**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC PHENIKAA**

\*\*\*\*\*



**VUONG\_COFFEE**

|  |  |
| --- | --- |
| **Course/Học phần** | : KỸ THUẬT PHẦN MỀM |
| **Code/Mã học phần** | : CSE702025 |
| **Lớp/Class**  **Instructor/Giảng viên**  **Group/Nhóm** | :COUR01.LT5  : Trịnh Thanh Bình  : 9  HÀ NỘI – 2025 |

MỞ ĐẦU

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng tự động hóa trong các trò chơi trực tuyến ngày càng trở nên phổ biến, nhằm nâng cao trải nghiệm người chơi và tối ưu hóa thời gian. Roblox với nền tảng đa dạng và cộng đồng người chơi rộng lớn, đã trở thành môi trường lý tưởng để phát triển các ứng dụng hỗ trợ người chơi.

Game **Anime Vanguard** trên Roblox là một tựa game chiến thuật hấp dẫn, thu hút người chơi bởi hệ thống nhân vật phong phú và các thử thách đa dạng. Tuy nhiên, để đạt được tiến độ trong game và thu thập tài nguyên như Gems, Gold Coins hay hoàn thành các nhiệm vụ (Bounties), người chơi cần dành nhiều thời gian và công sức. Điều này gây khó khăn cho những người chơi không có nhiều thời gian rảnh hoặc muốn tối ưu hóa hiệu quả chơi mà không phải can thiệp liên tục hoặc đơn giản là do lười.

Để giải quyết vấn đề này, chúng tôi đã phát triển một **hệ thống tự động hóa** dành riêng cho người chơi Anime Vanguard giúp tự động thực hiện các thử thách trong game, tự động farm tài nguyên, nâng cấp và quản lý các đơn vị trong game một cách hiệu quả. Hệ thống không chỉ giúp tiết kiệm thời gian mà còn nâng cao hiệu quả chơi, phù hợp với những người chơi bận rộn hoặc đơn giản là muốn trải nghiệm game thoải mái hơn.

Báo cáo này trình bày chi tiết về quá trình phát triển hệ thống, bao gồm phân tích yêu cầu, thiết kế kiến trúc và giao diện, quản lý dự án theo phương pháp Scrum, triển khai và kiểm thử, cũng như đánh giá kết quả và đề xuất hướng phát triển trong tương lai.

# GIỚI THIỆU ĐỀ TÀI

## Bối cảnh và vấn đề đặt ra

Trong thế giới game trực tuyến hiện đại, việc tối ưu hóa thời gian chơi và tài nguyên là nhu cầu thiết yếu của nhiều người chơi, đặc biệt là đối với các trò chơi chiến thuật như **Anime Vanguard** trên nền tảng Roblox. Trò chơi này yêu cầu người chơi phải dành nhiều thời gian để tham gia các thử thách, thu thập tài nguyên và nâng cấp nhân vật. Tuy nhiên, không phải ai cũng có đủ thời gian hoặc kiên nhẫn để thực hiện các nhiệm vụ này một cách thủ công.

Đối tượng người chơi chủ yếu là những người bận rộn, không có nhiều thời gian để cày cuốc, hoặc đơn giản là muốn tối ưu hóa trải nghiệm chơi game mà không phải can thiệp quá nhiều. Họ cần một giải pháp tự động hóa, giúp tiết kiệm thời gian và công sức, đồng thời vẫn đạt được hiệu quả cao trong trò chơi.

## Mục tiêu và phạm vi dự án

### Mục tiêu

Dự án này hướng đến việc phát triển một hệ thống tự động hóa toàn diện cho game **Anime Vanguard**, nhằm:

* **Tiết kiệm thời gian**: Giúp người chơi hoàn thành các nhiệm vụ và thu thập tài nguyên mà không cần phải trực tiếp tham gia.
* **Tối ưu hóa tài nguyên**: Tự động hóa quá trình thu thập và sử dụng tài nguyên trong game, giúp người chơi đạt hiệu quả cao mà không tốn nhiều công sức.
* **Cải thiện trải nghiệm người chơi**: Mang đến một trải nghiệm chơi game mượt mà, liên tục và không gián đoạn, phù hợp với những người chơi bận rộn hoặc muốn thư giãn.

