**­­**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA** **CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

🙞🕮🙜

****

**LUẬN VĂN TỐT NGHIỆP**

**Đề tài:**

**PHÁT TRIỂN TRÒ CHƠI CHINH PHỤC**

**THÁP HÀ NỘI TRÊN NỀN TẢNG SILVERLIGHT**

**Sinh viên thực hiện:**

**MSSV: 1111325**

**Họ tên: Nguyễn Thanh Phi**

Cần Thơ, 05/2015

**LỜI CẢM ƠN**

# **CHƯƠNG 1: TỔNG QUAN**

* 1. **ĐẶT VẤN ĐỀ**

Bên cạnh sự phát triển vượt bậc của các hệ điều hành như iOS, Android trên các thiết bị di động. Hệ điều hành Windows Phone cũng ngày càng phát triển mạnh mẽ và thu hút sự quan tâm của nhiều nhà lập trình. Hiện nay, Windows Phone đã trở thành nền tảng cho những điện thoại thông minh hàng đầu với hàng trăm model trên thị trường và con số này vẫn không ngừng tăng lên với tốc độ nhanh chóng. Với sự phát triển đó, nhu cầu về các ứng dụng trên nền tảng này cũng ngày không ngừng tăng theo, tạo ra một thị trường lớn cho những nhà lập trình.

Hệ điều hành Windows Phone ra đời các đây không lâu nhưng những gì mà Windwos Phone đêm đến là thực sự đáng kể. Bên cạnh các công nghệ mới được ra đời như Frameword XNA hay Silverlight mà Microsoft xây dựng để phát triển trên nền tảng Windows Phone thì tiếp theo đó là mô hình lập trình mang lại sự hiệu quả cao như mô hinh MVVM, mô hình được xây dựng trên nền tảng Silverlight một nền tảng mới và mạnh mẽ. Nối tiếp đó là việc phát triển trên các mô hình giao tiếp truyền tin WCF để phục vụ cho các ứng dụng chơi trực tuyến. Chúng ta cũng thấy rằng sự phát triển của nền tảng Windows Phone ngày nay thực sự rất mạnh mẽ và đang thu hút nhiều sự quan tập của những nhà phát triển tương lai trong và ngoài nước.

Với những phát triển mạnh mẽ như đã trình bày, chúng ta có thể thấy rằng phát triển một ứng mang tính giải trí cao kết hợp với việc học tập tư tuy là hết sức cần thiết. Từ những điều đó, một trò chơi mang tính logic như Chinh phục tháp Hà Nội nên được phát triển. Vấn đề chủ yếu để góp phần làm cho trò chơi hấp dẫn và hứng thú là mở rộng thêm số cọc là 4 cọc và cho phép người chơi có thể chơi trực tuyến với nhau. Bên cạnh đó, cũng có sự hỗ trợ giải quyết di chuyển tự động khi người chơi có nhu cầu cần thiết.

* 1. **LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ**

Chinh phục tháp Hà Nội một trò chơi xuất hiện từ rất sớm và được đưa sang phương Tây lần đầu bởi nhà toán học người Pháp Edouard Lucas vào năm 1883. Trò chơi tháp Hà Nội ngày càng được các nhà toán học quan tâm. Với sự phát triển nhanh chóng của tin học, bài toán đã thu hút sự chú ý của của các toán học và tin học. Trò chơi tháp Hà Nội là một ví dụ điển hình về phương pháp giải đệ quy và phương pháp lập trình cơ bản.

Với sự phát triển mạnh mẽ của các thiết bị di động, trò chơi này đã được phát triển trên các nền tảng Java, C#, Android Windows Phone và Web. Tuy nhiên, trò chơi này được phát triển trên nền tảng Windows Phone rất ít và các trò chơi còn đơn giản chưa hỗ trợ nhiều trợ giúp cho người chơi.

Phát triển trò chơi Chinh phục tháp Hà nội trên nền tảng Silverlight –Windows Phone sẽ mang lại một ứng dụng mới cho hệ điều hành di động. Đặc biệt, hệ điều hình Windows Phone là một nền tảng mới với sự phát triển tiềm năng trong tương lai.

Trò chơi Chinh phục tháp Hà Nội có từ lâu, trài qua nhiều phiên bản phát triển nhưng hiện tại vẫn còn có nhiều điểm hạn chế như: hướng dẫn luật chơi còn sơ sài, giao diện chưa thực sự linh động, không có hỗ trợ khi người chơi có nhu cầu hay cho phép chơi trực tuyếns. Đặc biệt, tất cả các phiên bản hiện tại chưa phát triển được cho phép người chơi chơi với 4 cọc. Mặc dù có có nhiều ý tưởng để giải quyết bài toán tháp Hà Nội tổng quát.

Vì thế, khi đề tài được thực hiện thành công, trò chơi Chinh phục tháp Hà Nội sẽ trở thành một trò chơi hoàn hiện khắc phục được những khuyết điểm nêu trên và sẽ được phát triển trên nền tảng Silverlight -Windows Phone. Đặc biệt, ứng dụng này khi hoàn thành sẽ bổ sung kiến thức lập trình trên nền tảng Silverlight và góp phần làm phong phú hơn cho trò chơi Việt Nam.

* 1. **PHẠM VI VẤN ĐỀ**
     1. **VỀ NGƯỜI DÙNG**

Trò chơi chinh phục tháp Hà nội cung cấp cho người dùng những tính năng cho phép người chơi lựa chọn 2 mức độ chơi là chơi đơn và thách đấu. Phần này chỉ mô tả những chức năng cơ bản trong ứng dụng bao gồm:

**Thách đấu trực tuyến:**

+ Đăng kí

+ Đăng nhập

+ Danh sách phòng chơi

+ Đăng kí phòng chơi

+ Bảng xếp hạng trực tuyến

**Chơi đơn**

+ Chọn số cọc để chơi bao gồm 3 cọc và 4 cọc

+ Thành tích cá nhân của 3 cọc và 4 cọc

+ Trợ giúp

+ Làm mới phần chơi

+ Làm mới thành tích cá nhân

* + 1. **VỀ LÝ THUYẾT**

Để có thể phát triển trò chơi Chinh phục tháp Hà nội một cách hoàn thiện và hiểu quả trên nền tảng Windows Phone, ta cần tìm hiểu và sử dụng các nền kiến thức chủ yếu sau đây:

* Tìm hiểu thuật toán đệ quy cho bài toán tháp Hà nội đặc biệt là Thuật toán Frame- Stewart giải bài toán tháp Hà Nội tổng quát.
* Công nghệ Silverlight cho phát triển ứng dụng Windows Phone, ngôn ngữ đánh dấu mở rộng XAML phát triển trên ứng dụng.
* Mô hình MVVM trong việc quản lý và phát triển ứng dụng trên nền tảng Silverlight.
* Công nghệ WCF phục vụ cho đồng bộ thông tin giữa các người chơi
* Ngôn ngữ truy vấn tích hợp ngôn ngữ lập trình LINQ để phục vụ cho quá trình truy xuất dữ liệu trực tuyến.
  + 1. **VỀ KĨ THUẬT**

Chinh phục tháp Hà nội cần đạt được các yêu cầu kĩ thuật:

* Chạy ổn định trên hệ điều hành Windows Phone 8.0 trở lên
* Chạy ổn định trên thiết bị giả lập Windows Phone Emulator
* Trong phần kết nối trực tuyến với máy chủ, thời gian phản hồi giữa các thao tác tối đa 5 giây.
  1. **PHƯƠNG PHÁP NGHIÊN CỨU**
     1. **THU THẬP DỮ LIỆU**

Thu thập thông tin về trò chơi Chinh phục tháp Hà nội như cách chơi, điểm số cho mỗi mức của trò chơi.

Tìm hiểu thuật giải cho bài toán tháp Hà nội để ứng dụng vào phần “Trợ giúp” cho người chơi khi cần thiết.

Tìm hiểu về quy trình phát triển ứng dụng trên thiết bị di động cụ thể là hệ điều hành Windows Phone 8.0. Song song đó tìm hiểu lập trình trên hệ điều hành mới này với nền tảng là Silverlight

Chọn lọc và tổng hợp lại các tài liệu về công nghệ Silverlight, XAML để phục vụ cho thiết kế giao diện ứng dụng

Tìm hiểu về WCF để hỗ trợ chơi trực tuyến

Chọn một số game đã được phát triển trước đó trên các nền tảng Java, Android, Web. Thu thập nhưng ưu, khuyết điểm của từng loại để đưa ra những cải tiết phù hợp trên nền tảng Windows Phone.

* + 1. **PHÂN TÍCH VÀ XÂY DỤNG MÔ HÌNH HỆ THỐNG**

Phân tích yêu cầu của người chơi

Thiết kế mô hình mức quan niệm và mức vật lý cho hệ thống bằng công cụ Power Designer hỗ trợ phần chơi thách đấu trực tuyến

* + 1. **CHỌN CÔNG NGHỆ THỰC HIỆN**

Sử dụng công nghệ Silverlight để phát triển ứng dụng

Sử dụng ngôn ngữ lập trình C# để viết mã nguồn (code-behind) của ứng dụng

Sử dụng LINQ để thao tác với cơ sở dữ liệu ở phần chơi đơn

Ngôn ngữ XAML để thiết kế giao diện ứng dụng trên nền tảng Windows Phone 8.0

Sử dụng SQL Server để thao tác với cơ sở dữ liệu tại phần thách đấu trức tuyến.

Sử dụng WCF để giao tiếp truyền tin giữa các máy

Mô hình MVVM để tổ chức và quản lý mã nguồn

* + 1. **CHỌN CÔNG CỤ LẬP TRÌNH**

Visual Studio 2012

SQL Server 2012

Microsoft Expression Blend 4

Power Designer 15.2

Windows Phone SDK 8.0 có tích hợp bộ giả lập Emulator 8.0 để tạo và thực thi ứng dụng cho Windows Phone 8.0

Kế hoạch thực hiện luận văn

|  |  |  |
| --- | --- | --- |
| **Tuần** | **Thời điểm** | **Nội dung thực hiện** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## 2.1 CĂN BẢN NGÔN NGỮ C#

**2.1.1 TẠI SAO PHẢI SỬ DỤNG NGÔN NGỮ C#**

Nhiều người tin rằng không cần thiết có một ngôn ngữ lập trình mới. Java, C++, Perl, Microsoft Visual Basic, và những ngôn ngữ khác được nghĩ rằng đã cung cấp tất cả những chức năng cần thiết. Ngôn ngữ C# là một ngôn ngữ được dẫn xuất từ C và C++, nhưng nó được tạo từ nền tảng phát triển hơn. Microsoft bắt đầu với công việc trong C và C++ và thêm vào những đặc tính mới để làm cho ngôn ngữ này dễ sử dụng hơn. Nhiều trong số những đặc tính này khá giống với những đặc tính có trong ngôn ngữ Java. Không dừng lại ở đó, Microsoft đưa ra một số mục đích khi xây dựng ngôn ngữ này.

