**Cờ cá ngựa**

**Đợt 1: (1/2025)**

Vì cô yêu cầu tất cả cùng hiểu nên đây sẽ là nơi anh em chia sẻ kiến thức của mình lên nhé, đồng thời cũng là cái sườn để anh em mình dựa vào để làm (nhớ ghi tên phần của mình vào để ai không hiểu thì biết đường hỏi người đó).

(Đức)

1. **Phân tích cấu trúc game:**
   1. Bàn cờ (Board)
   2. Quân cờ (Piece)
   3. Xúc xắc (Dices)
   4. Người chơi (Player)
   5. Logic giữa các lượt chơi: xử lý thứ tự chơi và luật chơi.
   6. Giao diện người chơi (UI): bắt đầu, cài đặt, âm thanh, hướng dẫn chơi (có thể có hoặc không).
   7. Tính năng: xuất hiện ngẫu nhiên ở các ô và có 1 chức năng riêng
2. **Nhiệm vụ (có thể thêm):**
   1. **Bàn cờ (Board)**
      1. Tạo giao diện bàn cờ
      2. Chia bàn cờ thành các ô (cell), trong 4 cell của 4 người chơi thì chia tiếp các cell đường đi
      3. Định nghĩa tọa độ cho từng ô
      4. **Vẽ bàn cờ bằng Java Swing/JavaFX**
   2. **Quân cờ (Piece)**
      1. Định nghĩa thuộc tính: màu sắc, trạng thái (ở chuồng, di chuyển về đích).
      2. Xử lý logic di chuyển quân và animation của chúng.
      3. **Tích hợp bàn cờ vào quân cờ**
   3. **Xúc xắc (Dices)**
      1. Tạo xúc xắc và logic sinh số ngẫu nhiên từ 1-6 (cái này có thể 1 con xúc xắc hoặc 2 con – rcm 2 con để game chơi cho nhanh và có tính đột biến hơn).
      2. Hiển thị kết quả xúc xắc lên giao diện và animation của chúng.
   4. **Logic game và giao diện người chơi (UI & Logic)**
      1. Xử lý lượt chơi : chuyển lượt giữa các người chơi.
      2. Kiểm tra các điều kiện đặc biệt : ra quân, ăn quân, về chuồng (ví dụ xúc ra 2 con 6 cùng lúc mới xuất được 1 con), thắng thua.
      3. Hiển thị thông tin người chơi : Lượt hiện tại, điểm số, kỹ năng (nó khá hay)
   5. **Tính năng (Kỹ năng xuất hiện ngẫu nhiên ở các ô trên bàn cờ) – khá phức tạp**
      1. Màu sắc ở đây là gì ? (có thể góp ý thêm)
         1. Màu đỏ : Hiệu ứng có lợi cho bản thân, mạnh và cần hi sinh 1 cái gì đó để thực hiện
         2. Màu xanh nước biển nhạt : Hiệu ứng có lợi nhưng mà khá yếu
         3. Màu đen : Hiệu ứng bất lợi cho bản thân

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên skill** | **Mô tả** | **Màu sắc** |
| **1** | Bành trướng lãnh địa | Hi sinh 1 quân cờ của bản thân và loại bỏ tất cả các quân cờ của đối phương trước mặt trong 15 ô | Màu đỏ |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |
| **8** |  |  |  |
| **9** |  |  |  |
| **10** |  |  |  |
| **11** |  |  |  |
| **12** |  |  |  |
| **13** |  |  |  |
| **14** |  |  |  |
| **15** |  |  |  |
| **16** |  |  |  |
| **17** |  |  |  |
| **18** |  |  |  |
| **19** |  |  |  |
| **20** |  |  |  |
| **21** |  |  |  |
| **22** |  |  |  |
| **23** |  |  |  |
| **24** |  |  |  |
| **25** |  |  |  |
| **26** |  |  |  |
| **27** |  |  |  |
| **28** |  |  |  |
| **29** |  |  |  |
| **30** |  |  |  |
| **31** |  |  |  |
| **32** |  |  |  |
| **33** |  |  |  |
| **34** |  |  |  |
| **35** |  |  |  |
| **36** |  |  |  |
| **37** |  |  |  |
| **38** |  |  |  |
| **39** |  |  |  |
| **40** |  |  |  |

1. **Các sơ đồ cần xây dựng**
   1. **Sơ đồ lớp**

**A black screen with white text

Description automatically generated**

* 1. **Sơ đồ trình tự**

Chưa có ý tưởng huhu

A diagram of a process

Description automatically generated

Đợt 2: (2/2025)

