

KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



ĐỒ ÁN MÔN HỌC
THIẾT KẾ KIẾN TRÚC PHẦN MỀM

Đề tài

Game Match - 3

Nhóm sinh viên thực hiện: **Nhóm 45**

Danh sách sinh viên:

1. Lưu Quốc Gia	58265	65KSPM
-----------------	-------	--------

Giảng viên: **ThS. Phạm Hữu Tùng**

Hà Nội, tháng 3/2025

Mục lục

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	20
1.1 Lý do chọn đề tài	20
1.2 Mục tiêu đồ án	20
1.3 Phạm vi và giới hạn	20
1.4 Phương pháp thực hiện	20
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG	21
2.1 Mô tả bài toán	21
2.2 Yêu cầu chức năng	21
2.3 Yêu cầu phi chức năng	21
2.4 Use Case Diagram tổng quát	21
2.5 Mô tả các Use Case chính	22
2.6 Biểu đồ hoạt động	23
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	23
3.1 Lựa chọn mô hình kiến trúc	23
3.2 Sơ đồ kiến trúc tổng thể	24
3.3 Thiết kế biểu đồ trình tự	25
3.4 Thiết kế biểu đồ lớp	30
3.5 Thiết kế giao diện	31
3.6.1 Giao diện người dùng	31
CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG	32
4.1 Môi trường triển khai	32
4.2 Cài đặt hệ thống	32
4.2.1 Yêu cầu hệ thống	32
4.2.2 Các bước cài đặt chương trình	32
CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ	33
5.1 Kết quả thử nghiệm hệ thống	33
5.2 Đánh giá hiệu quả hệ thống	33

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Lý do chọn đề tài

Trò chơi Match-3 là một trong những thể loại game phổ biến, có lối chơi đơn giản nhưng thu hút người chơi. Việc xây dựng một trò chơi như vậy giúp sinh viên áp dụng kiến thức về lập trình hướng đối tượng, kiến trúc phần mềm, và phát triển phần mềm thực tế.

1.2 Mục tiêu đề án

- Xây dựng trò chơi Match-3 hoàn chỉnh bằng Unity
- Hiểu và áp dụng kỹ thuật phát hiện match, swap, fill, tạo BonusItem
- Hiện thị giao diện người dùng đơn giản, thân thiện

1.3 Phạm vi và giới hạn

- Phạm vi: Game Match-3 cơ bản có hiệu ứng, âm thanh, điểm số
- Giới hạn: Không tích hợp đa nền tảng, không có lưu game/online play

1.4 Phương pháp thực hiện

- Nghiên cứu tài liệu về game Match-3
- Thiết kế hệ thống
- Cài đặt bằng Unity C#
- Kiểm thử và đánh giá

CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

2.1 Mô tả bài toán

Tạo một game Match-3: người chơi hoán đổi các ô để tạo ra các chuỗi 3 ô cùng loại trở lên và được tính điểm.