Những mục đích này được tóm tắt như sau:

C# là ngôn ngữ đơn giản

C# là ngôn ngữ hiện đại

C# là ngôn ngữ hướng đối tượng

C# là ngôn ngữ mạnh mẽ và mềm dẻo

C# là ngôn ngữ có ít từ khóa

C# là ngôn ngữ phổ biến

C# là ngôn ngữ hướng module

**2.1.2 C# LÀ NGÔN NGỮ ĐƠN GIẢN**

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và C++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class). Chúng là những nguyên nhân gây ra sự nhầm lẫn hay dẫn đến những vấn đề cho người phát triển C++. Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi. Ví dụ như, trong C++ có ba toán tử làm việc với các thành viên là::, . , và ->. Để biết khi nào dùng ba toán tử này cũng phức tạp và dễ nhầm lẫn. Trong C#, chúng được thay thế với một toán tử duy nhất gọi là. (dot). Đối với người mới học thì điều này và những việc cải tiến khác làm bớt nhầm lẫn và đơn giản hơn.

**2.1.3 C# LÀ NGÔN NGỮ HIỆN ĐẠI**

Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Trong C#, bộ thu gom bộ nhớ tự động và kiểu dữ liệu an toàn được tích hợp vào ngôn ngữ, sẽ loại bỏ những vấn đề rắc rối của C++.

**2.1.3 C# LÀ NGÔN NGỮ HƯỚNG ĐỐI TƯỢNG**

C# hỗ trợ tất cả những đặc tính chính của ngôn ngữ hướng đối tượng (Object-oriented language) là sự đóng gói (encapsulation), sự kế thừa (inheritance), và đa hình (polymorphism).

**2.1.4 C# LÀ NGÔN NGỮ MẠNH MẼ VÀ MỀM DẼO**

Ngôn ngữ này không đặt những ràng buộc lên những việc có thể làm. C# được sử dụng cho nhiều các dự án khác nhau như là tạo ra ứng dụng xử lý văn bản, ứng dụng đồ họa, bản tính, hay thậm chí những trình biên dịch cho các ngôn ngữ khác.

**2.1.5 C# LÀ NGÔN NGỮ ÍT TỪ KHÓA**

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

**2.1.6 C# LÀ NGÔN NGỮ PHỔ BIẾN**

Microsoft muốn ngôn ngữ C# trở nên phổ biến. Mặc dù một công ty không thể làm một sản phẩm trở nên phổ biến, nhưng nó có thể hỗ trợ. Cách đây không lâu, Microsoft đã gặp sựthất bại về OS Microsoft Bob. Mặc dù Microsoft muốn Bob trở nên phổ biến nhưng thất bại. C# thay thế tốt hơn để đem đến thành công so với Bob. Thật sự là không biết khi nào mọi người trong công ty Microsoft sử dụng Bob trong công việc hằng ngày của họ. Tuy nhiên, với C# thì khác, nó được sử dụng bởi Microsoft. Nhiều sản phẩm của công ty này đã chuyển đổi và viết lại bằng C#.

Bằng cách sử dụng ngôn ngữ này Microsoft đã xác nhận khả năng của C# cần thiết cho những người lập trình. Micorosoft .NET là một lý do khác để đem đến sự thành công của C#. .NET là một sựthay đổi trong cách tạo và thực thi những ứng dụng.

Ngoài hai lý do trên ngôn ngữ C# cũng sẽ trở nên phổ biến do những đặc tính của ngôn ngữ này được đề cập trong mục trước như: đơn giản, hướng đối tượng, mạnh mẽ...

**2.1.7 C# LÀ NGÔN NGỮ HƯỚNG MODULE**

Mã nguồn C# có thể được viết trong những phần được gọi là những lớp, những lớp này chứa các phương thức thành viên của nó. Những lớp và những phương thức có thể được sử dụng lại trong ứng dụng hay các chương trình khác. Bằng cách truyền các mẫu thông tin đến những lớp hay phương thức chúng ta có thể tạo ra những mã nguồn dùng lại có hiệu quả.

**2.1.8 SO SÁNH C# VỚI CÁC NGÔN NGỮ KHÁC:**

Chúng ta đã từng nghe đến những ngôn ngữ khác như Visual Basic, C++ và Java. Có lẽ chúng ta cũng tự hỏi sự khác nhau giữa ngôn ngữ C# và nhưng ngôn ngữ đó. Và cũng tự hỏi tại sao lại chọn ngôn ngữ này mà không chọn một trong những ngôn ngữ kia. Có rất nhiều lý do và chúng ta hãy xem một số sự so sánh giữa ngôn ngữ C# với những ngôn ngữ khác giúp chúng ta phần nào trả lời được những thắc mắc.

Microsoft nói rằng C# mang đến sức mạnh của ngôn ngữ C++ với sự dễ dàng của ngôn ngữ Visual Basic. Có thể nó không dễ như Visual Basic, nhưng với phiên bản Visual Basic.NET (Version 7) thì ngang nhau. Bởi vì chúng được viết lại từ một nền tảng. Chúng ta có thể viết nhiều chương trình với ít mã nguồn hơn nếu dùng C#.

Mặc dù C# loại bỏ một vài các đặc tính của C++, nhưng bù lại nó tránh được những lỗi mà thường gặp trong ngôn ngữ C++. Điều này có thể tiết kiệm được hàng giờ hay thậm chí hàng ngày trong việc hoàn tất một chương trình. Một điều quan trọng khác với C++ là mã nguồn C# không đòi hỏi phải có tập tin header. Tất cả mã nguồn đều được viết trong khai báo một lớp.

Như đã nói ở bên trên. .NET runtime trong C# thực hiện việc thu gom bộ nhớ tự động. Do điều này nên việc sử dụng con trỏ trong C# ít quan trọng hơn trong C++. Tuy nhiên cũng có thể sử dụng con trỏ trong C#, khi đó những đoạn mã nguồn này sẽ được đánh dấu là không an toàn (unsafe code).

C# cũng bỏ ý tưởng đa kế thừa như trong C++. Và sự khác nhau nữa là C# đưa thêm thuộc tính vào trong lớp giống như trong Visual Basic. Và những thành viên của lớp được gọi duy nhất bằng toán tử “.” khác với C++ có nhiều cách gọi trong các tình huống khác nhau.Một ngôn ngữ khác rất mạnh và phổ biến là Java, giống như C++ và C# Java cũng được phát triển dựa trên C. Điểm giống nhau C# và Java là cả hai cùng biên dịch ra mã trung gian: C# biên dịch ra MSIL còn Java biên dịch ra bytecode. Sau đó chúng được thực hiện bằng cách thông dịch hoặc biên dịch just-in-time trong từng máy ảo tương ứng. Tuy nhiên, trong ngôn ngữ C# nhiều hỗ trợ được đưa ra để biên dịch mã ngôn ngữ trung gian sang mã máy. C# chứa nhiều kiểu dữ liệu cơ bản hơn Java và cũng cho phép nhiều sự mở rộng với kiểu dữ liệu giá trị. Tương tự như Java, C# cũng từ bỏ tính đa kế thừa trong một lớp, tuy nhiên mô hình kế thừa đơn này được mở rộng bởi tính đa kế thừa nhiều giao diện

## 2.2 SILVERLIGHT

### 2.2.1 Silverlight là gì?

Silverlight (SL) là một dạng plug-in dựa trên công nghệ của Microsoft.NET, nó độc lập với đa nền tảng và đa trình duyệt, nó cho phép phát triển các ứng dụng đa phương tiện đặc biệt là các ứng dụng trên web và Windows Phone. SL cung cấp một mô hình lập trình mềm dẻo và đồng nhất, nó hỗ trợ Ajax, Python, Ruby và các ngôn ngữ lập trình như Visual Basic, C#.

### Các đặc tính của Siverlight

#### 2.2.2.1 Sự kết hợp của WPF và XAML

SL là một gói nhỏ của công nghệ Windows Presentation Foundation (WPF). Nó được mở rộng nhiều hơn các Element trong trình duyệt, ứng dụng Windows Phone để tạo giao diện người dùng. WPF cho phép ta tạo ra đồ họa 3D, hình ảnh động, đa phương tiện và nhiều tính năng phong phú khác trên máy khách. XAML (Extensible Application Markup Language) cung cấp các cú pháp đánh dấu đặc trưng cho việc tạo các Element.

#### Mở rộng cho ngôn ngữ kịch bản

SL cung cấp việc mở rộng cho các ngôn ngữ kịch bản (Javascript) ở một số trình duyệt phổ biến để thể hiện việc trình bày giao diện và thao tác người dùng một cách phong phú hơn.

#### Sự tích hợp với các ứng dụng đã có

SL tích hợp liền mạch với ngôn ngữ Javascript và mã Ajax của ASP.NET để bổ sung các chức năng ta đã xây dựng được. Ta có thể tạo ra những tài nguyên trên nền máy chủ có trong ASP.NET và sử dụng các khả năng của Ajax để tương tác với tài nguyên trên nền máy chủ đó mà không làm gián đoạn người dùng.

#### Mô hình ngôn ngữ lập trình trên nền tảng .NET

Ta có thể tạo các ứng dụng trên nền tảng SL và sử dụng các ngôn ngữ động như InronPython, C#, Visual Basic. Ta cũng có thể sử dụng các công cụ phát triển như Visual Studio để tạo ứng dụng trên nền tảng SL.

#### Hỗ trợ mạng

SL bao gồm các hỗ trợ cho HTTP qua TCP. Ta có thể kết nối đến các dịch vụ của WCF, SOAP hoặc ASP.NET AJAX và nhận về các định dạng theo cấu trúc XML, JSON hay dữ liệu RSS.

#### Hỗ trợ ngôn ngữ tích hợp truy vấn (LINQ to SQL)

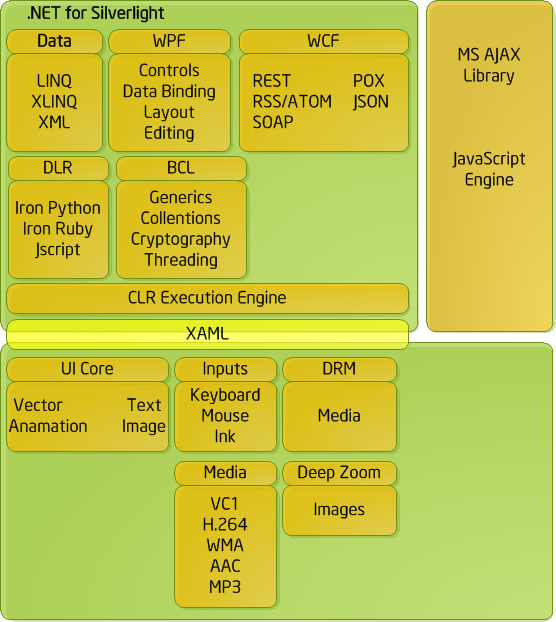
Điều này cho phép ta truy cập dữ liệu bằng cách sử dụng cú pháp trực quan tự nhiên và mạng mẽ, được gõ bởi các đối tượng có trong các ngôn ngữ .Net Framwork.

