

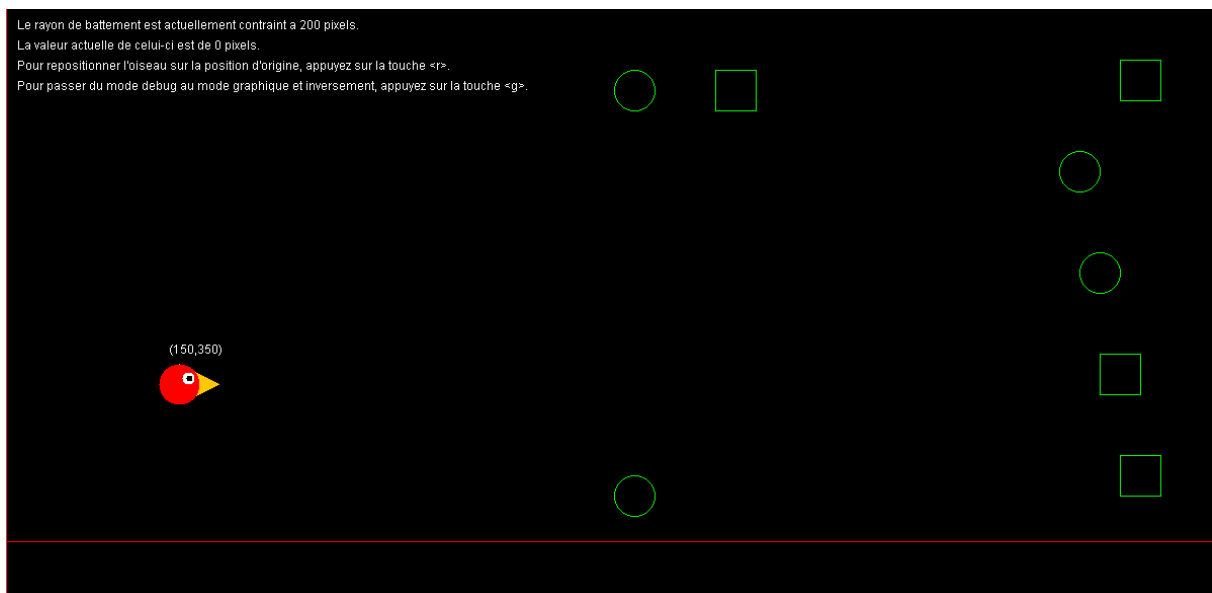
## Projet AngryNerds, N3P1, groupe L2

L'équipe de ce projet est composée de AL YASINI Omar, BETTE Guillaume, DEJONGHE Rémi, LEFEBVRE Robin et RAES Rémy.

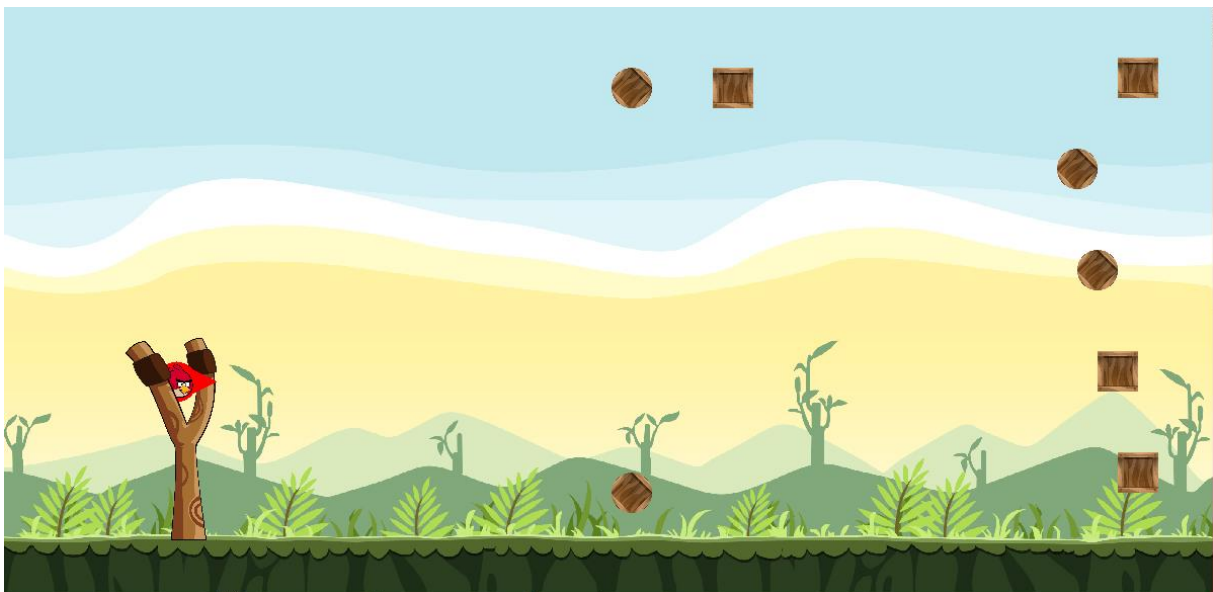
Organisation du travail : Nous avons séparé le projet en grandes tâches à accomplir (fonction qui gère les courbes, obstacles...) que nous avons réalisées en équipe, sans avoir vraiment assigné de tâche particulière aux membres de l'équipe.

Vous pourrez trouver les schémas UML des différents jalons du projet dans le dossier 'ProjetMode/docs/uml'.

Malheureusement, la javadoc n'a pas été entièrement écrite car certaines personnes ne s'impliquent pas dans le projet.



Il est possible de passer du mode 'debug' au mode graphique en appuyant sur la touche 'g', à n'importe quel moment lors de l'exécution du programme.



## Liste des tests

La classe OiseauTest teste la classe Oiseau . Elle est composée de plusieurs tests sur les méthodes de cette classe.

testOiseauCourant sert à tester le cas où l'oiseau est à sa position initial et tester si les valeur trouvées dans les constantes correspondent grâce à la méthode getC.

testSetOiseau sert à tester la méthode setTaille et la méthode setC.

La classe Obstacle est testée par la classe ObstacleTest. Avant la modification de cette dernière classe, elle servait à tester deux types d'obstacles représentés par les classes Obstacle et ObstacleMouvant.

testObstacleNonMouvant sert à tester la classe Obstacle en testant les méthodes getTaille, setActif, isActif, setTaille, setCarre et isCarre.

testObstacleMouvant a la meme utilité que la classe précédente pour maintenant la classe ObstacleMouvant avec la test des méthodes moveX, moveY, setLimites\_x et setLimites\_y.

La classe OutilsPoints, quand à elle, est testée par la classe OutilsPointsTest. Elle teste le calcul de la distance entre deux points et l'angle que crée ces deux points.

testOutilsPoints sert à tester les méthodes getDistance et getAngle.

## Design Patterns

Le modèle MVC a été intégré à notre projet.

De plus, nous avons intégré deux design patterns :

- ➔ La classe Constantes a été conçue suivant le pattern Singleton.
- ➔ Les obstacles ont été gérés avec le pattern Factory.