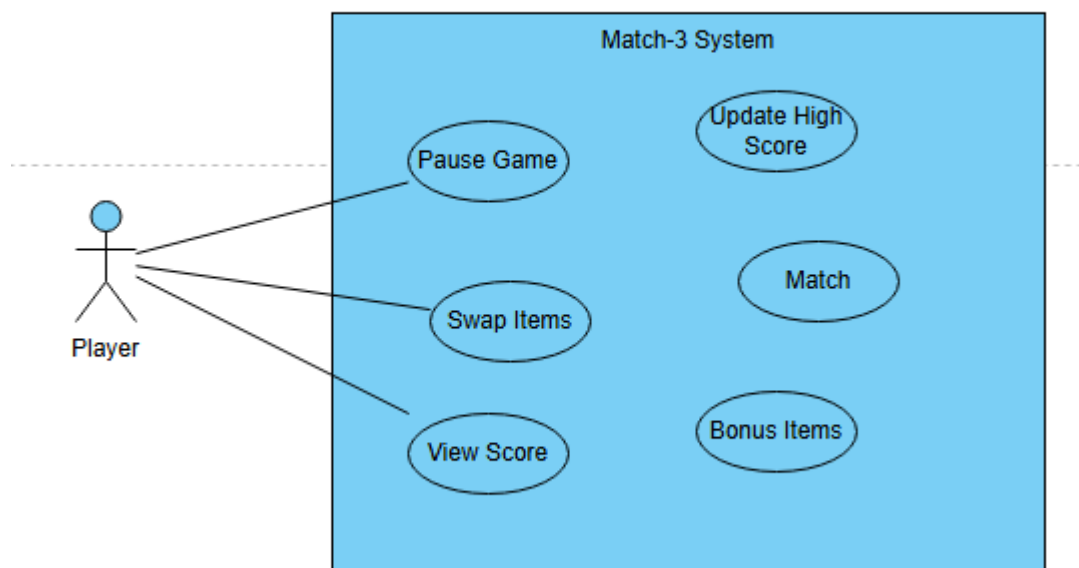
2.2 Yêu cầu chức năng

- Swap các ô liền kề
- Phát hiện và xử lý match
- Tạo bonus khi match đặc biệt
- Hiện thị điểm số và hiệu ứng

2.3 Yêu cầu phi chức năng

- Giao diện đơn giản, dễ chơi
- Hiệu suất ổn định
- Dễ mở rộng tính năng

2.4 Use Case Diagram tổng quát



Hình 1: Use case tổng quát hệ thống

2.5 Mô tả các Use Case chính

Use Case: Swap Items

Thành phần	Mô tả
Ac tor	Người chơi
Mục đích	Thay đổi vị trí 2 ô liền kề để tạo thành chuỗi match
Tiền điều kiện	Game đang ở trạng thái Playing, 2 ô được chọn cùng hàng/cột và cách nhau 1 ô
Luồng chính	<ol style="list-style-type: none">1. Player chọn ô thứ nhất (giữ chuột/touch)2. Kéo sang ô thứ hai (liền kề)3. Hệ thống kiểm tra tính hợp lệ:<ul style="list-style-type: none">- Nếu hợp lệ: hoán đổi vị trí → Kích hoạt Detect Matches- Không hợp lệ: reset vị trí
Hậu điều kiện	Board được cập nhật trạng thái mới
Ngoại lệ	Ô bị khóa (lock) hoặc không thể swap do rào cản

Bảng 1: Use case Swap

Use Case: Phát hiện Match

Thành phần	Mô tả
Actor	Hệ thống
Trigger	Sau khi swap hoặc fill board
Luồng chính	<ol style="list-style-type: none">1. Quét board theo 2 chiều:<ul style="list-style-type: none">- Chiều ngang: tìm ≥ 3 ô giống nhau liên tiếp- Chiều dọc: tương tự2. Nếu phát hiện match:<ul style="list-style-type: none">- Đánh dấu các ô matched- Nếu match ≥ 4 ô: tạo bonus item
Quy tắc nghiệp vụ	<ul style="list-style-type: none">- Match 3: +30 điểm- Match 4: +50 điểm + tạo Horizontal Bomb- Match 5: +100 điểm + tạo Rainbow Ball

Bảng 2: Use case Match

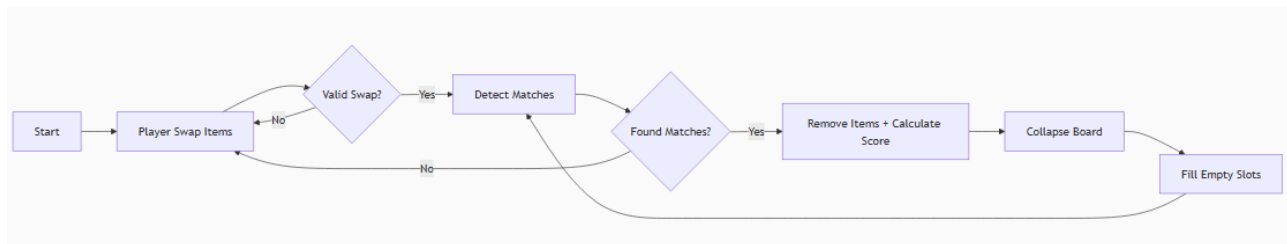
Use Case: Collapse Board

Thành phần	Mô tả
------------	-------

Luồng chính	<ol style="list-style-type: none"> 1. Các ô phía trên ô bị xóa rơi xuống 2. Tạo hiệu ứng di chuyển mượt 3. Đánh dấu vị trí trống cần fill
Extension Points	- Nếu ô rơi xuống tạo match mới → xử lý combo