* + 1. **Kiến trúc tổng thể và các mô hình lập trình của SL**
       1. **Kiến trúc và các thành phần**

Về cơ bản SL là một nền tảng thống nhất của nhiều thành phần khác nhau. Tuy nhiên ta có thể nhóm lại các thành phần chính của SL vào bảng sau đây:

|  |  |
| --- | --- |
| Thành phần | Diễn tả |
| Nền tảng trình bày cơ sở | Các thành phần và dịch vụ hướng tới giao diện người dùng và tương tác người dùng, bao gồm các control cho dữ liệu mà người dùng nhập, thiết bị đa phương tiện, quản lý phân quyền số, trình bày dữ liệu, đồ họa vector, chữ, hình ảnh động, cũng bao gồm XAML để đặc tả việc bố trí giao diện. |
| .Net Framework cho SL | Là một gói nhỏ trong .Net Framework, bao gồm các thành phần và các thư viện, kể cả việc tương tác dữ liệu, khả năng mở rộng các control, mạng, garbage collection và CRL. |
| Cài đặt và cập nhật | Là thành phần để xử lý các tiến trình cài đặt làm sao để đơn giản hóa cho lần cài đặt đầu tiên, tiếp sau đó chỉ cung cấp cơ chế tự động cập nhật và tương tác ở mức thấp. |

Dưới đây là hình ảnh mô tả những thành phần trong kiến trúc của SL cùng với các thành phần và dịch vụ liên quan khác.



*Hình . Thành phần kiến trúc của Silverlight (nguồn* [*MSDN*](http://msdn.microsoft.com/en-us/library/bb404713(v=vs.95).aspx)*)*

**Core Presentation Components**

|  |  |
| --- | --- |
| Tính năng | Mô tả |
| Dữ liệu vào (Input) | Xử lý dữ liệu vào từ các thiết bị phần cứng như bàn phím, chuột, bảng vẽ hoặc các thiết bị đầu vào khác. |
| Trình bày giao diện người dùng (UI Rendering) | Trình bày các vector và các đồ họa ảnh bitmap, ảnh động và văn bản |
| Thiết bị nghe nhìn (Media) | Các tính năng phát và quản lý một vài thể loại file âm thanh và hình ảnh như .WMA, .MP3 |
| Controls | Hỗ trợ mở rộng cho các control để có khả năng tùy chỉnh về kiểu dáng và khuôn mẫu |
| Xếp đặt Layout | Cho phép khả năng xếp đặt vị trí động các thành phần giao diện người dùng |
| Trình bày dữ liệu (Data Binding) | Cho phép việc kết nối dữ liệu của các đối tượng và các thành phần giao diện người dùng |
| DRM | Khả năng quản lý phân quyền số |
| XAML | Cung cấp trình phân tách cho XAML |

**.Net Framwork for Silverlight**

|  |  |
| --- | --- |
| Tính năng | Mô tả |
| Data | Hỗ trợ ngôn ngữ truy vấn tích hợp (LINQ to SQL) và LINQ với đặc tả XML, dễ dàng xử lý việc tích hợp và làm việc với dữ liệu từ nhiều nguồn khác nhau. Hỗ trợ việc sử dụng XML và các lớp biến đổi hóa (serialzation) để xử lí dữ liệu |
| Base class library | Thuộc thư viện của .Net Framwork, nó cung cấp các chức năng lập trình chủ yếu như việc xử lí chuỗi, biểu thức chính quy, đầu ra và đầu vào, ánh xạ, tập hợp và toàn cục hóa. |
| WCF | Cung cấp các tính năng để đơn giản hóa việc truy cập dữ liệu từ xa. Cơ chế này bao gồm một đối tượng trình duyệt, HTTP request và HTTP response, RSS, JSON, POX và các SOAP |
| Common Language Runtime (CLR) | Cung cấp việc quản lí bộ nhớ, dọn dẹp bộ nhớ thừa, xử lí ngoại lệ… |
| WPF | Cung cấp các control giàu tính năng như Button, Listbos, Textbox, PasswordBox, Grid, ScrollViewer,… |
| Dynamic Language Runtime (DLR) | Hỗ trợ việc biên dịch và thi hành với tính năng động của các ngôn ngữ kịch bản như Javascript và IronPython cho các chương trình trên nền tảng SL. |

* + - 1. **Các công cụ và công nghệ liên qua của Silverlight**

**Microsoft Expression Blend:** sử dụng công cụ này ta có thể tạo và thay đổi cách sắp xếp trình bày các lớp ứng dụng bằng cách thao tác đến các Grid và Control trong XAML, làm việc với các chức năng đồ họa, lập trình với ngôn ngữ Javascript.

**Visual Stuido 2010:** Visual Studio cung cấp các công cụ hiệu quả cho việc phát triển các ứng dụng có hỗ trợ thao tác code-behind. Tất cả các phiên bản đã có của Visual Studio đều có khả năng hỗ trợ SL. Tuy nhiên ở phiên bản mới này nó còn hỗ trợ các tính năng đặc biệt hơn như bao gồm khả năng IntelliSense, Debugging và các template cho việc tạo mới một ứng dụng SL.

**Internet Servers**: Bao gồm IIS (Microsoft Internet Information Services) và Apache Web server

**Microsoft Windows Communication Foundation (WCF)** services

Ngoài ra còn có:

**ASP.NET AJAX**: Bao gồm tập các control, service và các thư viện cần thiết cho việc tạo và tương tác với nền ứng dụng.

**Microsof ASP.NET 3.5 Extensions Preview**: Công nghệ này cung cấp chức năng thêm để việc tăng cường các ứng dụng ASP.NET AJAX. Nó bao gồm 2 control sử dụng hữu ích cho việc xây dựng úng dụng nền tảng SL cũng như là một phần của ứng dụng: ASP.NET Siverlight Server Control

## XAML

### 2.3.1 Sơ lược về XAML

XAML (Extensible Apllication Markup Lanuage) là một ngôn ngữ dạng khai báo. Ta có thể tạo ra các phần tử đồ họa (UI) với những khai báo thông qua thẻ trong XAML. Sau đó ta có thể dùng file mã lệnh tách biệt của nó (code-behind) để trả về những sự kiện và điều khiển những đối tượng mà ta đã định nghĩa trong XAML. Nó là một ngôn ngữ mô tả dựa trên XML là rất trực quan trong việc xây dựng giao diện từ những bước phác thảo cho đến sản xuất sản phẩm.

### Khai báo đối tượng

Có 2 cách để khai báo đối tượng trong XAML:

Khai báo trực tiếp: sử dụng thẻ đóng mở để khai báo một đối tượng giống như là phần tử XML. Ta cũng có thể sử dụng cú pháp này để khai báo đối tượng gốc (root object) hoặc để xét các giá trị của các thuộc tính.

Khai báo gián tiếp: sử dụng giá trị trực tiếp để khai báo một đối tượng. Ta có thể sử dụng cú pháp này để thiết lập các giá trị của các thuộc tính. Thông thường, điều này có nghĩa là chỉ những thuộc tính được hỗ trợ bởi một đối tượng nào đó mới có thể sử dụng được phương pháp này.

### Thiết lập đặc tính cho đối tượng

Có những cách sau để khai báo đặc tính cho đối tượng:

Sử dụng cú pháp theo thuộc tính: Dưới đây là ví dụ xét giá trị cho các thuộc tính: Width, Height, Fill của đối tượng Rectangle

<Rectangle Width="100" Height="100" Fill="Blue" />

* Sử dụng cú pháp theo đặc tính của thành phần (Element): Dưới đây là ví dụ xét đặc tính Fill theo cách này cho đối tượng Rectangle

<Rectangle Width="100" Height="100">

<Rectangle.Fill>

<SolidColorBrush Color="Blue" />

</Rectangle.Fill>

</Rectangle>

Sử dụng cú pháp theo nội dung: Dưới đây là ví dụ xét đặc tính Text cho đối tượng TextBlock (giống với đối tượng Label trong Winform, Webform)

<TextBlock>

Hello!

</TextBlock>

Sử dụng theo một tập hợp: đây là một trường hợp khá thú vị trong XAML, bởi có những cách khác nhau để thể hiện tập hợp này. Hơn nữa các khai báo này có thể xuất hiện ở phần đầu tiên của XAML cho phép ta xét những đặc tính chỉ đọc (read-only) của đối tượng. Dưới đây là ví dụ xét đặc tính theo những cách khác nhau sử dụng theo kiểu tập hợp.

**Cách 1:**

<LinearGradientBrush>

<LinearGradientBrush.GradientStops>

<!--Ở đây thẻ GradientStopCollention được chỉ rõ-->

<GradientStopCollection>

<GradientStop Offset="0.0" Color="Red" />

<GradientStop Offset="1.0" Color="Blue" />

</GradientStopCollection>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

**Cách 2:**

<LinearGradientBrush>

<LinearGradientBrush.GradientStops>

<!--Không cần khai báo GradientStopCollention, bộ phân tích sẽ hiểu và tạo ra nó-->

<GradientStop Offset="0.0" Color="Red" />

<GradientStop Offset="1.0" Color="Blue" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

Cách 3: Ngoài ra, có những đặc tính mà chúng có những tập hợp đặc tính nhưng chúng được xác định như là đặc tính nội dung của lớp. Trong trường hợp này ta xét đến đặc tính GradientStops được xử dụng ở trên. Ta có thể loại bỏ đặc tính này và sẽ có kết quả như sau:

<LinearGradientBrush>

<GradientStop Offset="0.0" Color="Red" />

<GradientStop Offset="1.0" Color="Blue" />

</LinearGradientBrush>

### Root elements và Namespace trong XAML

Một file XAML chỉ được có duy nhất một Root element và phải thỏa mãn cả hai tiêu chí sau: wellformed XML (có thẻ mở và đóng) và valid XML (tuân thủ Document Type Definition (DTD)). Ví dụ dưới đây cho Root element điển hình của XAML cho SL với Root element là thành phần PhoneApplicationPage

<phone:PhoneApplicationPage

x:Class="PanoramaApp1.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"

xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"

xmlns:controls="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="800"

d:DataContext="{d:DesignData SampleData/MainViewModelSampleData.xaml}"

FontFamily="{StaticResource PhoneFontFamilyNormal}"

FontSize="{StaticResource PhoneFontSizeNormal}"

Foreground="{StaticResource PhoneForegroundBrush}"

SupportedOrientations="Portrait" Orientation="Portrait"

shell:SystemTray.IsVisible="False">

</phone:PhoneApplicationPage>

### Sự kiện trong XAML

XAML là một ngôn ngữ khai báo cho đối tượng và những đặc tính của chúng, nhưng nó cũng bao gồm những cú pháp cho sự kết hợp các sự kiện của đối tượng trong những thẻ đánh dấu. Ta chỉ rõ tên của sự kiện như là một thuộc tính tên của đối tượng mà sự kiện được thực hiện. Về giá trị của thuộc tính, ta chỉ cần chỉ rõ tên của hàm nghe sự kiện mà ta định nghĩa ở phần code-behind.