### Phạm vi

Phạm vi bao gồm: phần mềm chạy trên Windows với GUI trực quan, tích hợp với trò chơi Roblox để thực hiện các hành động trong game với các mục chính:

* + Phát triển hệ thống tự động tham gia các thử thách trong game.
  + Tự động hóa quá trình thu thập tài nguyên và nâng cấp nhân vật.
  + Cung cấp giao diện người dùng thân thiện và dễ sử dụng.
  + Đảm bảo tính ổn định và hiệu suất cao của hệ thống.

## Đối tượng người dùng

Đối tượng chính của hệ thống là:

* + **Người chơi game Anime Vanguard**: Những người yêu thích trò chơi nhưng không có nhiều thời gian để tham gia trực tiếp.
  + **Người chơi muốn tối ưu hóa trải nghiệm**: Những người muốn đạt được hiệu quả cao trong game mà không cần phải dành quá nhiều thời gian.
  + **Người chơi mới**: Những người mới bắt đầu chơi và cần một công cụ hỗ trợ để làm quen với trò chơi.

## Công nghệ sử dụng

Phần mềm được phát triển bằng **AutoHotkey v2** — ngôn ngữ lập trình mã nguồn mở chuyên dành cho tự động hóa trên AutoHotkey v2 cho phép thiết lập **phím tắt**, **macro**, tự động kích chuột và nhập phím, hoàn toàn phù hợp với mục tiêu tự động hóa nhiệm vụ trong game.

## Phương pháp phát triển và tổ chức nhóm

Dự án áp dụng phương pháp **Scrum**, tổ chức theo các Sprint 1–2 tuần để đảm bảo tính linh hoạt và tiến độ liên tục với các thành viên chính:

|  |  |  |  |
| --- | --- | --- | --- |
| Thành viên | MSSV | Vai trò | Phân công |
| Đào Mạnh Vương | 23010586 | PO/Scum Master/Dev |  |
| Nguyễn Hữu Hiển |  | Scrum Master/Dev |  |
| Nguyễn Văn Chiến |  | Scrum Master/Dev |  |
| Trần Trọng Minh |  | Dev |  |

# Tổng quan về mô hình Scrum

## Giới thiệu Scrum

Scrum là một mô hình phát triển phần mềm thuộc nhóm phương pháp Agile, tập trung vào việc phát triển linh hoạt, thích ứng nhanh với thay đổi và liên tục cải tiến. Scrum thường được sử dụng trong các dự án có yêu cầu thay đổi thường xuyên hoặc không thể xác định đầy đủ ngay từ đầu, phù hợp với đặc điểm của các dự án game tool như **tool tự động cho Roblox: Anime Vanguard** – nơi tính năng, map, hay cơ chế game có thể thay đổi theo từng bản cập nhật.

Scrum hướng đến việc chia nhỏ quá trình phát triển thành các giai đoạn ngắn gọi là **Sprint** trong đó đội ngũ sẽ tập trung hoàn thành một phần chức năng có thể sử dụng hoặc kiểm thử được. Qua từng Sprint sản phẩm sẽ dần được hoàn thiện với sự phản hồi và cải tiến liên tục.

## Lý do chọn Scrum

Scrum được lựa chọn vì phù hợp với đặc điểm của nhóm và tính chất của dự án:

