# Hệ thống cấp độ

3 level và endless

Level : sử dụng bitmap

Chiều ngang màn chơi: 10 viên gạch, bên phải là điểm cùng mặt của ai đó

Chiều dọc: không quá nửa màn hình

Endless(ưu tiên thấp): level generator

Làm thêm level nếu thừa thời gian

Thắng : phá hết gạch

Thua: khi bị rơi bóng. Có thể cân nhắc thêm một yếu tố: vạch dừng

# Các loại Brick

Ngoại hình: hình chữ nhật

Brick: gạch, kim loại, vàng, kim cương. Tương ứng với lượng máu(health) là 1, 2, 3, 4.

Obsidian: loại brick không thể phá hủy

Explode: sát thương 1, máu 1, diện tích nổ giống hệt quân xe trong cờ vua

TeleportationBrick: máu 1, né nếu đạn đến gần, hồi chiêu 10s

Portal: Dịch chuyển bóng đến cổng còn lại

Shield: bảo vệ bóng mà nó hướng đến, trừ chính nó

# Các loại Power-ups

Để nhận power-ups, mình cần phải tiêu diệt những viên gạch. Viên gạch cấp 1 sẽ có tỉ lệ rơi ra ít nhất, cao nhất là kim cương.

Đạn xuyên gạch: 5s(ai code nên test để điều chỉnh).

Phân thân(Clone bullets): Vĩnh viễn tạo thêm 2 bullet phóng ra từ Paddle

Tăng kích cỡ bàn nảy: 45s, tăng lên 150%

Đạn nổ: nổ khi va chạm, 10s

# Save and load

3 thứ cần lưu:

Array 2D

Trạng thái: vị trí, hướng bay viên đạn

Buff: loại buff và thời gian

Vị trí của thảm

Bảng xếp hạng

# Multiplayer

Làm sau

# Nhiệm vụ cần làm

## Task lần 1

### GameObject và các class sơ khai:

Bình, làm trước trong tối nay. Mọi người hạn là hết thứ 4 tuần sau

### Paddle, Ball: Diễn

Paddle: nhận input người dùng để di chuyển. Paddle là vật bị động, sẽ không kiểm tra va chạm

Đã có file Ball.java trong dự án. Cần kiểm tra va chạm với 4 cạnh màn hình và Paddle, tạm thời chưa cần kiểm tra với Brick

### Brick: Cường

Thiết kế hệ thống bitmap, Brick.java. Brick là vật bị động, sẽ không kiểm tra va chạm

### Power-ups:

tô đức anh, để sau

### Lưu ý khi code

Sử dụng github desktop để download repository của chúng ta về máy mình

Quy định commit: không giới hạn số lần commit, nhưng mỗi lần commit đều phải ghi title, và nội dung(không bắt buộc). Không có quy tắc nào cả, viết gì có ý nghĩa là được

Không được tự ý CHỈNH SỬA code của đồng nghiệp, nếu cần, hãy báo trước

Ngoại lệ đối với hàm main() của GameManager là nơi mọi người có thể CHỈNH SỬA thoải mái, test, logic, tạo object gì đó…. Không động vào gameloop(), GameManager’s contructor, GamePanel…. chỉ duy nhất hàm main()

Kiến thức áp dụng: java cơ bản và xử lý 2D, kế thừa. Để biết một class có nên sử dụng kế thừa hay không, hãy trả lời câu hỏi “class này trong thực tế có sở hữu toàn bộ thuộc tính và hành vi của cha không?”