Việc có khai báo hay không đặc tính x:Class ở trong thẻ Root của XAML có ảnh hưởng đến việc xử lí sự kiện. Nếu ta khai báo x:Class tức là việc xử lí sự kiện sẽ được thực hiện trong code-behind.

Ví dụ dưới đây chỉ rõ cho ta thấy cách tạo một sự kiện trong XAML và thực thi sự kiện trong code-behind:

<phone:PhoneApplicationPage

x:Class="PanoramaApp1.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"

xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"

xmlns:controls="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="800"

d:DataContext="{d:DesignData SampleData/MainViewModelSampleData.xaml}"

FontFamily="{StaticResource PhoneFontFamilyNormal}"

FontSize="{StaticResource PhoneFontSizeNormal}"

Foreground="{StaticResource PhoneForegroundBrush}"

SupportedOrientations="Portrait" Orientation="Portrait"

shell:SystemTray.IsVisible="False">

<Grid x:Name="LayoutRoot" Background="White">

<Button Click="Button\_Click" Content="Click me"></Button>

</Grid>

</phone:PhoneApplicationPage>

Đoạn code trên, trong thẻ root <phone:PhoneApplicationPage> chúng ta đã khai báo đặc tính x:Class=” ThapHaNoi\_NguyenThanhPhi.MainPage”, điều này có nghĩa trong chương trình của chúng ta có một file chứa class ThapHaNoi\_NguyenThanhPhi.MainPage. Class này sẽ đảm nhiệm việc xử lý các sự kiện đã khai báo ở file XAML. Trong thẻ <Grid> chúng ta tạo thêm một thẻ <Button> và khai báo trong thẻ đó một sự kiện Click=”Button\_Click”. Sự kiện này được xử lí trong code-behind như sau:

namespace ThapHaNoi\_NguyenThanhPhi

{

public partial class MainPage : PhoneApplicationPage

{

// Constructor

public MainPage()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

//Xử lí sự kiện Button\_Click tại đây

MessageBox.Show("Tower of Hanoi", “Message”);

}

}

}

## Windows Communication Foundation (WCF)

### Sơ lược về WCF

WCF là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất. Vào 11.2005, .NET 2.0 được MS phát hành trong đó cung cấp các hàm API riêng biệt cho các liên lạc dựa trên SOAP để tối đa hóa sự làm việc giữa các nền tảng sử dụng WebServices, đồng thời .NET 2.0 còn cung cấp các API để tối ưu việc liên lạc dựa trên mã nhị phân giữa các ứng dụng chạy trên hệ thống Windows gọi là .NET Remoting, các API cho các giao dịch phân tán, và API cho liên lạc dị bộ. WCF thống nhất các API này thành một mô hình duy nhất nhằm đáp ứng mô hinh lập trình hướng dịch vụ.

WCF có thể sử dụng các bản tin SOAP giữa hai tiến trình, do đó làm cho các ứng dụng dựa trên WCF có thể làm việc với các tiến trình khác thông qua việc giao tiếp sử dụng bản tin SOAP. Khi một tiến trình WCF liên lạc với một tiến trình không là WCF, các bản tin SOAP được mã hóa trên cơ sở XML, nhưng khi nó liên lạc với một tiến trình WCF khác, bản tin SOAP có thể được tối ưu dựa trên mã hóa nhị phân.

### Tại sao sử dụng WCF

Như phần trên đã trình bày, .NET 2.0 hỗ trợ rất nhiều phương pháp liên lạc giữa các ứng dụng khác nhau nhằm vào các mục tiêu khác nhau. Các phương pháp liên lạc này khá phức tạp và mất nhiều thời gian để làm chủ được công nghệ. Tuy nhiên kiến thức thu được từ việc triển khai một phương pháp ít có khả năng dùng được khi làm việc với phương pháp khác.

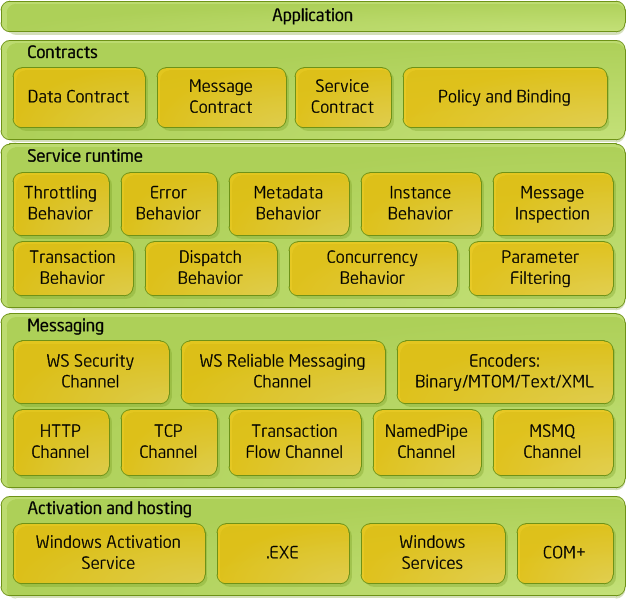
Với việc ra đời của WCF, mọi phương pháp liên lạc trước kia đều có thể thực hiện trên WCF. Do vậy nhà phát triển chỉ cần làm chủ công nghệ WCF là có thể xây dựng các ứng dụng một cách nhanh chóng.

WCF là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định và bảo mật, thậm chí là đảm bảo giao dịch. Nó làm đơn giản hóa việc phát triển các ứng dụng kết nối và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản và quản lí được.

Do WCF được xây dựng trên cơ sở của .NET 2.0 CLR, nó là tập hợp các lớp cho phép nhà phát triển xây dựng các ứng dụng hướng dịch vụ bằng môi trường lập trình quen thuộc của họ như C# hay VB.NET.

### Kiến trúc của WCF

Hình sau mô tả các lớp chủ yếu trong kiến trúc của WCF.



Hình 2. Kiến trúc của WCF (nguồn [MSDN](http://www.google.com/imgres?hl=vi&client=aff-maxthon-maxthon4&sa=X&channel=t1&biw=1368&bih=610&tbm=isch&tbnid=1GiMREw5rBK5BM:&imgrefurl=http://msdn.microsoft.com/en-us/library/ms733128.aspx&docid=aglT2E1_7xWauM&imgurl=http://i.msdn.microsoft.com/dynimg/IC5864.gif&w=322&h=451&ei=5VplUfeVFqf_iAf-vIDACw&zoom=1&ved=1t:3588,r:0,s:0,i:82&iact=rc&dur=303&page=1&tbnh=179&tbnw=128&start=0&ndsp=19&tx=93&ty=85))

#### Các Contracts

Các contract trong WCF cũng giống như các hợp đồng/hiệp nghị mà ta kí trong đời sống thật. Một hợp đồng ta kí có thể chứa các thông tin như kiểu công việc ta sẽ làm gì và những thông tin mà ta muốn đưa ra cho các bên khác. WCF contract cũng chứa những thông tin tương tự như vậy. Contract định nghĩa các đặc tả trong hệ thống bản tin. Thông thường có các loại contract sau:

* Contract dữ liệu mô tả các tham số cho các bản tin mà một dịch vụ có thể tạo ra hay sử dụng. Các tham số bản tin được định nghĩa bằng các tài liệu sử dụng ngôn ngữ đặc tả XML Schema (XSD), điều này cho phép các hệ thống hiểu XML có thể xử lý tài liệu dễ dàng. Các dịch vụ khi liên lạc với nhau có thể không cần đồng ý với nhau về các kiểu, nhưng cần đồng ý về contract dữ liệu, nghĩa là đồng ý về các tham số và các kiểu trả về.
* Contract bản tin định nghĩa các phần có trong bản tin sử dụng các giao thức SOAP và nó cho phép điều khiển sâu hơn tới các phần trong bản tin khi có yêu cầu sự chính xác trong giao tiếp.
* Contract dịch vụ đặc tả chi tiết các phương thức của dịch vụ, và được thực thi như một giao diện trong ngôn ngữ lập trình C#, VB.NET.
* Bindings – các chính sách và các kết nối mô tả các điều kiện cần có để giao tiếp với một dịch vụ. Các chính sách sẽ bao gồm các yêu cầu về bảo mật và các điều kiện khác cần phải có khi kết nối với một dịch vụ.

#### Runtime service

Lớp dịch vụ thực thi chứa các hành xử sẽ xảy ra trong quá trình thực thi một dịch vụ. Có các hành xử sau:

* Throttling behavior: điều khiển luồng nhằm quy định xem có bao nhiêu bản tin được xử lý.
* Error behavior: hành xử lỗi quy định những hành động khi lỗi xảy ra trong hệ thống
* Metadata behavior: hành xử với các siêu dữ liệu quy định xem làm thế nào và khi nào thì các siêu dữ liệu được đưa ra bên ngoài dịch vụ.
* Instance behavior: hành xử thực thể quy định xem có bao nhiêu thực thể của dịch vụ đó được chạy.
* Transaction behavior: hành xử giao dịch cho phép tạo ra rollback các giao dịch nếu xảy ra lỗi
* Message inspection: kiểm tra bản tin đem lại cho dịch vụ khả năng kiểm tra tất cả hay một số phần của bản tin
* Dispatch behavior: khi một bản tin được xử lý bởi nền tảng WCF, dịch vụ Dispatch behavior xác định xem bản tin được xử lí thế nào.
* Concurrency behavior: hành xử đồng thời xác định xem việc xử lí thế nào với việc đa luồn của mỗi dịch vụ hay mỗi thực thể của dịch vụ. Hành xử này giúp cho việc điều khiển số lượng luồng có thể truy cập đến một thực thể của dịch vụ.
* Parameter filtering: khi một bản tin được đưa đến một dịch vụ, sẽ xảy ra một số hành động dựa trên nội dung phần đầu của bản tin. Phần lọc tham số sẽ thực hiện lọc các đầu đề bản tin và thực hiện các hành động đặt sẵn dựa trên việc lọc đầu đề bản tin.

