**Kế Hoạch Bảo Trì Phần Mềm**

**cho**

**Game Đánh bài Summoner**

**Phiên bản 1.0**

**Được phê chuẩn**

**Ts.Huỳnh Quang Nghi**

**Được chuẩn bị bởi**

**Phan Thanh Liêm B1400701**

**Nguyễn Quang Phong B1400717**

**Lê Nguyên Thức B1400731**

**11/04/2018**

**MỤC LỤC**

[**1. Giới thiệu 3**](#_Toc499839153)

[**2. Tài liệu tham khảo 3**](#_Toc499839154)

[**3. Các định nghĩa 3**](#_Toc499839155)

[**4. Tổng quan về bảo trì phần mềm 3**](#_Toc499839156)

[**4.1 Tổ chức bảo trì 3**](#_Toc499839157)

[**4.2 Kế hoạch làm việc 4**](#_Toc499839158)

[**4.3 Tóm tắt tài nguyên 5**](#_Toc499839159)

[**4.4 Công cụ, kỹ thuật và phương pháp 6**](#_Toc499839160)

[**5. Quy trình bảo trì phần mềm 8**](#_Toc499839161)

[**5.1 Ngày đặt hàng trước ngày hiện tại. 8**](#_Toc499839162)

[**5.2 Khôi phục tài khoản khách hàng 9**](#_Toc499839163)

[**5.3 Gởi mail xác nhận đặt hàng. 9**](#_Toc499839164)

[**5.4 Hỗ trợ song ngữ (Việt - Anh) 9**](#_Toc499839165)

[**5.5 Hỗ trợ mua hàng online 10**](#_Toc499839166)

[**5.6 Hỗ trợ thanh toán online 10**](#_Toc499839167)

[**5.7 Xuất hóa đơn điện tử 10**](#_Toc499839168)

[**5.8 Hỗ trợ chức năng thống kê 11**](#_Toc499839169)

[**5.9 Hỗ trợ nhập sản phầm bằng file excel. 11**](#_Toc499839170)

[**5.10 Mã hóa mật khẩu MD5 11**](#_Toc499839171)

[**5.11 Số lượng truy cập đồng thời vào webite hạn chế 11**](#_Toc499839172)

[**5.12. Phục hồi dữ liệu và ghi log sau khi gặp sự cố 12**](#_Toc499839173)

[**6. Quản lý cấu hình 12**](#_Toc499839174)

[**6.1 Quy trình quản lý sự thay đổi 12**](#_Toc499839175)

[**6.2 Quy trình kiểm soát và cập nhật hệ thống 13**](#_Toc499839176)

[**7. Quản lý chất lượng 13**](#_Toc499839177)

# Giới thiệu

* + Tên phần mềm được bảo trì: Game đánh bài Summoner
  + Mục đích, mục tiêu bảo trì phần mềm:
* Cập nhật phần mềm
* Cải tiến phần mềm
* Hiệu chỉnh các điểm gây mất cân bằng
  + Phạm vi bảo trì phần mềm
    - Bảo trì cải tiến: bao gồm các cải tiến mang lại tính mới cho trò chơi.
    - Bảo trì thích ứng: là các bổ sung nhằm tăng tính thích ứng trên các môi trường sử dụng khác nhau.
    - Bảo trì hoàn thiện: tăng tính cân bằng trong trò chơi, cho phép gỡ bài và thêm giao diện trong trò chơi.
    - Bảo trì dự phòng: thay đổi phương thức nạp trò chơi

# Tài liệu tham khảo

* + Danh sách các tài liệu được tham khảo trong kế hoạch này: Giáo trình Bảo trì phần mềm, Huỳnh Xuân Hiệp, Phan Phương Lan. Nhà xuất bản Đại học Cần Thơ, 2014.

# Các định nghĩa

* + Các từ viết tắt, các thuật ngữ, các ký hiệu kỹ thuật mà chúng không được sử dụng một cách phổ biến

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ / Từ viết tắt** | **Định nghĩa / Mô tả** |
| 1 | RCS | Revision Control System |