Bảng 3: Use case Fill Board

2.6 Biểu đồ hoạt động



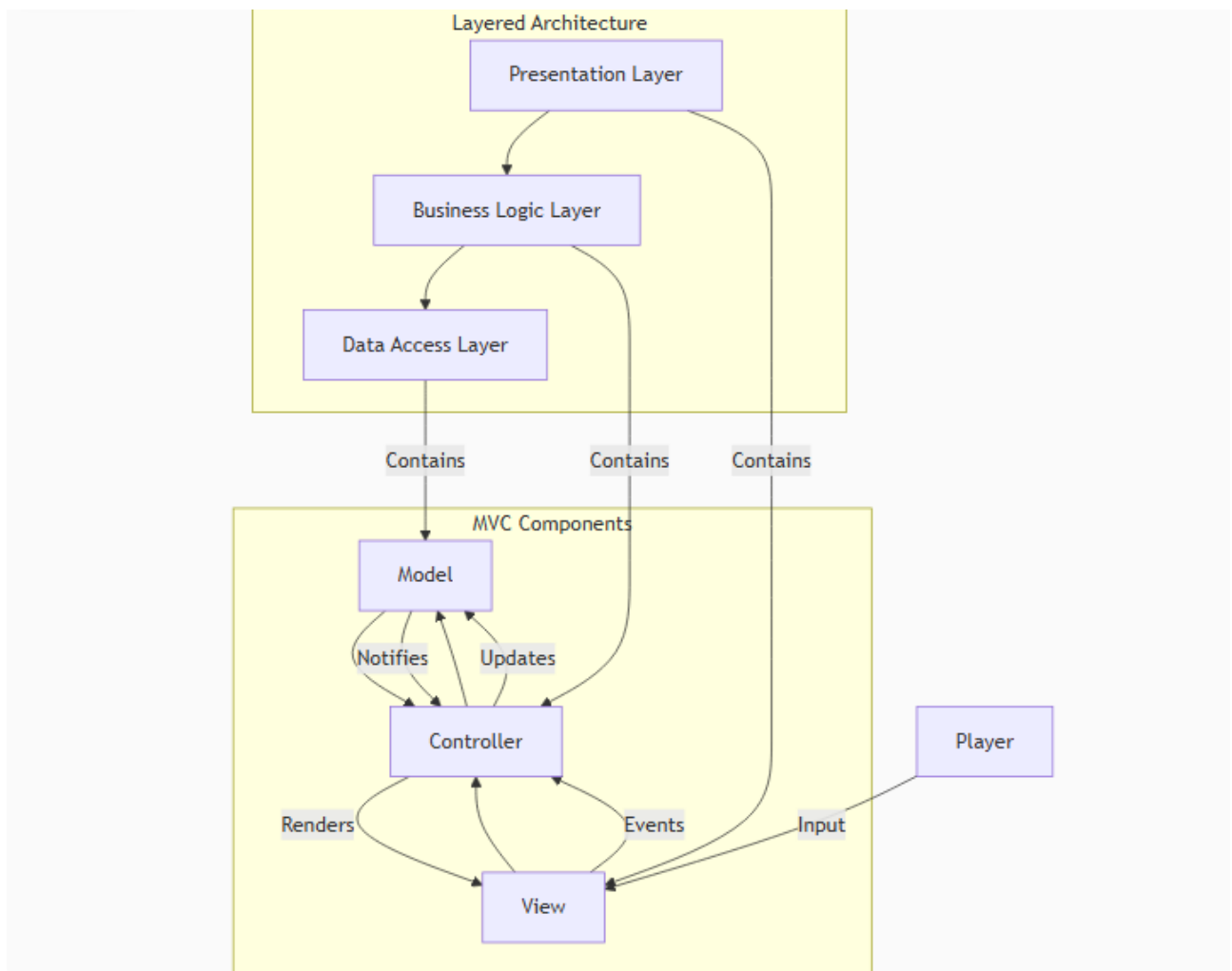
Hình 2: Biểu đồ hoạt động Play Game

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1 Lựa chọn mô hình kiến trúc

Sử dụng mô hình MVC kết hợp kiến trúc Layered.