1. **Luật chơi và điều kiện thắng (Tùng với Tuấn đang làm rồi thì hoàn thiện nhé)**
   1. Công việc
      1. Xác định khi nào một người chơi thắng (tất cả quân cờ về đích – tức là 4 vị trí cuối cùng của mảng = true -> win).
      2. Xử lý 1 số trường hợp khác
         1. Đi đúng số để lên điểm đích (tránh đi thêm vòng nữa)
         2. Đổ 6 sẽ có 2 options là đi quân hoặc xuất quân
         3. Ăn quân
   2. Triển khai
      1. Kiểm tra sau mỗi lần di chuyển xem có ai thắng không (tôi nghĩ nên để if này chạy từ đầu luôn để tránh bị khó khăn)
2. **Xử lý lượt chơi và vòng lặp game (Vì tuấn đang làm logicgame rồi nên phần này nhận luôn cho tiện nhé)**
   1. Công việc:
      1. Xác định người chơi hiện tại
      2. Sau khi move -> chuyển lượt cho người kế tiếp
      3. Khi xúc xắc ra 6, người chơi hiện tại có thể tiếp tục đi tiếp one more
   2. Triển khai
      1. Viết class TurnManager để quản lý lượt chơi.
      2. Cập nhật giao diện để hiển thị người chơi hiện tại (Có nên quay bàn cờ sau mỗi lượt chơi hay không ?)
3. **Tương tác giữa các quân cờ (TÙNG BẠN)**
   1. Công việc:
      1. Nếu quân cờ của “mình” đến vị trí của đối thủ -> kiểm tra điều kiện ăn ? ăn : không ăn được (vì có trường hợp quân đối thủ đang ở điểm xuất phát)
      2. …… (tính năng, có thể thêm ý tưởng)
   2. Triển khai:
      1. Kiểm tra danh sách các quân cờ sau mỗi lần di chuyển (check win, check ăn quân, sau này có thể check kĩ năng nữa,…)
4. **Cải thiện giao diện (Hải Đức)**
   1. Công việc:
      1. Hiển thị lượt chơi hiện tại
      2. Làm animation cho quân cờ di chuyển
      3. Thêm hiệu ứng xúc xắc (xúc lên trời rồi rơi xuống)
      4. Tô sáng các quân cờ có thể di chuyển (**như Tùng nói thì dùng mũi tên tam giác ở trên đầu cũng được nhưng mà tôi thấy không nổi bật, sợ người chơi khó nhìn**)
   2. Triển khai:
      1. Thêm hiệu ứng di chuyển bằng timer hoặc Thread.sleep().
5. **Âm thanh và hiệu ứng (Nguyễn Văn Tú – đi tìm âm thanh và cách triển khai nó)**
   1. Công việc:
      1. Âm thanh khi lắc xúc xắc
      2. Âm thanh khi quân cờ di chuyển
      3. Hiệu ứng khi ăn quân đối thủ
   2. Triển khai
      1. Sử dụng javax.sound.sampled.Clip để phát âm thanh
      2. Thêm animation nhỏ khi quân cờ bị ăn
6. **Màn hình menu và hướng dẫn (Hải Đức nhận)**
   1. Công việc
      1. Màn hình chính
      2. Hướng dẫn cách chơi game để người mới có thể hiểu luật
   2. Triển khai
      1. Tạo màn hình
      2. …
7. **Lưu trạng thái game và tiếp tục chơi (Hải Đức)**
   1. Công việc:
      1. Cho phép lưu game đang chơi và tiếp tục lại sau
      2. Ghi lại thông tin người chơi và vị trí quân cờ
   2. Triển khai
      1. Dùng Java Serialization hoặc ghi ra file JSON/XML để lưu trạng thái
      2. Khi mở game, đọc lại dữ liệu đã lưu
8. **AI đối thủ (Nếu có thời gian)**
   1. Công việc:
      1. Viết thuật toán
      2. AI sẽ tự lăn xúc xắc, chọn quân cờ hợp lí để di chuyển
   2. Triển khai:
      1. Chọn ngẫu nhiên
      2. Nâng cao thì dùng Minimax Algorithm để chọn nước đi tối ưu
9. **Kiểm thử và sửa lỗi (Hải Đức)**
   1. Công việc:
      1. Kiểm tra tất cả quy tắc di chuyển đã đúng chưa
      2. Xử lý trường hợp đặc biệt đã đúng chưa
      3. Chạy game với anh em để tìm lỗi
   2. Triển khai:
      1. Viết test case để kiểm tra các chức năng
      2. Ghi lại lỗi phát hiện và sửa chúng