TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

BÁO CÁO

**NHẬP MÔN JAVA**

NGÀNH CÔNG NGHỆ THÔNG TIN

**GAME GIỮA KỲ**

Sinh viên thực hiện: **Nguyễn Việt Hoàng (20176020)**

**Nguyễn Nam Anh (20176002)**

**Vũ Đức Duy (20176013)**

Lớp LTU - K16

Giáo viên hướng dẫn: Trần Vĩnh Đức

HÀ NỘI 12-2018

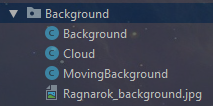
1. Ý tưởng

* Game mang tên “Rise of the birds” được đặt theo tên bộ phim “Rise of the planet of the apes”.
* Dự án game này được lấy ý tưởng từ 2 tựa game nổi tiếng là Gunny và Angry Birds.

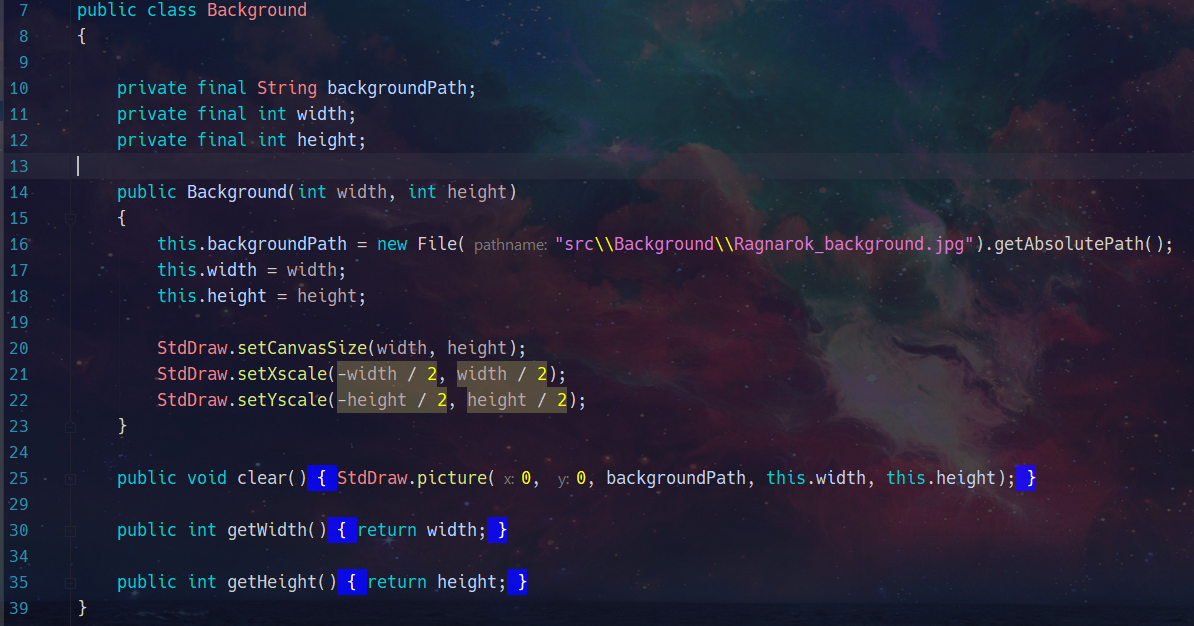
1. Lối chơi

* Người chơi cần căn chỉnh lực và góc sao cho chú chim bắn trúng vào đổi thủ (lấy hình mẫu là loài chuột) ít nhất 3 lần.
* Điều chỉnh thanh lực và chỉnh góc bằng cách nhấn giữ phím SPACE, thả phim để chọn mức lực và góc thích hợp.