# Tổng quan về bảo trì phần mềm

## 4.1 Tổ chức bảo trì

* + - Các chức năng cần được bảo trì là:
* Thêm giao diện preloading
* Thêm chức năng tháo bài
* Thêm một số thẻ bài mới
* Bổ sung kết thúc trò chơi khi hết bài
* Tinh chỉnh, cân bằng giá trị các thẻ bài
* Thay đổi đồ họa cho chỉ số HP trong trò chơi
* Thay đổi phương thức nạp và chọn thẻ bài.
  + - Cách thức giao tiếp trong hoạt động bảo trì phần mềm: trao đổi trực tiếp và email.
    - Những người thực hiện bảo trì bao gồm: Phan Thanh Liêm, Nguyễn Quang Phong, Lê Nguyên Thức.
    - Trách nhiệm từng cá nhận:

+ Developer: Phan Thanh Liêm

+ Tester: Nguyễn Quang Phong

+ Tester: Lê Nguyên Thức

## Kế hoạch làm việc

Xác định các gói công việc bảo trì (và các công việc con), các mốc thời gian thực hiện, người chịu trách nhiệm, các dự báo về nguồn tài nguyên cần để thực hiện gói công việc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Người chịu trách nhiệm** | **Công việc bảo trì** | **Mốc thời gian thực hiện** | **Cảnh báo về các nguồn tài nguyên** |
| **Phan Thanh Liêm** | **Thêm giao diện preloading** | **14/04/2018** | * + - **Source code**     - **Tài liệu đặc tả**     - **Tài liệu thiết kế**     - **Tài liệu kiểm thử** |
| **Thêm chức năng tháo thẻ bài** | **14/04/2018** |
| **Thêm một số thẻ bài mới** | **14/04/2018** |
| **Bổ sung giao diện kết thúc trò chơi** | **14/04/2018** |
| **Tinh chỉnh, cân bài giá trị thẻ bài** | **16/04/2018** |
| **Thay đổi đồ họa cho chỉ số HP trong game** | **16/04/2018** |
| **Thay đổi phương thức chọn thẻ bài** | **17/04/2018** |
| **Nguyễn Quang Phong** | **Kiểm tra chức năng tháo bài** | **15/04/2018** |
| **Kiểm tra giao diện đồ họa** | **17/04/2018** |
| **Kiểm tra logic game** | **18/04/2018** |
| **Viết tài liệu** | **19/04/2018** |
| **Lê Nguyên Thức** | **Kiểm tra các thẻ bài** | **15/04/2018** |
| **Kiểm tra phương thức chọn thẻ bài** | **19/04/2018** |
| **Kiểm tra font và giao diện đồ họa** | **19/04/2018** |
| **Quản lý cấu hình, mã nguồn và tài liệu** | **11/04/2018** |

## Tóm tắt tài nguyên

- Nhân sự: Phan Thanh Liêm, Nguyễn Quang Phong, Lê Nguyên Thức

- Công cụ:

+ NetBeans

+ Eclipse

+ Office 365

+ Web browser: Google Chorme, FireFox, Cốc cốc

- Tài chính: Tự túc.

- Quyền truy xuất: tất cả người chơi đều có quyền ngang nhau.

- Kiểm soát tài liệu: Nhóm bảo trì và khách hàng,

## Công cụ, kỹ thuật và phương pháp

* + - Liệt kê các công cụ bảo trì được sử dụng

Công cụ quản lý phiên bản (Revision Control System - RCS): là một hệ thống lưu giữ các phiên bản của [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) của sản phẩm phần mềm, giúp các [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn) có thể dễ dàng lấy lại phiên bản mong muốn. Hệ thống này có thể được sử dụng bởi một nhóm các lập trình viên, mỗi thành viên trong nhóm thường không được phép thay đổi mã nguồn của các thành viên khác, mà chỉ có thể xem. **RCS** cho phân chia các [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) cho từng thành viên tương ứng. Nó cũng cho phép các thành viên chia sẻ một số tập tin cho nhau trong khi phát triển. Các thành viên có thể phát hiện lỗi và sửa lỗi thuận tiện trong **RCS**. Trưởng nhóm phải có nhiệm vụ cập nhật lại nội dung của các tập tin đó. **RCS** giúp cho công việc này được thực hiện một cách tự động.

