLASSES

(en omettant l'héritage du prototype **OBJECT** présent dans tous les prototypes)

Le diagramme d'héritage est plutôt simple. Notre projet est constitué d'un prototype **ABSTRACT_SHAPE**, hérité en **Expanded** dans les prototypes **Box** et **Circle**. Les autres prototypes n'ont pas de lien d'héritage, et sont utilisés pour le calcul et la gestion des collision (i.e **Vecteur2D** , etc).



FONCTIONNEMENT GLOBAL

De manière périodique (selon la variable *framerate*) définie dans **CONSTANTS**, la fonction *next_frame* se lance avec comme argument l'ensemble des *ABSTRACT_SHAPE* présentes sur l'écran.

Cette fonction va calculer les nouvelles position des formes concernées en :

- Appliquant le vecteur *gravity* à leur vitesse
- Calculant les éventuelles collisions
- Appliquant le vecteur *speed* de chaque formes à leur vecteur *position*

Le calcul des collisions se fait comme suit :

Chaque couple de *ABSTRACT_SHAPE* est envoyé dans la fonction *gen_report*, qui va, en fonction du type dynamique des arguments reçus, appeler une des quatre fonctions *gen_report_X_X*.

La fonction *gen_report_X_X* appelée va calculer les informations relatives à la collision nécessaires pour traiter cette dernière (*distance*, *pénétration*, *vecteur*, etc). Ces informations sont ensuite mises dans un **REPORT** et renvoyées.

Si le couple de **ABSTRACT_SHAPE** n'entre pas en collision, le rapport renvoyé est généré via la fonction *create_no_collision*, mettant le booléen *collided* du rapport à false, et aucun calcul relatif à la collision n'est effectué.

Les informations obtenues sont alors utilisées par la fonction *handle_collision*, qui va, via l'équation de conservation de l'énergie de Newton, calculer les vecteurs d'impulsion devant être appliqués aux deux formes.

Dans le cadre de la collision entre deux **CIRCLE**, la génération du rapport est simple. Dans le cas d'une collision **CIRCLE vs BOX**, le problème est résolu comme suit :

- Un point X est calculé. Ce point est le point sur la **BOX** étant le plus proche du **CIRCLE**. Le vecteur de la collision est donc ramené à un vecteur entre X et le centre du cercle. La suite du traitement est à partir de ce point semblable au traitement d'une collision **CIRCLE vs CIRCLE**.

REPARTITION DU TRAVAIL

Le travail a été reparti de manière équitable. Le codage de chaque classe a été divisé entre nous deux, et les bugs / problèmes algorithmiques ont été abordés ensemble.