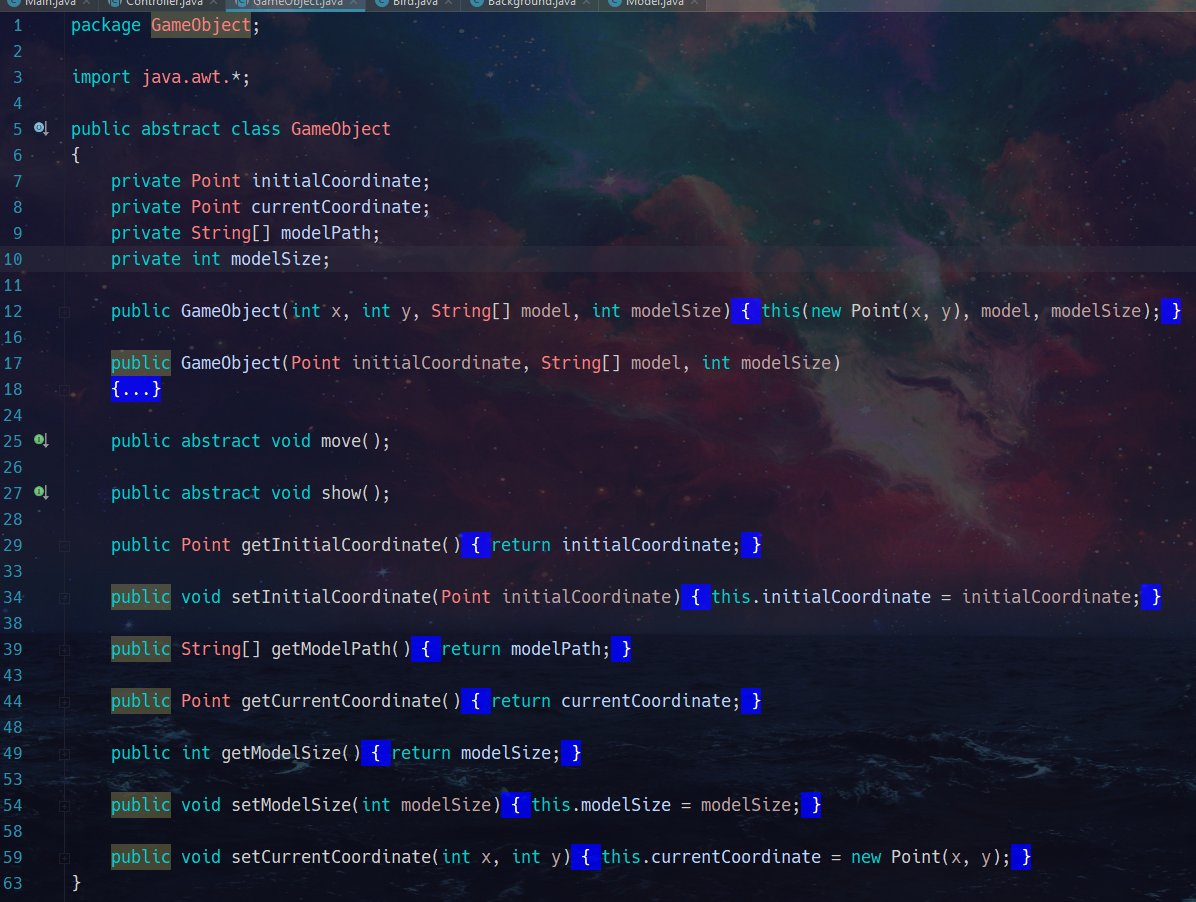
1. Chi tiết quá trình thực hiện
   1. Bố cục chính của project:
      1. Background: phần hình nền.
      2. Game object: các vật thể trong game (VD: Bird, Mouse,…).
      3. Model: các hình mẫu cho các vật thể trong game.
      4. Gameplay: tập trung các đoạn code vận hành, xử lý game.
   2. Thư viện được sử dụng
   * Các thư viện của Java
   * Standard Libraries (package version) - <https://introcs.cs.princeton.edu/java/stdlib/>
2. Giải thích code
   1. Background