3.2 Sơ đồ kiến trúc tổng thể



Hình 3: Sơ đồ kiến trúc tổng thể

3.2.1. Data Access Layer (Model)

- Nhiệm vụ:

- Lưu trữ dữ liệu game (trạng thái board, điểm số)
- Xử lý đọc/ghi file (JSON, PlayerPrefs)

3.2.2. Business Logic Layer (Controller)

- Nhiệm vụ:

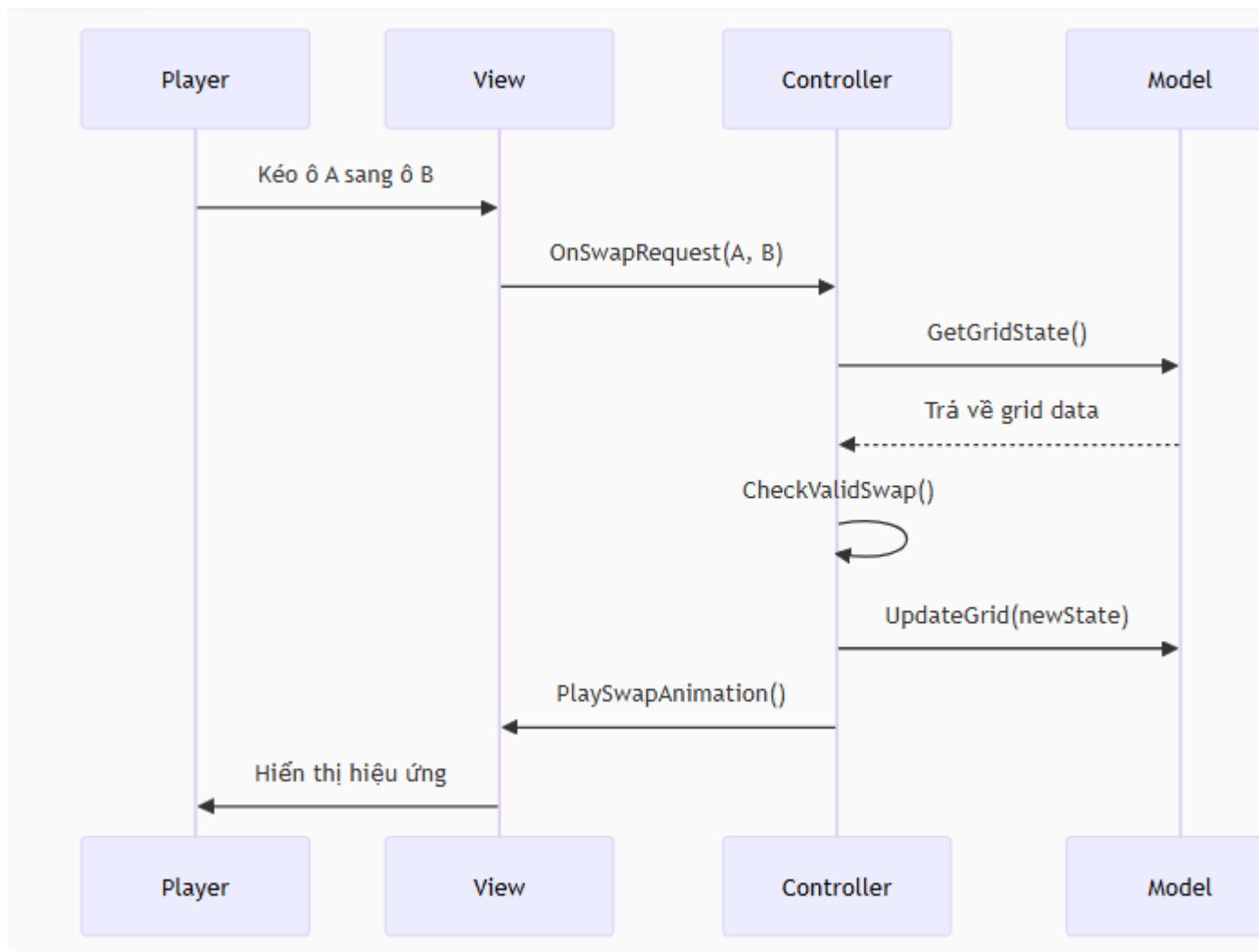
- Logic swap/match/collapse
- Quản lý combo chain, bonus items