#### message

Lớp bản tin là tập hợp các kênh. Mỗi kênh là một thành phần xử lí bản tin theo một cách nào đó. Mỗi tập các kênh thường được gọi là ngăn xếp kênh. Các kênh làm việc trên bản tin và trên đầu đề của bản tin. Lớp này khác với lớp thực thi dịch vụ trong việc xử lí nội dung bản tin.

Có hai kênh khác nhau là kênh vận chuyển (transport channel) và kênh điều khiển (control channel)

* Kênh vận chuyển phụ trách việc đọc và ghi các bản tin từ mạng (network) hoặc từ một số điềm giao dịch bên ngoài
* Kênh điều khiển thực hiện xử lí bản tin theo giao thức, thông thường làm việc bằng cách đọc và ghi thêm các đầu đề cho bản tin.

#### Host and activation

Nhìn một cách tổng thể thì một dịch vụ thực chất là một chương trình. Cũng giống như các chương trình khác, một dịch vụ cần phải chạy trong một file thực thi. Dịch vụ này thường được gọi là dịch vụ tự chứa.

Các dịch vụ còn có thể được chứa, hoặc chạy trong một file thực thi được quản lí bởi một agent bên ngoài như IIS hay Windows Activation Services (WAS). WAS cho phép WCF được kích hoạt một cách tự động khi phân phối đến một máy tính có chạy WAS.

#### Mô hình lập trình WCF

##### Phương pháp lập trình với WCF

Ta đã biết có 2 phương pháp lập trình chính là hướng thủ tục và hướng đối tượng. Và bây giờ với WCF ta có thêm một phương pháp lập trình mới là hướng dịch vụ (Service oriented programming). Như vậy khi làm việc với WCF ta có hai lựa chọn là hướng đối tượng và hướng dịch vụ. Vậy ta nên sử dụng phương pháp nào? Câu trả lời là dùng cả hai. Nói một cách đơn giản là phương pháp hướng đối tượng được sử dụng để phát triển các ứng dụng trên desktop, còn phương pháp hướng dịch vụ được sử dụng để kết nối các ứng dụng đó với nhau. Điều quan trọng ở đây là làm sao để hiểu được sự khác nhau giữa hai phương pháp và hiểu được khi nào chúng được sử dụng và sử dụng như thế nào, đồng thời cũng phải hiểu về các lợi ích chúng cung cấp.

Về hướng đối tượng có thể hiểu như sau. Các ứng dụng hướng đối tượng là hai hay nhiều lớp phụ thuộc lẫn nhau và chia sẻ chung các kiểu dữ liệu. Những lớp này liên lạc với nhau thông qua các lời gọi hàm mà lớp đối tượng cung cấp. Còn các ứng dụng hướng dịch vụ là các chương trình không biết gì về nhau. Mỗi ứng dụng liên lạc với ứng dụng khác thông qua các bản tin. Điểm đặc biệt là các bản tin này được gửi từ một ứng dụng sang ứng dụng khác mà không quan tâm đến nền tảng mà dịch vụ đang chạy.

Ví dụ, lớp sau đây định nghĩa một giao diện hướng đối tượng

public interface IService1

{

public bool CheckUser(string tk, string mk)

{

}

Public bool CreateUser(string tk, string mk)

{

}

Public int CreateNewRoom(string tk, string mk, int soluong, int sodia, string tenphong);

}

Ta sẽ có phần hướng dịch vụ cho dịch vụ WCF khi thêm vào các thuộc tính cho giao diện ở trên:

[ServiceContract]

public interface IService1

{

[OperationContract]

bool CheckUser(string tk, string mk);

[OperationContract]

bool CreateUser(string tk, string mk);

[OperationContract]

int CreateNewRoom(string tk, string mk, int soluong, int sodia, string tenphong);

}

Qua ví dụ trên ta có thể thấy mối liên hệ giữa phương pháp hướng đối tượng và hướng dịch vụ trong WCF.

Có một số phương pháp lập trình với WCF, mỗi phương pháp có ưu khuyết điểm riêng của nó. Điều đặc biệt về WCF là luôn có hơn một cách để giải quyết vấn đề trong WCF, và ta không nhất thiết phải chọn duy nhất một phương pháp nào. Có 3 phương pháp được sử dụng khi phát triển dịch vụ WCF như sau:

* Phương pháp khai báo
* Phương pháp lập trình trực tiếp
* Phương pháp sử dụng tập tin cấu hình.

Trong phạm vi phát triển ứng dụng Chinh phục tháp Hà Nội, ta chỉ cần sử dụng phương pháp đầu tiên là đủ. Lập trình khai báo được thông qua các thuộc tính. Những thuộc tính này được sử dụng để định nghĩa các contract và xác định hành xử của dịch vụ. Chúng được sử dụng để xác định thêm các tham số để thay đổi các chi tiết của contract và hành xử dịch vụ.

Thuộc tính ServiceContract dùng để quy định là giao diện này định nghĩa các chức năng của một dịch vụ. Thuộc tính OperationContract được sử dụng ở các hàm để quy định rằng hàm này được khai báo là một phần của dịch vụ. Đó là tất cả những gì cần để tạo ra một dịch vụ WCF.

Thêm vào đó, ta không nhất thiết phải sử dụng các giao diện (interface) khi cài đặt một dịch vụ, điều này cũng giống như việc ta không cần sử dụng giao diện để định nghĩa một lớp. Tuy vậy ta nhất thiết phải chỉ rõ phần nào thuộc về dịch vụ. Ta có thể định nghĩa những phần khác cần cho giao diện, nhưng chỉ những phương thức có gắn thuộc tính [OperationContract].

Ví dụ ta có một dịch vụ thực hiện phép tính cộng giữa 2 số nguyên AddInt và 2 số thực AddDouble. Ta khai báo dịch vụ như sau:

[ServiceContract]

public interface ICalcService

{

[OperationContract]

int AddInt(int x, int y);

[OperationContract]

double AddDouble(double x, double y);

}

Như vậy dịch vụ của chúng ta sau khi khai báo sẽ có 2 phương thức được khao báo với thuộc tính OperationContract là AddInt và AddDouble. Tuy nhiên khi khai báo trong C#, việc đặt tên AddInt và AddDouble có thể rút gọn lại thành một tên hàm Add mà thôi. Nhưng trong các dịch vụ lại không cho phép đặt trùng tên hàm như thế. Chúng ta có thể khai báo thêm với thuộc tính OperationContract để thực hiện, cách làm như sau:

[ServiceContract]

public interface ICalcService

{

[OperationContract (Name="AddInt"]

int Add(int x, int y);

[OperationContract (Name="AddDouble")]

double Add(double x, double y);

}

Như vậy ta có thể sử dụng được phép nạp chồng tên trong C# và sử dụng thêm tham số name để quy định cách gọi tên hàm ở dịch vụ. Ngoài ưu điểm trong việc giải quyết nạp chồng tên hàm, ta còn thấy một lợi ích khác nữa là việc quy định tham số Name trong thuộc tính OperationContract còn cho ta thêm linh hoạt trong việc đổi tên các hàm trong giao diện mà không làm thay đổi định nghĩa dịch vụ, nghĩa là các ứng dụng khác sử dụng dịch vụ này không cần phải biên dịch lại.

##### Service Model

Mô hình dịch vụ trên WCF cũng tương tự với mô hình dịch vụ Web, điểm khác biệt là ở cách đặt tên. Trong WCF các thành phần không được gọi là Service, Binding, và PortType như của dịch vụ Web, mà được gọi tương ứng là Address, Binding và Contract. Mô hình dịch vụ WCF được cung cấp trong không gian tên System.ServiceModel. Không gian tên này chứa rất nhiều lớp, nhưng trong khuôn khổ xây dựng trò chơi Chinh phục tháp Hà Nội, ta chỉ sử dụng một số lớp sau đây: (chi tiết xem tại [MSDN](http://msdn.microsoft.com/en-us/library/system.servicemodel.aspx))

|  |  |
| --- | --- |
| Lớp | Mô tả |
| BasicHTTPBingding | Là binding các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách và dịch vụ web |
| NetTCPBinding | Là binding mà các điểm cuối dịch vụ có thể sử dụng để liên lạc với các ứng dụng khách/dịch vụ ở các máy khác nhau. |
| EndpointAddress | Lớp biểu diễn địa chỉ duy nhất được cung cấp và truy xuất được cho máy khách để liên lạc với điểm cuối dịch vụ |
| ServiceHost | Phương pháp cung cấp các vật chứa cho các dịch vụ. |

##### Địa chỉ trong WCF

Địa chỉ được sử dụng trong WCF có dạng như sau:

http://localhost:9999/ WebServiceTowerHanoi

Ở ví dụ trên, ta nhân thấy địa chỉ có 4 phần:

* Giao thức vận chuyển, ở trên là http:
* Tên của máy thực hiện dịch vụ, trong ví dụ là ://localhost
* Đường dẫn tới điểm cuối dịch vụ, trong ví dụ là /WebServiceTowerHanoi
* Phần tùy chọn là cổng dịch vụ, trong ví dụ là 9999. Nếu cổng dịch vụ không được chỉ ra thì giá trị mặc định là cổng 80 cho giao thức http: hay cổng 443 cho giao thức https:
* Trong WCF có các kiểu địa chỉ sau đây:
* Endpoint Adress (địa chỉ điểm cuối): địa chỉ này giống như ví dụ trên, một địa chỉ điểm cuối quy định địa chỉ của một điểm cuối dịch vụ cụ thể. Máy khách (chương trình khách) có thể truy cập dịch vụ qua địa chỉ điểm cuối. Ví dụ qua địa chỉ sau:

http://169.254.80.80/WebServiceTowerHanoi

Khi máy khách truy cập dịch vụ thông qua địa chỉ điểm cuối, máy khách có thể nói chuyện với dịch vụ và mọi liên lạc từ dịch vụ đều thực hiện thông qua địa chỉ này.