Khi bắt đầu code, nên khai báo tên các hàm trước, nội dung(logic bên trong) để trống

Nếu muốn test chương trình, cách duy nhất được phép là chạy file **build and run.bat**, chương trình sẽ tự động biên dịch(tất cả file .class sẽ vào folder class). Không dùng button RUN trong IDE, không tự ý chạy lệnh terminal

## Task lần 2

### Nhận xét task trước

Code của diễn tương đối ổn

### Triển khai task mới

Tập trung khai thác triệt để viên gạch

Nguyên tắc: tận dụng code cũ

#### Diễn:

Code animation Tail cho quả bóng

animation Paddle bị đẩy xuống khi bóng va chạm(chỉnh sửa hàm render của GameObject để dịch ảnh)

animation khi brick bị vỡ, các mảnh vụn sẽ bắn ra(random hướng, dùng hàm sin cos để quyết định dx dy, random kích cỡ Particle, setWidth and setHeight, khi Particle mà đi xa thì nó sẽ tự hủy)

Âm thanh cho tất cả các sự kiện: va chạm bóng, gạch vỡ

#### Cường:

BrickGenerator

Portal

#### Tô Đức Anh

Power-up

+ Bóng Strikethrough: sử dụng if-else trong class Ball

+ Tăng kích cỡ bàn nảy: 45s, tăng lên 150%

#### Bình

Vẽ Sprite cho Brick

Vẽ Sprite cho Power up

GameManager: lobby

Pause: khi Pause, tất cả các vật thể sẽ ngừng di chuyển

Nghiên cứu Multiplayer

### Nhận xét Task tuần 2

Diễn: Ở hệ thống Trail lần trước, Diễn đang triển khai lại “GameObject(Trail) trong GameObject(Ball)”. Giải pháp mới là nên coi Trail như một GameObject. Trong hàm update() của Ball thì sẽ tạo gameObject Trail sau mỗi khoảng thời gian rồi nhét nó vào List<GameObject> trong GameInfo

Tên biến thống nhất là camelCase

## Task lần 3

Sau khi nghiên cứu Multiplayer xong, có một thay đổi quan trọng : Vì chế độ Multiplayer sắp ra mắt Paddle giờ đây không còn là Singleton nữa, thế nên Bình đã sửa code ở một số nơi, ví dụ là Ball.java, bỏ thuộc tính private Paddle paddle và va chạm với các Paddle bằng cách duyệt List

Có lẽ đây là lần cuối cùng giao task rồi, nếu có gì thắc mắc, thì nhắn tin lên nhóm luôn nhé hoặc lúc nào đi học lại thì bảo

Comment những hàm quan trọng nhé, vì cũng trong tiêu chí cộng điểm hàm getter/setter hoặc mấy hàm ngắn thì thôi

Khi xóa code ai đó, thông báo cho mọi người trước khi push để không bị conflict

## Diễn

**SaveAndLoad:**

-Trạng thái game: Thực hiện hành động Save khi người chơi ấn Pause rồi Return to Lobby. Những thứ cần lưu  
+Brick: x, y, loại gạch

+Ball: Vị trí bóng và dx, dy của bóng

+Paddle: Vị trí paddle

+Các Power-up hiện có: vị trí, loại

-Phương pháp lưu trạng thái: mỗi GameObject sẽ có một hàm tojson() để xuất file json, lưu vào máy. Đến lúc cần load lại từ bộ nhớ chuyển nó thành Object cần dùng

-Load lại trạng thái game của level cuối cùng đang chơi dở khi người chơi vào màn. Level nào hoàn thành rồi thì thôi

-Khi người chơi thoát game tức ấn quit thì cần phải lưu  
+Level

+Tên người chơi hiện tại

-Khi load vào Lobby:

+Có một dòng chữ xin chào : <Tên người chơi trước đó>

-Cập nhật score mỗi lần chết hoặc qua màn

-Toàn bộ những thứ này sẽ lưu vào file json

## Cường

Animation TNT để có xung kích bắn từ từ

Làm hệ thống Account và TopPlayer

Tạo leaderboard

Tạo UI để chọn level

### Bình

Thay đổi thuật toán collision thành collide theo đường thẳng

Hệ thống multiplayer

### Tô Đức Anh

Nghiên cứu Junit

## Pending

Boss