3.2.3. Presentation Layer (View)

- Nhiệm vụ:

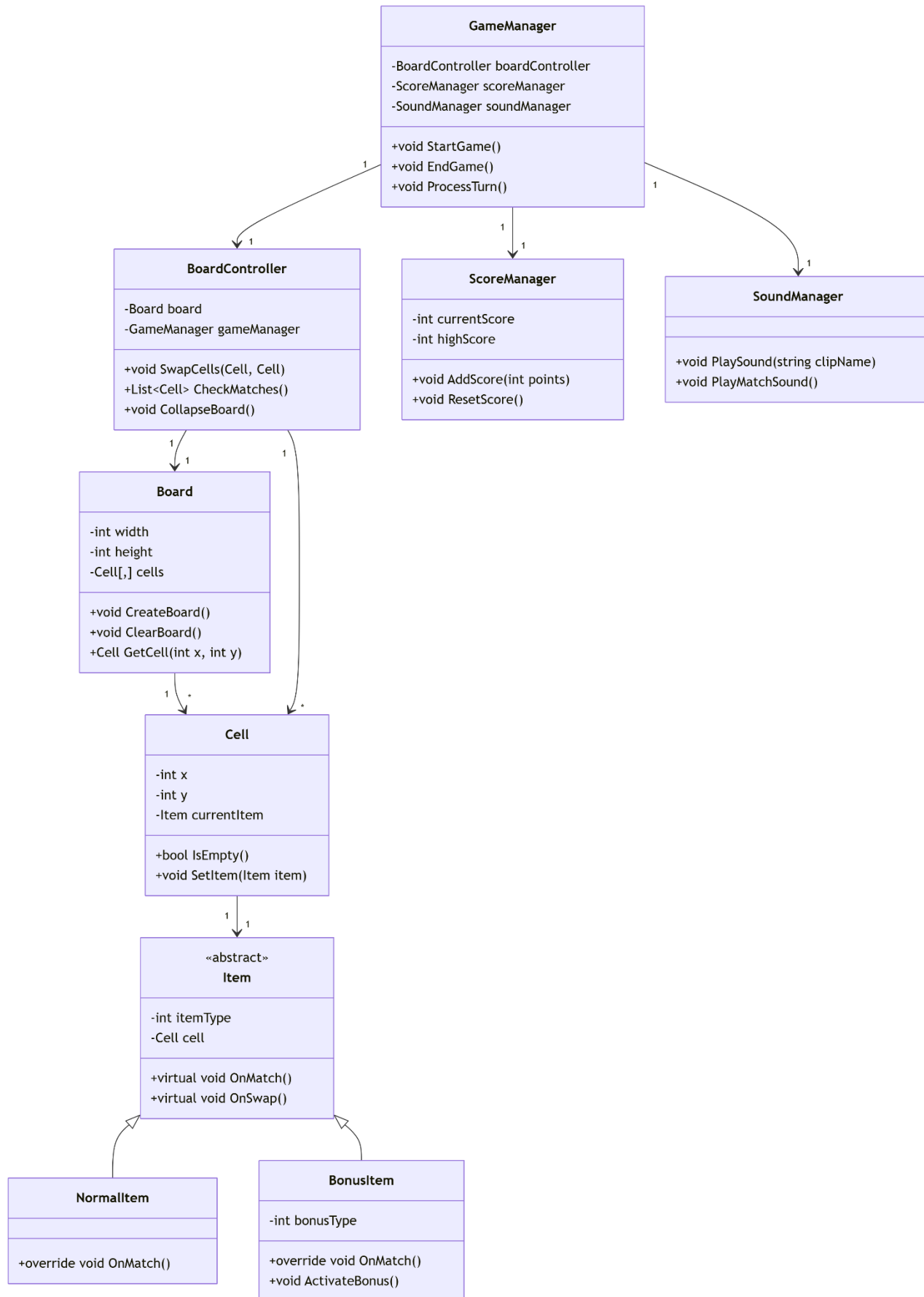
- Hiển thị đồ họa, animation
- Xử lý input (touch, click)