* Base Address (địa chỉ cơ sở): cung cấp cách để xác định một địa chỉ đơn nhất cho một dịch vụ và gán các địa chỉ tương đối cho từng điểm cuối riêng lẻ. Address formattings

Các địa chỉ điểm cuối được định dạng trên cơ sở của việc lựa chọn cách vận chuyển được sử dụng trong truyền thông. Điểm quan trọng cần nhớ là với bất kì định dạng nào, máy khách cần phải có khả năng lấy được địa chỉ tới dịch vụ và sử dụng dịch vụ. Phần lớn các định dạng địa chỉ chứa các phần sau:

* Phương thức (schema): xác định giao thức liên lạc
* Tên miền: xác định tên máy chứa dịch vụ
* Cổng: xác định cổng mà dịch vụ thực hiện, nếu không được quy định thì giá trị mặc định là 80
* Đường dẫn: đường dẫn đến dịch vụ

Thực chất, để phân biệt các định dạng địa chỉ khác nhau, ta căn cứ vào schema của địa chỉ. Ta có các định dạng địa chỉ sau:

* HTTP Address
* HTTPS Address
* TCP Address
* MSMQ Address
* Named Pipe Address
* IIS Address

Tower of Hanoi chỉ sử dụng HTTP Address, các địa chỉ khác sẽ không được trình bày tại đây (chi tiết xem thêm tại [MSDN](http://msdn.microsoft.com/en-us/library/aa751841.aspx)). HTTP là dạng địa chỉ được dùng nhiều, phổ biến. Định dạng của địa chỉ HTTP như sau:

http://domainname|hostname[:9999]/path

Định dạng địa chỉ HTTP chứa 4 phần. Sau đây là ví dụ của một địa chỉ HTTP sử dụng cả 4 phần:

http://169.254.80.80/WebServiceTowerHanoi

### Kiến trúc của client trong WCF

Một client trong WCF là một chương trình sử dụng các chức năng cung cấp bởi một dịch vụ WCF. Chương trình client sẽ liên lạc với dịch vụ thông qua điểm cuối dịch vụ. Để làm được điều này client cần phải biết một số thông tin về dịch vụ như địa chỉ của điểm cuối, binding mà dịch vụ sử dụng và contract dịch vụ.

Một trong những thứ mà ta có thể thấy trong kiến trúc của client là kênh thông tin được xây dựng dựa trên các cấu hình binding, những cấu hình này được quy định trong file cấu hình. Những binding này cho phép client và dịch vụ liên lạc với nhau một cách hiệu quả. Điểm thứ hai ta có thể thấy là cài đặt của giao diện IclientChannel. Giao diện này định nghĩa các thao tác cho phép nhà phát triển điều khiển các chức năng của kênh, như đóng phiên làm việc của client và hủy một kênh để thu hồi tài nguyên. Cuối cùng là contract dịch vụ sẽ được tạo ra tự động, nó cung cấp tính năng chuyển lời gọi phương thức ở phía client thành các bản tin đi và chuyển các bản tin tới thành thông tin mà chương trình client có thể sử dụng dưới dạng các giá trị trả về hay tham số đầu ra của phương thức.

Các client liên lạc với các điểm cuối dịch vụ thông qua một proxy. Sự liên lạc được thông qua một kênh. Sau khi proxy và kênh được tạo ra, client có thể truy xuất các phương thức có ở điểm cuối đó.



Hình 3. WCF Client Proxy

Có 2 cách để tạo ra các proxy:

* Tạo proxy từ mã nguồn được tạo ra tự động từ thông tin siêu dữ liệu. Siêu dữ liệu này được cung cấp bởi dịch vụ. Việc tạo mã nguồn được thực hiện thông qua việc sử dụng công cụ Svcutil.exe, công cụ này đi kèm với bộ cài đặt VS 2010 hoặc có thể lấy từ Microsoft.
* Tạo proxy thông qua mã nguồn sử dụng đối tượng ChannelFactory. Đối tượng này được cung cấp từ lớp System.ServiceModel.ChannelFactory. Phương pháp này cho phép nhà phát triển khả năng điều khiển nhiều hơn, ví dụ như tạo các kênh mới từ một channel factory có sẵn.

Ví dụ:

TaiKhoanNguoiChoiClient proxy = new TaiKhoanNguoiChoiClient();

Trong đó, TaiKhoanNguoiChoiClient là tên lớp dịch vụ mà máy khách nhận được khi kết nối thành công đến WCF Service. Trong lớp này chứa tất cả các phương thức mà ta đã dịnh nghĩa trong contract dịch vụ. Muốn sử dụng các phương thức trong contract tại máy khách ta cần tạo đối tượng proxy như trên. Muốn thực thi phương thức của dịch vụ thì thao tác như đối tượng thông thường: proxy.tên phương thức.

2.4 LINQ

## 2.5 WCF

## 2.6 MVVM

# **CHƯƠNG 3: ỨNG DỤNG XÂY DỰNG HỆ THỐNG**

## 3.1 ĐẶC TẢ YÊU CẦU PHẦN MỀM

### **3.1.1 GIỚI THIỆU**

#### **3.1.1.1 MỤC TIÊU**

Tài liệu đặc tả yêu cầu phần mềm mô tả tổng quan các chức năng của hệ thống và những ràng buộc chức năng mà phần mềm phải có. Tài liệu giúp người đọc có cái nhìn tổng quan về đặc điểm và chức năng của trò chơi Chinh phục tháp Hà nội.

Tài liệu là cơ sở để phân tích, thiết kế và triển khai xây dựng ứng dụng để đảm bảo tính chính xác, phù hợp và đầy đủ với yêu cầu đặc tả.

Ngoài ra, việc thiết kế tài liệu đặc tả một cách chi tiết giúp cho công việc ở các giai đoạn tiếp theo được diễn ra theo đúng kế hoạch đã định. Người thiết kế có cái nhìn đầy đủ, tổng quan về hệ thống và dễ dàng thực hiện các công việc thiết kế của mình.

#### **3.1.1.2 PHẠM VI SẢN PHẨM**

Mục tiêu chính của đề tài là phát triển nhằm nghiên cứu và xây dựng một ứng dụng mới, cụ thể là trò chơi Chinh phục tháp Hà nội trên nền tảng Silverlight -Windows Phone.

Tài liệu chỉ mô tả một cách tổng quá các yêu cầu chức năng và phi chức năng của trò chơi Chinh phục tháp Hà nội

#### **3.1.1.3 BỐ CỤC TÀI LIỆU**

Bố cục của tài liệu còn lại bao gồm:

* 2. Mô tả tổng quan.
* 3. Các yêu cầu giao tiếp bên ngoài.
* 4. Các yêu cầu chức năng của hệ thống.
* 5. Các yêu cầu phi chức năng.

### **3.1.2 MÔ TẢ TỔNG QUAN**

#### **3.1.2.1 BỐI CẢNH SẢN PHẨM**

Chinh phục tháp Hà Nội một trò chơi xuất hiện từ rất sớm và được đưa sang phương Tây lần đầu bởi nhà toán học người Pháp Edouard Lucas vào năm 1883. Trò chơi tháp Hà Nội ngày càng được các nhà toán học quan tâm. Với sự phát triển nhanh chóng của tin học, bài toán đã thu hút sự chú ý của của các toán học và tin học. Trò chơi tháp Hà Nội là một ví dụ điển hình về phương pháp giải đệ quy và phương pháp lập trình cơ bản..

Hiện tại, trò chơi chỉ hỗ trợ cho 1 người chơi vì vậy thực sự nhàm chán khi bảng xếp hạng thành tích chỉ có 1 mình. Khi đó sẽ kém phần hứng thú nếu không có ai chiêm ngưỡng được thành tích mà mỗi người chơi lập được.

Với sự phát triển của công nghệ hiện nay, việc phát triển nhanh chóng của các thiết bị di động với sự hỗ trợ cấu hình mạnh mẽ tạo điều kiện thuận lợi cho các lập trình viên phát triển các ứng dụng, đặc biệt là các ứng dụng trò chơi. Do đó, việc phát triển trò chơi chinh phục tháp Hà nội trên thiết bị di động với sự kết hợp của nhiều người chơi sẽ mang lại tính hấp dẫn, hứng thú mạnh mẽ so với việc chơi một mình.

Trò chơi cung cấp 2 chế độ chơi để người chơi lựa chọn: chơi đơn và thách đấu trực tuyến. Ở hai chế độ này về cơ bản sẽ không có sự khác nhau nhiều về cách chơi và cách tính điểm. Hai chế độ đều cho phép người chơi chọn số dĩa nhất định để di chuyển, cụ thể từ 3 đến 10 dĩa. Với chế độ chơi đơn, người chơi sẽ hoàn thành việc di chuyển dĩa từ cọc A sang cọc C với thời gian nhanh nhất có thể, chế độ này người chơi sẽ hiểu rõ quy luật chơi cũng như làm quen với việc tương tác với trò chơi. Với chế độ chơi thách đấu trực tuyến sẽ được thiết kế theo phòng, mỗi phòng sẽ có tối đa 2 người chơi.

Người chơi sẽ hoàn thành trò chơi bằng cách di chuyển thật nhanh tất cả các dĩa sang cọc C, điểm sổ sẽ được tính dựa vào thời gian và số lần di chuyển dĩa. Người chơi có số lần di chuyển ít nhất sẽ được lưu thông tin vào bảng xếp hạng. Người thua cuộc sẽ không được lưu lại thành tích của mình. Ở chế độ này, dựa vào cấp độ là số lượng dĩa mà có bảng xếp hạng tương ứng.

Bảng xếp hạng sẽ lưu lại thông tin của chơi đơn và thách đấu trực tuyến. Bảng thành tích này sẽ lưu lại thành tích tốt nhất mà người chơi hoàn thành được. Vì vậy, người chơi phá được kỉ lục đã lập trước đó sẽ lưu lại thông tin mới nhất. Ngược lại sẽ không cập nhật thông tin bảng xếp hạng.

Với ý tưởng phát triển trò chơi trên nền tảng thiết bị di động, kết hợp với chơi trực tuyến sẽ nhầm góp phần làm trò chơi trở nên phong phú hơn. Sau khi trò chơi được hoàn thành:

* Mô phỏng được giao diện của người chơi sao cho giống với bên ngoài thực tế đến mức có thể và kèm theo hiệu úng di chuyển, âm thanh trực quan nhất.
* Trò chơi phải cho người chơi lựa chọn cụ thể về số lượng đĩa
* Xác định luật chơi theo thực tế (có hướng dẫn cách chơi cho người chơi).
* Có sự hỗ trợ bất kì lúc nào khi người chơi nhấp “Trợ giúp”
* Trò chơi phải được tính điểm dựa vào thời gian và số lược di chuyển đĩa của người chơi.
* Chơi thách đấu trực tuyến