* + **Thời gian phát triển ngắn**: Scrum cho phép chia nhỏ công việc và có sản phẩm khả dụng sau từng Sprint.
  + **Game có tính chất update liên tục:** Game được cập nhập mỗi 2 tuần thêm event mới, thêm map mới, các unit và meta khác nhau.
  + **Đội nhóm chưa có nhiều kinh nghiệm làm việc nhóm**: Scrum giúp xác định rõ vai trò, quy trình và cách phối hợp giữa các thành viên.
  + **Dự án gồm nhiều tính năng nhỏ, độc lập**: Việc chia backlog và chọn task phù hợp cho từng Sprint giúp dễ quản lý và phát triển.
  + **Cần kiểm thử từng chức năng riêng biệt**: Scrum cho phép kiểm tra từng phần nhỏ (auto farm, detect map, webhook,…) ngay sau khi hoàn thành.
  + **Có thể thay đổi và cải tiến liên tục**: Khi người dùng hoặc tester có phản hồi, nhóm có thể linh hoạt cập nhật backlog và thay đổi kế hoạch phát triển.
  + **Khuyến khích học tập và cải tiến sau mỗi Sprint**: Qua Sprint Retrospective, nhóm có thể nhìn lại và nâng cao kỹ năng làm việc nhóm theo thời gian.

## Các vai trò chính

### Product Owner (PO)

Là người định hướng và xác định yêu cầu, tính năng cần có của sản phẩm. Product Owner là người làm việc trực tiếp với người dùng cuối hoặc thị trường để xây dựng và quản lý danh sách Product Backlog – những tính năng cần phát triển.

Trong dự án này, **Đào Mạnh Vương** giữ vai trò Product Owner. Vương là người đề xuất ý tưởng ban đầu và xác định các tính năng quan trọng như: auto challenge, nhận diện map, GUI thống kê, gửi webhook thông báo…

### Scrum Master

Là người đảm bảo nhóm phát triển làm việc đúng với quy trình Scrum, hỗ trợ loại bỏ các rào cản, đảm bảo tính liên kết và tiến độ của cả nhóm.

Trong dự án này, vai trò Scrum Master được chia sẻ giữa các thành viên:**Đào Mạnh Vương, Nguyễn Hữu Hiển, Nguyễn Văn Chiến.**

Các bạn thay phiên hỗ trợ nhóm trong việc lập kế hoạch Sprint, tổ chức Daily Scrum và theo dõi tiến độ hoàn thành các task.

### Development Team

|  |  |
| --- | --- |
| Thành viên | Nhiệm vụ chính |
| Đào Mạnh Vương | Viết script auto challenge, auto bounty, phân tích màn chơi, thiết kế GUI, hệ thống tự Reconnect, Viết script các màn chơi |
| Nguyễn Hữu Hiển | Viết script auto đặt và nâng cấp units, target priority, xử lí cập nhập thông tin lên GUI, Viết script các màn chơi |
| Nguyễn Văn Chiến | Xây dựng hệ thống thống kê và điều khiển, cải tiến hệ thống bấm, chức năng tìm vị trí chính xác, Viết script các màn chơi |
| Trần Trọng Minh | Hệ thống phát hiện map, màn hình, chụp ảnh màn hình |

# Phân tích yêu cầu hệ thống

## TỔNG QUAN HỆ THỐNG

### Mục Đích Phát Triển

Auto Challenge Tool là một công cụ hỗ trợ chơi game Roblox được thiết kế để tự động hóa toàn bộ quá trình chơi, đặc biệt trong các game dạng "tower defense" hoặc có yếu tố chiến thuật lặp đi lặp lại. Công cụ này được lập trình bằng ngôn ngữ AutoHotkey phiên bản 2 (AHK v2), nhằm mô phỏng các thao tác của người dùng và tự động hóa các hành vi lặp lại trong game như farm tài nguyên, hoàn thành nhiệm vụ và tối ưu hóa hiệu suất người chơi.

Mục đích chính của hệ thống là:

* **Tăng cường trải nghiệm người chơi** bằng cách giảm thiểu thao tác thủ công, tránh lãng phí thời gian và gây nhàm chán vì các nhiệm vụ lặp đi lặp lại.
* **Tối ưu hóa tài nguyên trong game** có thể tự động đi các thử thách để thu thập tài nguyên giúp người chơi, không lãng phí tài nguyên khi có việc, bận không thể tham gia thu thập.
* **Tự động hóa quá trình chơi game** có thể tự động làm mọi thứ, không cần thao tác thủ công từ đó tiết kiệm thời gian.
* **Cung cấp giao diện thân thiện** giúp người dùng có thể dễ dàng tùy chỉnh và điều khiển theo nhu cầu cá nhân.
* **Cơ chế sử dụng dễ dàng:** có thể dễ dàng sử dụng qua các thao tác đơn giản như bấm f2 để bắt đầu