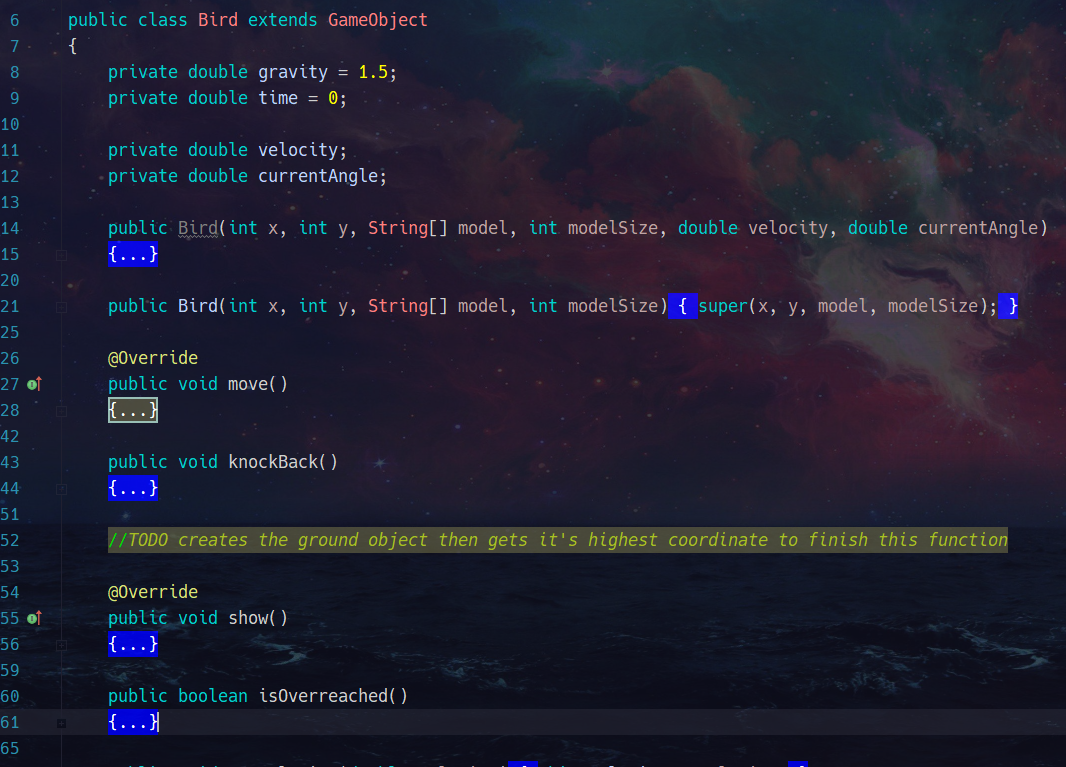
* Class Background:
  + Khởi tạo một instance Background sẽ đồng thời căn chỉnh kích thước của Canvas sao cho phù hợp với chiều dài và chiều rộng của Background.
  + Gọi hàm clear() sẽ làm mới lại Background (xóa toàn bộ những gì đang được “vẽ” trên Background).



* Class Cloud: chưa tìm ra giải pháp sử dụng
* Class MovingBackgound: chưa tìm ra giải pháp sử dụng
  1. GameObject
* Class GameObject:
  + Được khởi tạo dưới dạng abstract class với mục đích khai báo các đặc tính cơ bản cho các vật thể chính trong game: Bird và Mouse kế thừa.
  + Gồm các fields cơ bản của một vật thể trong game: tọa độ ban đầu, tọa độ hiện tại, path của hình mẫu vật thể trong game, kích thước vật thể.
  + Có 2 hàm abstract là move() và show() với chức năng thay đổi phụ thuộc vào subclass được kế thừa



* Class Bird:
  + Kế thừa từ class GameObject.
  + Có thêm các fields liên quan đến công thức ném vật theo quỹ đạo parabol: gravity, time, velocity, currentAngle.
  + Các hàm đặc trưng:
    - move(): override từ class GameObject nhằm xử lý quỹ đạo bay của Bird theo quỹ đạo Parabol.
    - knockBack(): đổi hướng bay của Bird theo chiều ngược lại.
    - show(): hiển thị Bird lên Background với tọa độ ban đầu cho trước và góc bắn ban đầu được căn chỉnh bởi người chơi.
    - isOverreached(): nhận biết khi Bird bay hết màn hình.

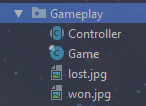


* Class Mouse:
  + Kế thừa từ GameObject
  + Các fields đặc trưng:
    - MAX\_RANGE: khoảng di chuyển tối đa của Mouse
    - step: khoảng cách mỗi lần di chuyển
    - movingDistance: quãng di chuyển của Mouse từ lúc bắt đầu
    - HP: số lần bị bắn trúng để tiêu diệt
    - justGotHit: nhận biết khi Mouse vừa bị bắn trúng bởi Bird (nhằm tránh lỗi Bird gọi hàm knockBack() liên tục khi chưa di chuyển một khoảng đủ xa khỏi Mouse)
  + Các hàm đặt trưng:
    - getDamaged(): giảm HP, giảm kích thước và tăng step của Mouse khi bị bắn trúng
    - move(): goị hàm moveBackAndForth() khi HP > 0
    - moveBackAndForth(): di chuyển Mouse qua trái phải liên tục

