

# Rapport du projet Starlight

Florian Knop (39310) - Gatien Bovyn (39189)

31 mars 2015

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>Sections</b>	<b>3</b>
<b>Présentation des différentes classes</b>	<b>3</b>
Modèle . . . . .	3
Line . . . . .	3
Ellipse . . . . .	3
Vue . . . . .	4
DestinationView . . . . .	4
<b>Conclusion</b>	<b>4</b>
<b>Bibliographie</b>	<b>4</b>
<b>Annexes</b>	<b>4</b>
Démarches mathématique . . . . .	4
Trouver l'intersection entre deux droites . . . . .	4
Trouver l'intersection entre une droite et un segment . . . . .	5
Trouver le(s) intersections entre une ellipse et une droite . . . . .	5
Trouver le(s) intersections entre une ellipse et un segment . . . . .	6

## Introduction

Ce document vise à présenter le travail d'analyse et de programmation effectué lors de la réalisation du projet du laboratoire Langage C++ : Starlight.

Ce projet a été réalisé en binôme par Florian Knop, matricule 39310 groupe 2G13, et Gatien Bovyn, matricule 39189 groupe 2G11.

Le programme à concevoir consiste en une implémentation du modèle et d'une interface graphique du jeu baptisé Starlight, puzzle à 2 dimensions basé sur la lumière.

## Sections

### Présentation des différentes classes

L'implémentation du projet est divisée entre la partie modèle et la partie vue. Elle est également basée sur le design pattern Observateur / Observé comme demandé dans les consignes.

### Modèle

Un squelette de l'application nous a été fourni par Monsieur Absil. Ce squelette contient les fichiers suivants :

'point.h, source.h, dest.h, nuke.h, wall.h, crystal.h, lens.h, mirror.h, ray.h, level.h'.

### Line

Cette classe représente une droite, elle possède un point et un angle. Grâce à cela on peut trouver n'importe quel point de la droite grâce en ayant la distance entre le point d'origine et le point d'arrivée.

Dans les méthodes intersects(...) de la classe Line, un passage par pointeur de pointeur est fait car un pointeur est passé par valeur et lors de l'initialisation il ne pointera donc pas la même adresse mémoire que le pointeur d'origine. Si on passait donc par simple pointeur, notre pointeur copié aurait une bonne zone mémoire mais notre pointeur d'origine resterait à \*nullptr\*.

### Ellipse

Cette classe représente une conique de forme elliptique. C'est à dire une ellipse ou un cercle.

## Vue

L'interface graphique a été réalisée en 'Qt' à la main.

Les classes composant la partie vue de l'application sont :

### DestinationView

**Description** Classe modélisant la destination à atteindre par le rayon émis depuis la source pour gagner la partie.

### MapView

### MirrorView

### NukeView

### SourceView

### WallView

## Conclusion

## Bibliographie

## Annexes

### Démarches mathématique

#### Trouver l'intersection entre deux droites

$$D \equiv y = ax + b \tag{1}$$

$$D \equiv x = k \tag{2}$$

(1) représente une droite non verticale. (2) représente une droite verticale où k est une valeur quelconque sur l'axe des x.

Dans un premier temps, il faut tester que la pente  $a$  est différente. Si cette dernière est égale, il n'y a pas d'intersections. En effet, si les pentes sont les mêmes, cela signifie que les droites sont soit parallèles soit égales. Dans le cas de droites égales, on considère qu'il n'y a pas d'intersections, bien qu'en réalité il y en a une infinité.

Si la pente est différente, l'intersection entre deux droites est assez simple, il suffit d'injecter une variable d'une des deux équations dans l'autre. Dans le cadre du projet, il faut évidemment faire attention aux droites verticales.

Le cas d'une droite verticale :

On injecte (2) dans (1).

$$D \equiv y = ak + b \quad (3)$$

Avec (3) on obtient  $y$  du point dont le  $x = k$  pour une droite verticale (2).

Pour deux droites non verticales on injecte le  $y$  d'une équation de type (1) dans le  $y$  d'une autre équation de typ (1).

$$aremplir \quad (4)$$

$x = \frac{(y-b)}{a}$  pour une droite non verticale dont le  $y$  est celui obtenu plus tôt (1).

### Trouver l'intersection entre une droite et un segment

### Trouver le(s) intersections entre une ellipse et une droite

La formule d'une ellipse est :

$$E \equiv \frac{(x - x1)^2}{a^2} + \frac{(y - y1)^2}{b^2} = 1$$

où  $x1$  et  $y1$  sont respectivement les coordonnées  $x$  et  $y$  du centre de l'ellipse. où  $a$  et  $b$  sont respectement les rayons de l'axe  $x$  et  $y$ .

Pour trouver une intersection entre une ellipse et une droite, il faut éгалer deux variables identiques :

On doit donc remplacer la variable  $x$  ou  $y$  de la droite dans l'équation de l'ellipse.

Le cas de la droite verticale :

Dans ce cas la, il n'y a pas de choix, il faut remplacer  $x$  dans l'équation de l'ellipse. Nous avons également décidé de refactoriser l'équation en prenant le PPCM (Plus petit commun multiple) de  $a^2 * b^2$  que nous apellerons ici  $lcm$  pour Least Commun Multiple. Ce choix a été fait pour éviter les overflows lorsque de

nombres trop grands sont mis au carré et multipliés. Bien que dans notre cas, on nous avons rarement des nombre pouvant obtenir un tel résultat.

$$E \equiv (lcm y \cdot (k - x1)^2) + ((y - y1)^2 \cdot lcm x) = lcm$$

où  $lcm y$  est le facteur par lequel il faut multiplier  $b^2$  (rayon  $y$  au carré) pour obtenir  $lcm$ ,

où  $lcm x$  est le facteur par lequel il faut multiplier  $a^2$  (rayon  $x$  au carré) pour obtenir  $lcm$ .

$$E \equiv lcm y \cdot (k - x1)^2 + (y^2 + y1^2 - 2 \cdot y1 \cdot y) \cdot lcm x = lcm$$

$$E \equiv lcm y \cdot (k - x1)^2 + lcm x \cdot y^2 + lcm x \cdot y1^2 - 2 \cdot lcm x \cdot y1 \cdot y - lcm = 0$$

Avec ceci, il reste plus qu'à résoudre l'équation du second degré avec

$$\rho = b^2 - 4ac$$

où

$$a = lcm x$$

$$b = 2 \cdot lcm x \cdot y1 \cdot y$$

$$c = (lcm y \cdot (k - x1)^2) + (lcm x \cdot y1^2) - lcm$$

Le nombre d'intersections est différent selon la valeur de  $\rho$ .

$$n = \begin{cases} 0 & \text{si } \rho < 0 \\ 1 & \text{si } \rho = 0 \\ 2 & \text{si } \rho > 0 \end{cases}$$

$$y = \frac{-b}{2a}$$

$$y1 = \frac{(-b + \sqrt{\rho})}{2a}$$

$$y2 = \frac{(-b - \sqrt{\rho})}{2a}$$

On a donc le(s)  $y$  du/des points d'intersections, et le  $x$  vaut  $k$  (de l'équation de départ).

Le cas de la droite non verticale :

C'est le même principe que le cas de la droite verticale sauf que pour trouver la deuxième variable finale il faudra remplacer la variable trouvée dans l'équation de la droite.

### Trouver le(s) intersections entre une ellipse et un segment