### Phạm Vi Áp Dụng

Auto Challenge Tool hoạt động như một hệ thống hỗ trợ tổng hợp, bao gồm các module chức năng như:

* **Tự động hóa các hành động trong game** như chọn map, nhận nhiệm vụ, đặt và nâng cấp unit.
* **Theo dõi và thống kê dữ liệu tài nguyên** mà người chơi thu thập được trong thời gian thực.
* **Gửi thông báo trạng thái, lỗi hoặc thành tích** thông qua Discord Webhook.
* **Cung cấp giao diện đồ họa người dùng (GUI)** để kiểm soát macro, cấu hình thông số và hiển thị log hoạt động.
* **Hỗ trợ nhiều map và chế độ chơi**, có khả năng thích nghi với các bản cập nhật game.

## YÊU CẦU CHỨC NĂNG

### Giao Diện Thống Kê (GUI Stats Dashboard)

Auto Challenge Tool tích hợp một giao diện thống kê chuyên sâu, cung cấp dữ liệu toàn diện và trực quan về mọi hoạt động trong game, giúp người dùng dễ dàng giám sát tiến trình farm, tài nguyên thu được, và trạng thái hệ thống macro.

#### Hiển Thị Thời Gian Chạy Macro

* + Hệ thống sẽ tự động kích hoạt bộ đếm thời gian ngay khi người dùng khởi động macro.
  + Thời gian được hiển thị theo định dạng quốc tế chuẩn: Giờ:Phút:Giây (HH:MM:SS), cập nhật liên tục với độ trễ gần như bằng 0.
  + Đồng hồ thời gian thực giúp người dùng đo lường hiệu suất chạy macro, ước lượng thời gian hoàn thành một phiên chơi, hoặc lập kế hoạch thời gian farm.

#### Thống Kê Tài Nguyên

* + **Gems:** Thống kê tổng số đá quý người chơi đã thu thập được kể từ khi khởi chạy macro.
  + **Gold Coins:** Ghi nhận số lượng vàng tích lũy, bao gồm từ nhiệm vụ, chiến đấu hoặc phần thưởng map.
  + **Trait Rerolls:** Hiển thị số lượt người dùng đã thực hiện thay đổi trait hoặc kỹ năng của nhân vật/unit.
  + **Bounties Completed:** Tổng hợp số nhiệm vụ đặc biệt hoặc bounty đã hoàn thành để đánh giá hiệu quả và tiến độ.

#### Activity Log (Nhật ký hoạt động)

* + Hiển thị dạng dòng thời gian các sự kiện gần nhất: đặt unit, đổi map, hoàn thành nhiệm vụ, báo lỗi,...
  + Tự động cuộn xuống khi có bản ghi mới xuất hiện, giúp theo dõi dễ dàng mà không cần thao tác thủ công.
  + Lưu trữ toàn bộ nhật ký hoạt động vào file .log (kèm theo timestamp) để tiện kiểm tra, phân tích lại hoặc chia sẻ lỗi cho developer.

### Tự Động đi Challenge (Auto Challenge System)

Module này xử lý toàn bộ chuỗi hoạt động từ nhận, chọn lọc đến thực thi nhiệm vụ một cách hoàn toàn tự động, giúp người chơi tối đa hóa tài nguyên mà không cần can thiệp thủ công.

#### Quản Lý Nhiệm Vụ

* + Tự động kiểm tra danh sách nhiệm vụ có sẵn theo chu kỳ và nhận nhiệm vụ mới ngay khi có thể.
  + Hệ thống sử dụng thuật toán đánh giá độ khó (dựa trên mô tả nhiệm vụ) và gán nhãn (Challenge, Bounty, một số map đặc biệt).
  + Ưu tiên xử lý các nhiệm vụ có phần thưởng giá trị cao hoặc có tính giới hạn thời gian như challenge half hours.