* + - Mô tả các kỹ thuật và phương pháp:
* Ở đây, nhóm sử dụng Github là một trong những công cụ dùng để quản lý phiên bản tiếp cận theo hướng phân tán phổ biến nhất hiện nay. Vì những lý do dưới đây:
* Git dễ sử dụng, an toàn và nhanh chóng.
* Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch).
* Bạn có thể làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.
* Dễ dàng trong việc deployment sách.
* **Git** là tên gọi của một **Hệ thống quản lý phiên bản phân tán** (Distributed Version Control System – ***DVCS***) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (**clone**) từ một kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (**commit**) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là **Working Tree**.
* Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản,  họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.
* Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “*ảnh chụp*” (*snapshot*) trên mỗi tập tin và thư mục sau khi commit, từ đó nó có thể cho phép bạn tái sử dụng lại một ảnh chụp nào đó mà bạn có thể hiểu đó là một phiên bản. Đây cũng chính là lợi thế của Git so với các DVCS khác khi nó không “lưu cứng” dữ liệu mà sẽ lưu với dạng snapshot.
  + - Môi trường vận hành và kiểm thử:
      * Trang chủ: <https://github.com/>
      * Mô hình: phân tán
      * Ngôn ngữ viết: JAVA
      * Có 2 workflow chính: local workflow và server workflow
        + Local:

Git add: lưu tập tin đã thay đổi vào stage

* + - * Git commit: commit các tập tin đã add vào stage lên repository ở local.
        + Server:

push: push thay đổi từ repository local lên repository server

fetch: cập nhật thay đổi từ repository server về repository local

pull/rebase: sao chép source code từ server về local workspace

* + - Khi sử dụng Github trong phát triển game thẻ bài Summoner này, nhóm chúng tôi đã đạt được nhiều kết quả mong đợi:
* Các phiên bản được quản lý một cách hớp lý, chặt chẽ.
* Khi có sự thay đổi không mong muốn, có thể tìm kiếm được dữ liệu cũ đã sao lưu với phiên bản trước đó.
* Khi làm việc nhóm sẽ rất dễ xảy ra trường hợp những thành viên trong nhóm nâng cấp tính năng hiện có hoặc một vài người fix bug, nếu không dùng Git, khả năng cao là mọi người sẽ làm việc trùng lên việc của nhau, những task sắp hoàn thành sẽ bị trì hoãn. Ngoài ra tất cả mọi task lớn nhỏbắt buộc phải hoàn thành trước khi deploy, nếu chỉ một task còn dang dở thì cả chương trình sẽ bị trì hoãn.
* Ít tốn thời gian và sai sót khi sử dụng USB hoặc email,… để gửi mã nguồn công việc của từng thành viên cho nhau.

# Quản lý chất lượng

* + Xác định các chuẩn, các quy ước chi phối sự thực hiện của hoạt động bảo trì *(chuẩn của chính tổ chức bảo trì hoặc chuẩn quốc tế (VD: IEEE 1219)). Ví dụ:*
  + Các quy định lập trình: đặt tên biến/hàm, viết chú thích, v.v.
* Đặt tên biến/hàm:
* Tên biến là một chuỗi kí tự liên tiếp không có khoảng cách, tiếng việt có dấu và chứa kí tự đặc biệt.
* Tên biến phải bắt đầu bằng kí tự không phải dạng số hoặc dấu \_
* Tên biến không được đặt trùng với từ khóa
* Tên biến phân biệt hoa thường
* Tên biến đặt ngắn gọn, dễ hiểu và mô tả được ý nghĩa của việc sử dụng
* Viết chú thích
* Sử dụng comment theo 2 cách: dùng dấu ( //nội dung chú thích ) hoặc cặp dấu (/\*nội dung chú thích \*/)
* Comment với tất cả Lớp, Chức năng, Biến, quá trình xử lý điều kiện nào đó.
* Tất cả nội dung trong comment bao gồm: Biến, Function, Class, Text. Tất cả đều không có tác dụng trong phần comment và không có tác dụng, không hiển thị trong quá trình xử lý
  + Các chuẩn nhận dạng tài liệu: cách các tài liệu được nhận biết là duy nhất.
  + Các chuẩn trình bày tài liệu: sử dụng font Times New Roman, size 13.
  + Chuẩn cập nhật tài liệu: Các tài liệu được cập nhật và sử dụng phiên bản cuối cùng.
  + Các chuẩn trao đổi cho phép các tài liệu điện tử được nhận, được gửi: gửi qua mail, skype, hoặc tải lên Github.
  + Kiểm soát chất lượng
  + Một nhóm kiểm tra một phần hay toàn bộ quy trình hay hệ thống và các tư liệu của nó để tìm ra các vấn đề tiềm ẩn.
  + Mục đích của xem lại chất lượng là phát hiện ra các nhược điểm của hệ thống và các mâu thuẫn.
  + Bất cứ tài liệu nào được tạo ra trong quy trình đều có thể được xem lại.