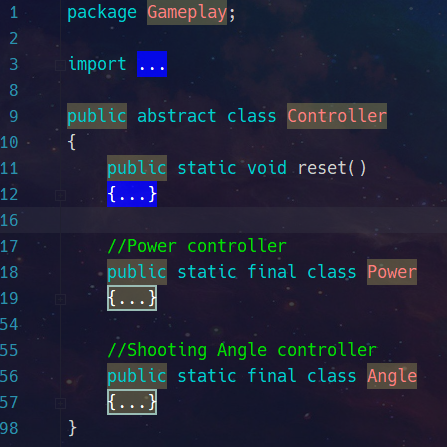


* 1. Gameplay

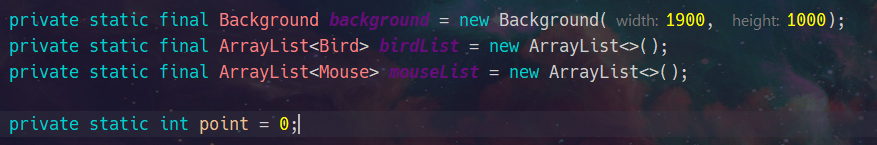
Tập trung các hàm xử lý và vận hành game



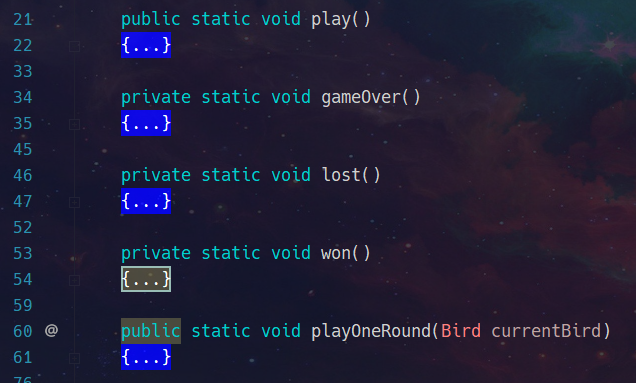
* Class Controller:
  + Inner class Power: điều chỉnh thanh lực để thiết lập lực bắn cho Bird
  + Inner class Angle: điều chỉnh góc bắn cho Bird



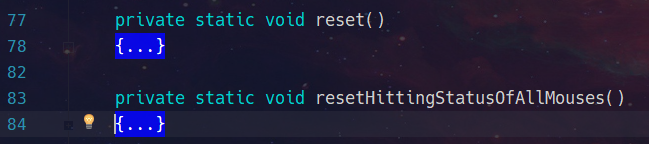
* Class Game:
  + Tập trung chủ yếu các đoạn code vận hành game
  + Chi tiết:
    - Khởi tạo Background, list lưu trữ Bird và Mouse, field tính điểm



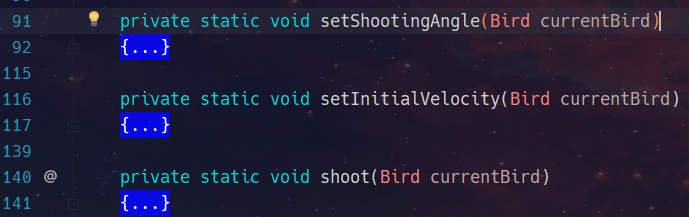
* + - Các hàm khởi chạy game, kết thúc



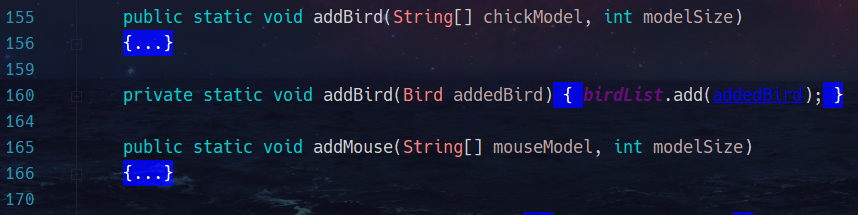
* + - Reset các thông số về ban đầu để bắt đầu lượt bắn mới



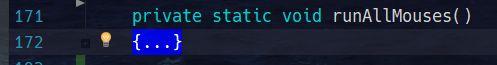
* + - Lần lượt: chỉnh góc bắn, chỉnh lực bắn, bắn (bắt đầu quá trinh bay)