#### Xử Lý Thử Thách

* + Phân tích logic map, đặc điểm kẻ địch boss, các map khó sẽ sử dụng kĩ năng đặc biệt…
  + Dựa trên từng map sẽ có chiến thuật khác nhau: chỉ đặt dps chính, đặt farm trước khi đặt dps, nâng cấp farm trước ở những map khó
  + Khi thất bại sẽ tự động từ bỏ map và đợi đến challenge tiếp theo thay vì mất thời gian vào một bản đồ quá khó.

### Tự Động Đặt Và Nâng Cấp Units

#### Chiến Thuật Đặt Unit

* Tự động xác định vị trí tối ưu cho trước để tiến hành phòng thủ
* Sử dụng tọa độ, công cụ di chuyển chuột chính xác để đặt tháp và phòng thủ
* Tùy từng map sẽ ưu tiên nâng dps hoặc nâng farm nhằm tối ưu tỉ lệ thắng và thời gian
* Nhận biết map khó để sử dụng các kĩ năng hoặc unit đặc biệt như: bắn toàn map, vô hiệu hóa kĩ năng…

#### Chiến Lược phân loại

* + Phân loại đơn vị thành nhóm: **DPS chính**, **Hỗ trợ**, **Farm**, **AoE**, **CC**.
  + Thực hiện nâng cấp theo cấu trúc ưu tiên đã cấu hình (ví dụ: nâng cấp DPS 3 lần trước khi nâng support).
  + Tự điều chỉnh thứ tự nâng cấp dựa trên map.

#### Auto Upgrade

* + Tự động nâng cấp farm units ở wave 1-3 để tối đa hóa lượng tài nguyên thu được về sau.
  + Khi tài nguyên đủ, ưu tiên nâng DPS hoặc những unit có khả năng gây sát thương diện rộng.
  + Hệ thống có logic kiểm tra dư tài nguyên để tránh upgrade sai mục tiêu hoặc nâng cấp vượt ngưỡng cần thiết.

### Gửi Thông Báo Qua Discord Webhook

#### Cấu Hình Webhook

* Cho phép người dùng nhập và lưu URL webhook Discord, đi kèm user ID nếu muốn gửi riêng cho một cá nhân.
* Thiết lập dạng thông báo cần nhận, ví dụ:
* Chỉ báo lỗi
* Chỉ khi hoàn thành challenge
* Thông báo theo chu kỳ (mỗi 30 phút)

#### Loại Thông Báo

* Gửi thông báo dạng embed đẹp mắt với thông tin:
  + Tên map, wave hoàn thành, thời gian hoàn thành
  + Tổng số gems/vàng kiếm được
  + Lỗi phát sinh và số lần retry
  + Thống kê cuối mỗi phiên chơi

#### Hệ Thống Macro Phím Tắt

|  |  |
| --- | --- |
| Phím Tắt | Chức Năng |
| F1 | Căn chỉnh lại vị trí cửa sổ Roblox để macro hoạt động chính xác theo tọa độ chuẩn |
| F2 | Kích hoạt toàn bộ hệ thống macro, bao gồm auto farm, auto đặt unit, nâng cấp |
| F3 | |  | | --- | |  |  |  | | --- | | Tắt hoàn toàn macro và dọn dẹp bộ nhớ tạm | |
| F4 | Tạm dừng hoặc tiếp tục macro, sử dụng trong trường hợp người dùng muốn can thiệp thủ công tạm thời |

## YÊU CẦU PHI CHỨC NĂNG

### Tính Ổn Định và Khả Năng Chịu Lỗi

* Sử dụng cơ chế bắt lỗi nâng cao: kiểm tra disconnect.
* Khi phát hiện lỗi, thực hiện reconnect và ghi vào logs, thông báo webhook đến người dùng.
* Log lỗi chi tiết với timestamp, loại lỗi, điều kiện phát sinh, để hỗ trợ debugging nhanh chóng.