#### 3.1.2.2 CÁC CHỨC NĂNG CƠ BẢN

|  |  |  |
| --- | --- | --- |
| STT | Tên chức năng | Mô tả chi tiết |
| 1 | Đăng nhập | Chức năng này cho phép người chơi đã đăng kí tài khoản trước đó có thể đăng nhập và tiến hành chơi game. Mỗi người chơi sẽ có một tên đăng nhập để nhằm hiển thị điểm đạt được thông qua mỗi cấp độ của trò chơi |
| 2 | Đăng kí | Cho phép người chơi mới đăng kí một tài khoản để chơi, tài khoản này sẽ lưu trữ điểm số mỗi khi người chơi thực hiện chơi game. |
| 3 | Chơi game | Cho phép người chơi chọn số lượng đĩa ban đầu trước khi bắt đầu. Bắt đầu với 3 cột A, B, C, Mục tiêu trong trò chơi này là di chuyển tất cả các vòng từ cột A đến cột C và xếp chúng theo thứ tự ban đầu qua trung gian cột B trống. Điểm số được đưa ra dựa trên hai yếu tố: thời gian và số lượng vòng đúng di chuyển được. |
| 4 | Bảng xếp hạng | Hiển thị xếp hạng theo 2 cách thức:  +Hiển thị thông tin các kỉ lục mà mỗi người chơi thực hiện ở các cấp độ chơi khác nhau  +Hiển thị thông tin tổng hợp của các người chơi có trong dữ liệu lưu trữ. |
| 5 | Bật/Tắt âm thanh | Chức năng này cho phép người chơi tùy chọn âm thanh nền, âm thanh chạm khi chơi. Ngoài ra trò chơi còn hỗ trợ điều chỉnh cường độ âm thanh theo ý muốn của người chơi |
| 6 | Trợ giúp | -Hỗ trợ giao diện mô phỏng lại cách thức di chuyển các cọc với số lượng đĩa do người chơi chọn trước đó.  -Hiển thị trợ giúp thông qua danh sách các bước di chuyển |
| 7 | Hướng dẫn luật chơi | Hiển thị thông tin hướng dẫn cách thức chơi game cho người chơi hiểu tổng quát về quy luật chơi game. |
| 8 | Hiển thị danh sách phòng chơi | Sau khi người chơi chọn thách đấu trực tuyến, hệ thống sẽ liệt kê danh sách phòng chơi hiện đang có trên máy chủ để người chơi xem và có thể lựa chọn. |
| 9 | Tạo phòng mới | Cho phép người chơi có thể tạo phòng chơi mới |
| 10 | Rời khỏi phòng chơi | Chức năng này cho phép người chơi có thể rời khỏi phòng chơi khi cần thiết. |
| 11 |  |  |

#### 3.1.2.3 ĐẶC ĐIỂM NGƯỜI SỬ DỤNG

Đối tượng chủ yếu của trò chơi không phân biệt sự hiểu biết về công nghệ quá cao, phù hợp với mọi lứa tuổi. Người chơi chỉ cần biết cách thao tác trên các thiết bị cảm ứng kết hợp với tìm hiểu quy luật của trò chơi là có thể trải nghiệm được.

#### 3.1.2.4 MÔI TRƯỜNG VẬN HÀNH

Trò chơi Chinh phục tháp Hà nội được phát triển để chạy trên thiết bị di dộng, cụ thể là hệ điều hành Windows Phone 8. Vì vậy, các yêu cầu cần thiết để ứng dụng có hệ hoạt động hiểu quả là:

* Phần cứng: RAM 512MB, bộ nhớ trong còn trống 100MB trở lên
* Hệ điều hành: Windows Phone 8.0 hoặc cao hơn.

#### 3.1.2.5 CÁC RÀNG BUỘC THỰC THI VÀ THIẾT KẾ

Thực thi:

* Phần mềm chạy trên nền Windows Phone.
* Giao diện thân thiện với người sử dụng, trực quan không mang tính công nghệ cao, không yêu cầu tính thẩm mỹ quá cao.
* Tương tác được với các thao tác trên thiết bị di động.

Thiết kế:

* Ngôn ngữ viết phần mềm: C#.
* Công nghệ Silverlight
* Công cụ hỗ trợ thiết kế và lập trình: Visual Studio Express for Windows Phone (SDK 8.0), Adobe Photoshop CS5, Blend for Visual Studio
* Ngôn ngữ XAML

#### 3.1.2.6 CÁC GIẢ ĐỊNH VÀ PHỤ THUỘC

Người chơi xóa (làm rỗng) bảng xếp hạng thì bắt buộc người chơi phải chơi lại ít nhất 1 lần để hiển thị lại bảng xếp hạng.

Khi người chơi đang chơi ở một cấp độ nào đó, khi có sự cố hoặc người chơi thoát đột ngột thì ứng dụng phải lưu lại trạng thái hiện tại.

Người thu thập yêu cầu phần mềm thiếu kinh nghiệm, thu thập yêu cầu không đầy đủ chi tiết dẫn đến thiếu chức năng của ứng dụng

Người phát triển thiếu kiến thức, phải rèn luyện hoặc kiến thức tiếp thu chưa vững dẫn đến chậm trễ về mặt thời gian

Do áp lực về công việc và thời gian khi chưa hiểu rõ về công nghệ, thuật toán để giải quyết bài toán.

### 3.1.3 CÁC YÊU CẦU GIAO TIẾP NGOÀI

#### 3.1.3.1 GIAO TIẾP NGƯỜI SỬ DỤNG

Giao diện thân thiện với người sử dụng, kết hợp với các thao tác chạm trên màn hình cảm ứng của thiết bị di động. Màu chủ đạo là màu xanh xanh.

Thiết kế giao diễn trách các lỗi về màu sắc như không tương phải, không làm nội dung nổi bật.

Các nút chuẩn và các giao diện hiển thị thông báo phải đồng nhất với nhau trong suốt toàn bộ ứng dụng. Khi người chơi chọn thao tác bất kì thì giao diện hiển thị thay đổi để người chơi dễ nhận biết thao tác vừa chọn.

Giao diện có sự liên kết chặt chẽ với nhau, cụ thể người chơi ở bất cứ vị trí nào có thể điều hướng về giao diện trước đó.

#### 3.1.3.2 GIAO TIẾP PHẦN CỨNG

Để sử dụng ứng dụng mượt mà điện thoại cần có cầu hình phần cứng tương đối. RAM từ 512 MB trở lên.

Chip Qualcomm MSM7x30, MSM8x55

Độ phân giải màn hình 480x800

Màn hình nhận diện 4 điểm tiếp xúc

Camera ít nhất 5MP

#### 3.1.3.3 GIAO TIẾP PHẦN MỀM

Để thao tác với chương trình người dùng phải có thiết bị chạy Windows Phone (smartphone hoặc tablet) từ phiên bản 8.0 trở lên.

BIOS phải hỗ trợ những số tính năng sau: bộ xử lý phải hỗ trợ Second Level

Address Translation – SLAT và bật Hyper-V

Microsoft Visual Studio 2012 Express for Windows Phone

Windows Phone Emulator

Windows Phone SDK 8.0 Assemblies

Silverlight 4 SDK and DRT

Windows Phone SDK 8.0 Extensions for XNA Game Studio 4.0

Microsoft Expression Blend SDK for Windows Phone 8

Microsoft Expression Blend SDK for Windows Phone OS 8.1

WCF Data Services Client for Window Phone

Microsoft Advertising SDK for Windows Phone

#### 3.1.3.4 GIAO TIẾP TRUYỀN THÔNG TIN

Ứng dụng sử dụng giao thức truyền tin “Basic HTTP Binding in WCF” trong chế độ thách đấu trực tuyến.

## 3.1.4 CÁC TÍNH NĂNG HỆ THỐNG

### 3.1.4.1 TÍNH NĂNG ĐĂNG KÍ

Mô tả và mức ưu tiên

Chức năng này được sử dụng để đăng kí tài khoản để có thể vào ứng dụng game. Chức năng này có mức ưu tiên cao.

Tác nhân / Chuỗi đáp ứng

Người chơi lần đầu cài đặt ứng dụng phải đăng kí một tài khoản, cụ thể người chơi phải nhập các thông tin và tiến hành đăng kí. Sau khi đăng kí thành công, người chơi đăng nhập và ứng dụng sẽ lưu thông tin điểm ở mỗi mức độ khác nhau.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ01 |
| Tên chức năng | Đăng kí |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Phải cài đặt ứng dụng game thành công về thiết bị di động, truy cập vào game |
| Cách xử lý | Bước 1: Người chơi nhập tên đăng nhập, mật khẩu, xác nhập mật khẩu.  Bước 2: Ứng dụng kiểm tra hợp lệ và tiến hành lưu vào CSDL.   * Nếu hợp lệ, báo thành công và chuyển giao diện cho người chơi về trạng thái đã đăng nhập * Nêu không, báo lỗi cụ thể khi người chơi không nhập bất kì trường thông tin nào. |
| Kết quả | Người chơi đăng kí thành công hoặc thất bại |
| Ghi chú | Các trường hợp thông báo đăng kí không thành công bao gồm:  -Tên đăng nhập rỗng  -Mật khẩu rỗng  -Xác nhận mật khẩu rỗng  -Mật khẩu và xác nhận mật khẩu không trùng khớp |

### 3.1.4.2 TÍNH NĂNG ĐĂNG NHẬP

Mô tả và mức ưu tiên

Mỗi người chơi lần đầu vào trò chơi phải đăng nhập vào ứng dụng. Chức năng này có độ ưu tiên cao.

Tác nhân / Chuỗi đáp ứng

Người chơi tiến hành đăng nhập vào ứng dụng, ứng dụng sẽ kiểm tra tên đăng nhập và mật khẩu có tồn tại hay không. Nếu tồn tại, ứng dụng chuyển đến giao diện chính cho phép người chơi chọn chức năng.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ02 |
| Tên chức năng | Đăng nhập vào hệ thống |
| Đối tượng sử dụng | Nhóm người dùng |
| Tiền điều kiện | Phải có tên đăng nhập và password trong hệ thống |
| Cách xử lý | Bước 1: Người dùng nhập tên đăng nhập và password.  Bước 2: Kiểm tra hợp lệ form đăng nhập. Hợp lệ khi tên đăng nhập và password trùng khớp với cơ sở dữ liệu.   * Nếu hợp lệ, báo thành công và chuyển người dùng về trạng thái đã đăng nhập * Nêu không, báo lỗi sai username hoặc password. Nếu sai, quá 3 lần, hiển thị captchar |
| Kết quả | Người dùng đăng nhập thành công vào hệ thống. Nếu thất bại, hiện bảng thông báo lỗi “Bạn đã nhập sai tên đăng nhập hoặc mật khẩu” và yêu cầu nhập lại. |
| Ghi chú | Tên đăng nhập là mã cán bộ hoặc mã sinh viên hoặc mã admin, là kí tự bắt đầu bằng chữ cái, chuỗi không chứa các kí tự đặc biệt.  Password là chuỗi kí tự bất kì từ 6 đến 15 kí tự, chuỗi không chứa kí tự đặc biệt. Các kí tự đăc biệt bao gồm ~ ! @ # $ % ^ & \* ( ) - + ? <> : ; ‘ { } [ ] ,. |

### 3.1.4.3 TÍNH NĂNG CHƠI GAME

Mô tả và mức ưu tiên

Chức năng này hỗ trợ người chơi thực hiện các thao tác di chuyển các đĩa với số lượng đã chọn trươc đó. Khi người chơi di chuyển đúng sẽ cập nhật số lần chuyển đĩa và hiển thị đĩa lên cột đã di chuyển thành công. Trường hợp người chơi di chuyển sai sẽ không chuyển đĩa được và trạng thái đĩa không thay đổi. Chức năng này có độ ưu tiên cao.