* + - Thêm Bird, Mouse vào danh sách (số Bird = số lượt chơi)

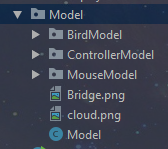


* + - Di chuyển các Mouse liên tục và theo dõi lượng HP từng Mouse trong danh sách, nhận diện khi bị bắn trúng

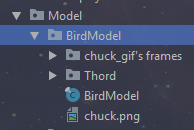


* 1. Model

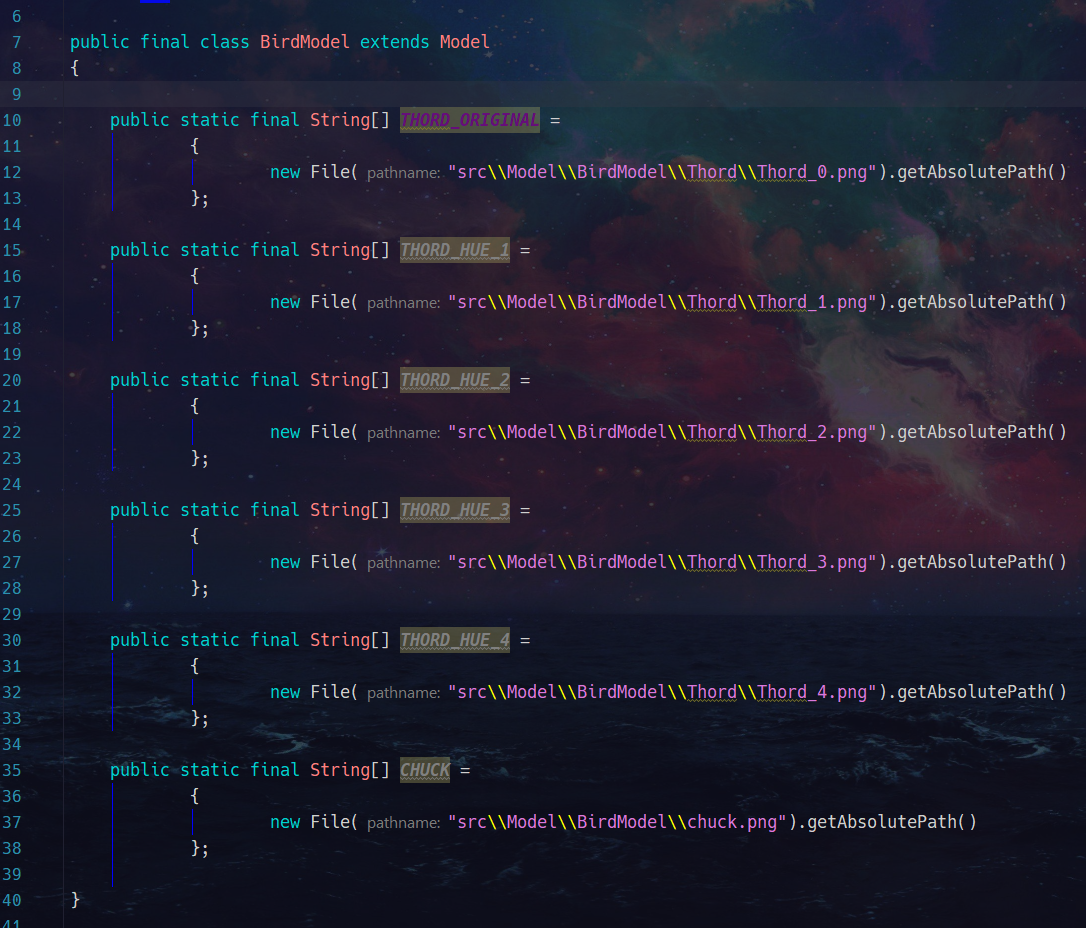
Tập trung các file ảnh model của các vật thể trong game



* Package BirdModel: lưu trữ hình mẫu cho Bird



* + Class BirdModel: các field được khai báo dưới dạng static và có access modified là public để có thể gọi trực tiếp các model 1 cách dễ dàng từ class khác mà không cần thông qua khởi tạo 1 instance cục bộ



* Package MouseModel: tương tự như package BirdModel
* Package ControllerModel: lưu trữ hình mẫu cho thanh lực và mũi tên định hướng bắn cho class Controller trong package Gameplay