**BÁO CÁO TÓM TẮT GAME KHỦNG LONG (DINO RUN)**

**Tóm tắt**

Dự án trò chơi Khủng Long được phát triển bằng ngôn ngữ lập trình Java. Trò chơi này bao gồm nhiều đối tượng trò chơi như Khủng Long, Chim, Xương Rồng, Mây, và Đất. Mỗi đối tượng có các thuộc tính và phương thức riêng để quản lý trạng thái và hành vi của nó. Ngoài ra, trò chơi còn có các lớp quản lý để xử lý điều khiển, âm thanh, và kẻ thù, cũng như các lớp hỗ trợ và giao diện người dùng.

## Thiết kế MVC

### Model

#### Đối tượng Trò chơi

* **Dino.java**: Quản lý nhân vật khủng long, chuyển động và trạng thái của nó.
* **Birds.java**, **Cactuses.java**, **Clouds.java**, **Land.java**: Quản lý các thực thể khác trong trò chơi và hành vi của chúng.
* **Score.java**: Quản lý điểm số của trò chơi.

#### Trạng thái Trò chơi

* **DinoState.java**: Đại diện cho các trạng thái khác nhau của khủng long (chạy, nhảy, v.v.).
* **EnemyType.java**: Liệt kê các loại kẻ thù khác nhau.
* **GameState.java**: Quản lý trạng thái tổng thể của trò chơi (đang chơi, kết thúc trò chơi, v.v.).

### View

#### Giao diện Người dùng

* **GameScreen.java**: Xử lý việc hiển thị các đối tượng trò chơi trên màn hình.
* **GameWindow.java**: Quản lý cửa sổ trò chơi và vòng lặp chính của trò chơi.

### Controller

#### Quản lý

* **ControlsManager.java**: Xử lý đầu vào từ người dùng và điều khiển.
* **EnemyManager.java**: Quản lý việc tạo kẻ thù và tương tác.
* **SoundManager.java**: Xử lý âm thanh và nhạc nền của trò chơi.

### Tiện ích

* **Resource.java**: Tải và cung cấp truy cập đến các tài nguyên trò chơi.

Bằng cách tổ chức trò chơi theo mô hình MVC này, chúng ta có thể tách biệt các mối quan tâm về quản lý dữ liệu, giao diện người dùng và đầu vào từ người dùng, giúp mã nguồn trở nên dễ bảo trì hơn.

### **Thiết kế khác**

Ngoài sơ đồ lớp, dự án trò chơi Khủng Long có thể cần thêm các thiết kế sau để hoàn thiện:

1. **Sơ đồ trình tự (Sequence Diagram):** Mô tả các tương tác giữa các đối tượng trong các kịch bản khác nhau, ví dụ như khi khủng long nhảy, khi va chạm với kẻ thù, hoặc khi trò chơi kết thúc.
2. **Sơ đồ hoạt động (Activity Diagram):** Minh họa luồng hoạt động trong trò chơi, từ khi bắt đầu đến khi kết thúc, bao gồm các hành động như cập nhật vị trí, kiểm tra va chạm, và vẽ đồ họa.
3. **Sơ đồ trạng thái (State Diagram):** Diễn tả các trạng thái khác nhau của khủng long (chạy, nhảy, cúi xuống, chết) và các chuyển đổi giữa các trạng thái này.
4. **Sơ đồ thành phần (Component Diagram):** Hiển thị các thành phần phần mềm của trò chơi và mối quan hệ giữa chúng, bao gồm các lớp, gói, và các thư viện bên ngoài.
5. **Sơ đồ triển khai (Deployment Diagram):** Mô tả cách triển khai phần mềm trên phần cứng, bao gồm các thiết bị và hệ điều hành.

Việc sử dụng các sơ đồ này sẽ giúp nhóm phát triển hiểu rõ hơn về cấu trúc và hành vi của hệ thống, từ đó dễ dàng duy trì và mở rộng trò chơi trong tương lai.

### Các Mẫu Thiết Kế Đã Sử Dụng

1. **Singleton**: GameScreen
2. **Strategy**: Dino
3. **Observer**: Score
4. **Composite**: CactusGroup, Clouds
5. **Decorator**: Animation
6. **Command**: ControlsManager
7. **Thread Pool (hoặc Proactor Pattern)**: SoundManager
8. **State**: DinoState, GameState
9. **Factory Method**: EnemyType

**Thiết kế lớp (Class Diagram)**