3.3 Thiết kế biểu đồ trình tự



Hình 4: Luồng dữ liệu.

3.4 Thiết kế biểu đồ lớp

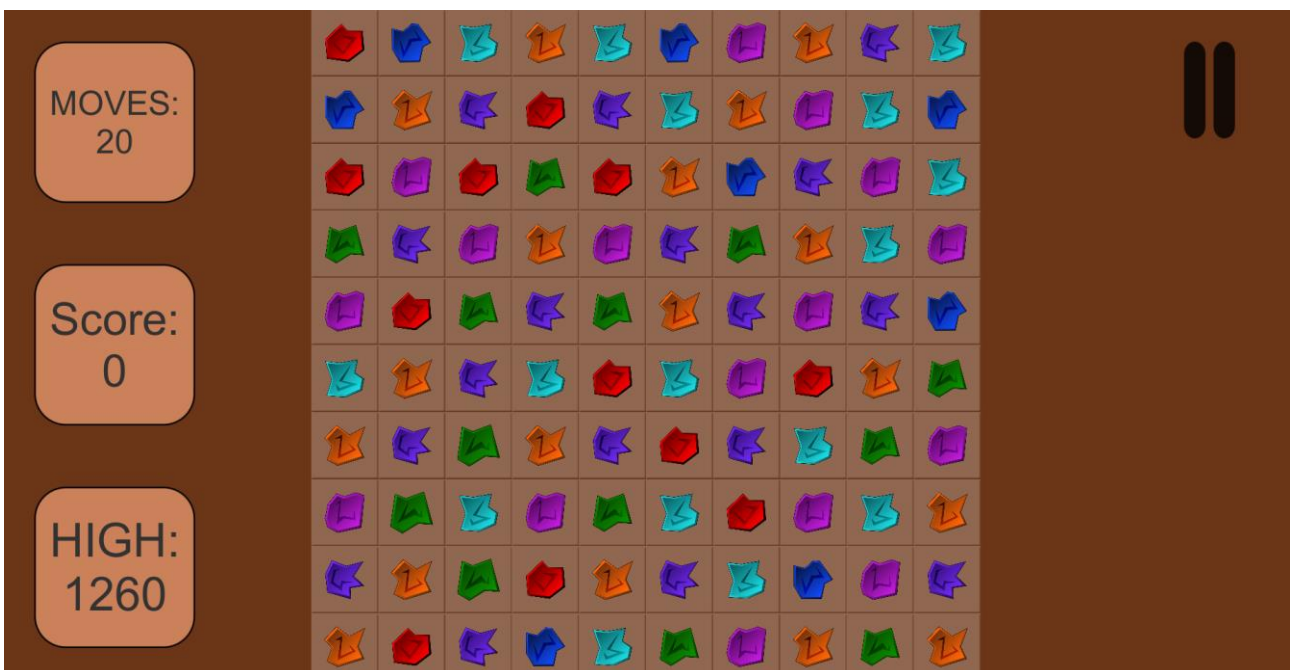


3.5 Thiết kế giao diện

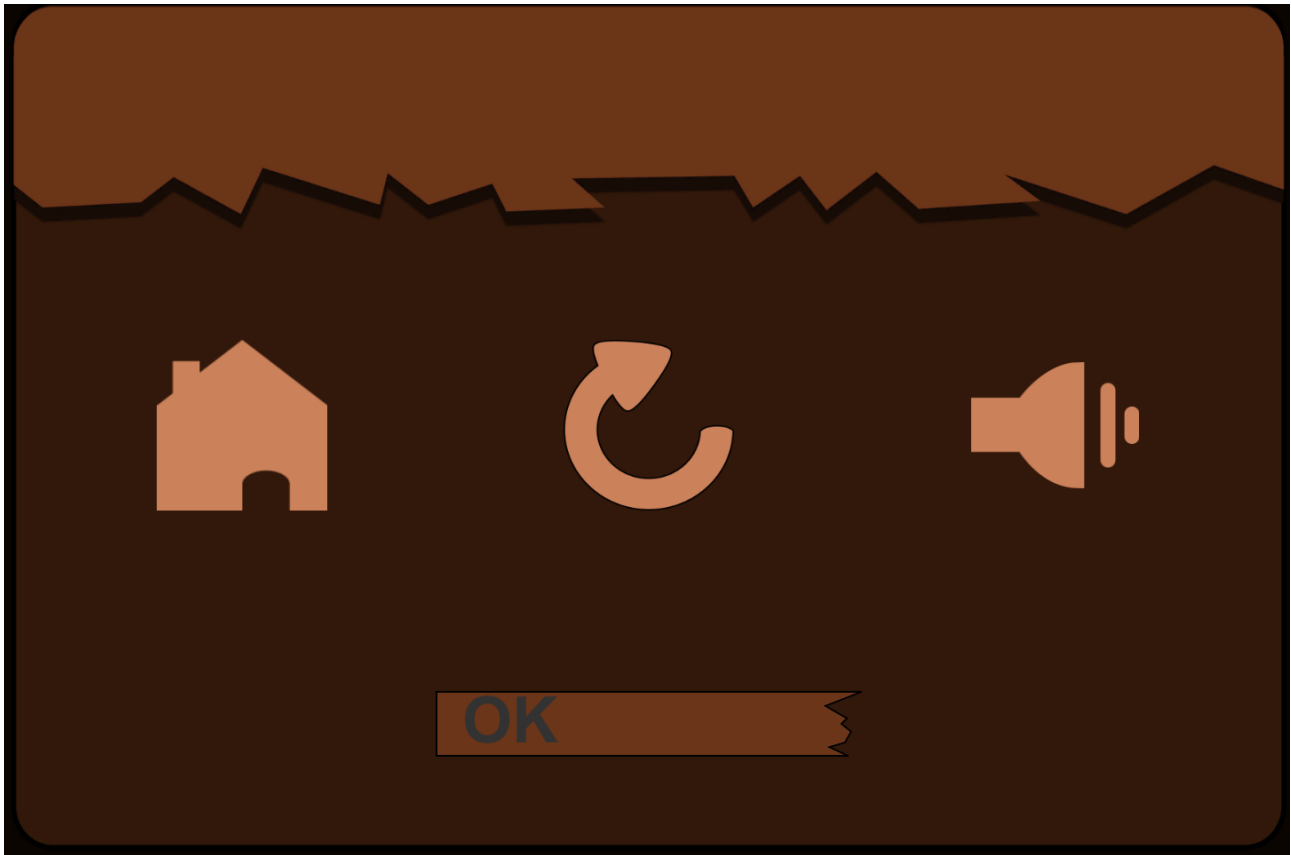
3.6.1 Giao diện người dùng



Hình 5: Giao diện Home



Hình 6: Giao diện Game



Hình 7: Giao diện Pause Game

CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

4.1 Môi trường triển khai

- Unity 2020
- Máy tính Windows

4.2 Cài đặt hệ thống

4.2.1 Yêu cầu hệ thống

- Unity đã cài đặt
- RAM \geq 4GB, CPU i5

4.2.2 Các bước cài đặt chương trình

- Mở project Unity
- Build & Run windows
- Chạy file .exe

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ

5.1 Kết quả thử nghiệm hệ thống

- Các tính năng hoạt động đúng
- Game chạy ổn định

5.2 Đánh giá hiệu quả hệ thống

- Game đơn giản nhưng đầy đủ tính năng
- Có thể mở rộng (Level, booster, UI đẹp hơn...)