### Giao Diện Người Dùng (UI/UX)

* Thiết kế theo mô hình Material UI hoặc Fluent Design.
* Sử dụng hiệu ứng visual nhẹ, chuyển cảnh mượt, màu sắc dịu mắt nhưng rõ ràng.
* Có xác nhận khi thao tác quan trọng (bắt đầu/dừng macro, xoá cấu hình).
* Tương thích màn hình độ phân giải cao (Full HD, 2K).

### Phản Hồi Nhanh và Hiệu Năng Cao

* Đảm bảo thời gian phản hồi nhanh cho các thao tác cơ bản.
* Áp dụng cơ chế ngắt, countinue khi đạt các điều kiện cụ thể.
* Giao diện luôn được cập nhật real-time mà không giật lag hoặc chậm trễ.

### Khả Năng Mở Rộng và Duy Trì

* Kiến trúc Module: từng thành phần (auto đặt unit, auto challenge, webhook, UI, pixel scan) hoạt động độc lập.
* Có khả năng hỗ trợ plugin mở rộng như thêm unit, tích hợp AI đặt chiến thuật,...
* Dễ dàng update và điều chỉnh khi có update, cập nhập map mới

# Scrum Artifact

### Product Backlog

#### Epic 1: Phân Tích & Yêu Cầu Game

[Ưu tiên: Cao] Phân tích lối chơi và cơ chế game:

* Nghiên cứu hệ thống điều khiển nhân vật
* Phân tích cơ chế đặt units và nâng cấp
* Xác định các loại nhiệm vụ và phần thưởng

[Ưu tiên: Cao] Nghiên cứu bản đồ và thử thách

* Lập danh sách tất cả các map trong game
* Phân tích đặc điểm và yêu cầu của từng map
* Xác định vị trí đặt units tối ưu

[Ưu tiên: Cao] Phân tích hệ thống chống gian lận

* Tìm hiểu cơ chế phát hiện auto
* Nghiên cứu giới hạn tốc độ thao tác
* Xác định các hành vi có thể bị phát hiện

#### Epic 2: Hệ Thống Cốt Lõi

[Ưu tiên: Quan trọng] Module điều khiển

* Xây dựng hệ thống mô phỏng chuột
* Phát triển mô phỏng bàn phím
* Tối ưu tốc độ và độ chính xác

[Ưu tiên: Quan trọng] Hệ thống nhận diện

* Module quét màn hình
* Nhận diện các nút và menu
* Phát hiện trạng thái game

#### Epic 3: Tính Năng Tự Động

[Ưu tiên: Cao] Auto farm các bản đồ

[Ưu tiên: Cao] Tự động đi challenge và bounty

* Auto đi khi được reset
* Tự động hoàn thành
* Thu thập phần thưởng

#### Epic 4: Giao Diện Người Dùng

[Ưu tiên: Cao] Cửa sổ điều khiển chính

* Nút bắt đầu/dừng
* Hiển thị trạng thái
* Cài đặt cơ bản

[Ưu tiên: Cao] Giao diện cài đặt

* Cài đặt Discord webhook
* Cài đặt private server
* Tùy chọn nâng cao

#### Epic 5: Tính Năng Tích Hợp

[Ưu tiên: Cao] Tích hợp Discord webhook

* Gửi thông báo kết quả
* Báo cáo lỗi
* Theo dõi tài nguyên

#### Epic 6: Kiểm Thử

[Ưu tiên: Cao] Xây dựng bộ test

* Test các tính năng cơ bản
* Test các trường hợp đặc biệt
* Test hiệu năng
* Tiêu Chí Hoàn Thành
* Đạt yêu cầu về hiệu năng
* Không còn lỗi nghiêm trọng
* Discord webhook hoạt động tốt
* Đã xử lý các trường hợp lỗi

#### Yếu Tố Rủi Ro

* Game cập nhật làm hỏng tính năng
* Bị phát hiện bởi anti-cheat
* Vấn đề về hiệu năng
* Thay đổi API Discord
* Bố cục map phức tạp
* Yêu cầu tài nguyên cao