Tác nhân / Chuỗi đáp ứng

Người chơi đã đăng nhập thành công vào ứng dụng và chọn tính năng “Chơi”. Ứng dụng hiển thị giao diện cho người chơi với số lượng đĩa đã được chọn trước đó. Ứng dụng dựa vào sự kiện chạm và giữ để quyết định di chuyển đĩa theo mong muốn của người chơi.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ03 |
| Tên chức năng | Chơi game |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập và chọn tính năng thành công. Ứng dụng hiển thị giao diện cho phép người chơi chọn số lượng đĩa nhất định. Mặc định số lượng đĩa được chọn là 3 đĩa.  Bước 2: Người chơi chọn “Bắt đầu”, ứng dụng dựa vào số lượng đĩa đã chọn để hiển thị lên màn hình và bắt đầu tính thời gian cdi chuyển đĩa của người chơi.  Bước 3: Ứng với mỗi lần di chuyển đĩa mà người chơi chọn, ứng dụng sẽ so sánh theo quy tắt đĩa nhỏ phải nằm trên đĩa lớn. Thứ hai, mỗi lần chỉ chuyển được một đĩa di nhất. Tính số lần di chuyển đĩa thêm 1 đơn vị khi di chuyển đúng hoặc sai vị trí. Khi chuyển hết tất cả đĩa sang cọc C thì chuyển sang bước 4.  Bước 4: Thông báo chiến thắng cho người chơi và hiển thị lại giao diện cho phép người chơi thực hiện chơi lại. |
| Kết quả | Di chuyển đĩa thành công và thông báo thắng cuộc nếu di chuyển tất cả các đĩa từ cọc A sang cọc C. |
| Ghi chú | -Mỗi lần chỉ chuyển được một đĩa duy nhất và chọn địa nhỏ nhất trong mỗi cọc.  -Chỉ duy chuyển được đĩa nhỏ đặt trên đĩa lớn hơn nó và ngược lại thì không.  -Có âm anh và hiệu ứng phù hợp cho lần chuyển đĩa đúng và sai.  -Tính số lần di chuyển đĩa và hiển thị thời gian đã thực hiện tính tới thời điểm hiện tại. |

### 3.1.4.4 TÍNH NĂNG XEM BẢNG XẾP HẠNG

Mô tả và mức ưu tiên

Tính năng này cho phép người chơi xem danh sách thành tích của mình hay danh sách thành tích chung của của người chơi. Tính năng này có độ ưu tiên trung bình.

Tác nhân / Chuỗi đáp ứng

Người chơi sau khi đăng nhập thành công vào ứng dụng, chọn chức năng xem bảng xếp hạng. Ứng dụng sẽ hiển thị danh sách bảng xếp hạng với 2 tiêu chỉ lớn là thành tích cá nhân và thành tích chung.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ04 |
| Tên chức năng | Bảng xếp hạng |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập và chọn tính năng xem bảng xếp hạng thành công.  Bước 2: Người chơi chọn “Thành tích cá nhân” hoặc “Thành tích chung”.  Bước 3: Ứng dụng sẽ thực hiện các thao các truy xuất CSDL để hiển thị. Nội dung hiển thị bao gồm:  + STT  + Tên người chơi  + Số lần di chuyển  + Thời gian  + Cấp độ: từ 1 đến 10  Bước 4: Người chơi có thể chọn xóa tất cả thành tích. Ứng dụng xác nhận thao tác xóa và hiển thị lại danh sách bảng xếp hạng. |
| Kết quả | Xem bảng xếp hạng thành công hoặc xóa thành công bảng xếp hạng. |
| Ghi chú | -Khi người chơi xác nhận xóa thì ứng dụng xóa vĩnh viễn ra khỏi dữ liệu và không phục hồi được.  -Bảng thành tích cá nhân được sắp xếp theo điểm số, tiếp theo là thời gian, số lần di chuyển. |

### 3.1.4.5 TÍNH NĂNG BẬT/TẮT ÂM THANH

Mô tả và mức ưu tiên

Tính năng này cho phép người chơi có thể lựa chọn bật.tắt âm thanh nền và âm nhanh di chuyển. Mở rộng thêm cho người chơi điều chỉnh cường độ âm thanh theo ý muốn. Tính năng này có độ ưu tiên trung bình thấp.

Tác nhân / Chuỗi đáp ứng

Người chơi đã đăng nhập thành công vào ứng dụng và chọn cài đặt, ứng dụng hiển thị giao diện điều chỉnh âm thanh cho người chơi điều chỉnh.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ05 |
| Tên chức năng | Bật/Tắt âm thanh |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập thành và chọn chức năng “Bật/tắt”  Bước 2: Giao diện hiển thị trạng thái biểu tượng âm thanh theo ý định của người chơi. Nếu người chơi cần điều chỉnh cường độ âm thanh thì sang bước 3  Bước 3: Người chơi chọn “Cài đặt” và ứng dụng hiển thị thang điều chỉnh âm thanh nhạc nền và âm thanh hiệu ứng. |
| Kết quả | Âm thanh được bật/tắt thành công và cường độ âm thanh được điều chỉnh theo tùy chọn của người chơi. |
| Ghi chú | -Khi âm thanh bật/tăt cần có biểu tượng để hiển thị trạng thái của âm thanh. |

### 3.1.4.6 TÍNH NĂNG TRỢ GIÚP

Mô tả và mức ưu tiên

Tính năng này sẽ mô tả lại các bước chuyển theo mức độ mà người chơi chọn. Tính năng này có độ ưu tiên trung bình.

Tác nhân / Chuỗi đáp ứng

Người chơi yêu cầu hiển thị trợ giúp, khi đó ứng dụng sẽ di chuyển các dĩa tương ứng với số lượng người chơi đã chọn.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ06 |
| Tên chức năng | Trợ giúp |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập thành và chọn chức năng trợ giúp.  Bước 2: Giao diện hiển thị lại các bước chuyển ứng với số đĩa đã chọn trước đó. Hiển thị thêm bước chuyển dạng văn bản cho người chơi dễ theo dõi. |
| Kết quả | Thực hiện trợ giúp người chơi thành công theo từng mức độ khác nhau |
| Ghi chú | -Quy tắc trợ giúp cũng tương tự như chơi thật: mỗi lần di chuyển được 1 đĩa và đĩa nhỏ phải nằm trên đĩa lớn. |

### 3.1.4.7 TÍNH NĂNG HƯỚNG DẪN LUẬT CHƠI

4.1.1 Mô tả và mức ưu tiên

4.1.2 Tác nhân / Chuỗi đáp ứng

4.1.3 Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ07 |
| Tên chức năng | Hướng dẫn luật chơi |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập thành và chọn chức năng hướng dẫn luật chới  Bước 2: Giao diện hiển thị thông tin luật chơi cụ thể để người chơi có thể nắm rõ được quy tắt chơi. Thông tin cụ thể như sau:  Luật chơi Chinh phục Tháp Hà nội dựa trên bài toán Tháp Hà nội truyền thống. **Bắt đầu với 3 cột A, B, C, Mục tiêu trong trò chơi này là di chuyển tất cả các vòng từ cột A đến cột C và xếp chúng theo thứ tự ban đầu qua trung gian cột B trống.** Khi trò chơi bắt đầu, có thể thiết lập số lượng các vòng (chẳng hạn từ 1-10). Điểm số được đưa ra dựa trên hai yếu tố: thời gian và số lượng vòng đúng di chuyển được. |
| Kết quả | Người chơi xem thông tin luật chơi thành công |
| Ghi chú |  |

### 3.1.4.8 TÍNH NĂNG HIỂN THỊ DANH SÁCH PHÒNG CHƠI

Mô tả và mức ưu tiên

Tính năng này sẽ hiển thị danh sách các phòng hiện có trên máy chủ cho phép người chơi có thể lựa chọn để tham gia vào thách đấu. Chức năng này có độ ưu tiên trung bình

Tác nhân / Chuỗi đáp ứng

Người chơi phải đăng nhập vào hệ thống và hệ thống sẽ hiển thị danh sách phòng cho người chơi lựa chọn.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ08 |
| Tên chức năng | Hiển thị danh sách phòng chơi |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi đăng nhập thành công, hệ thống hiện thị danh sách các phòng đang có trên máy chủ. |
| Kết quả | Hiển thị danh sách phòng chơi thành công hoặc thất bại |
| Ghi chú | Yêu cầu khi hiển thị các phòng chơi:  + Hiển thị cấp độ (số dĩa) của mỗi phòng.  + Trạng thái của phòng chơi còn trống, đầy với màu sắc cụ thể xanh cho phòng còn trống, đỏ cho phòng đã đủ người chơi. |

### 3.1.4.9 TÍNH NĂNG TẠO MỚI PHÒNG

Mô tả và mức ưu tiên

Chức năng này cho phép người chơi có thể tạo phòng mới với ràng buộc về mật khẩu và số dĩa cụ thể. Chức năng này có độ ưu tiên cao

Tác nhân / Chuỗi đáp ứng

Người chơi sau khi đăng nhập thành công vào ứng dụng, chọn chức năng tạo phòng mới. Ứng dụng sẽ cho phép tạo 1 phòng chơi mới.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ09 |
| Tên chức năng | Tạo phòng mới |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Tại giao diện danh sách phòng chơi, người chơi chạm tạo mới phòng chơi.  Bước 2: Ứng dụng tạo phòng mới, người chơi sẽ được đưa vào phòng chơi với các thông tin mặc định: tên phòng và chủ phòng là tên tài khoản người chơi, không đặt mật khẩu cho phòng, số lượng người chơi trong phòng là 1. Nếu người chơi có chọn mật khẩu thì sẽ nhập vào trường mật khẩu.  Bước 3: Ứng dụng sẽ hiển thị phòng vừa mới khởi tạo lên danh sách phòng chơi để người chơi thách đấu có thể vào tranh tài bất cứ lúc nào. |
| Kết quả | Tạo phòng thành công hoặc thất bại |
| Ghi chú | Nếu không tạo được phòng do lỗi nào đó, thì cần hiển thị thông báo tạo phòng thất bại.  Mặc định trường mật khẩu sẽ là rỗng mang ý nghĩa không đặt mật khẩu cho phòng chơi |

### 3.1.4.10 TÍNH NĂNG RỜI PHÒNG CHƠI

Mô tả và mức ưu tiên

Chức năng này cho phép người chơi ở có thể rời phòng chơi bất kì thời gian nào. Chức năng này có độ ưu tiên trung bình.

Tác nhân / Chuỗi đáp ứng

Người chơi sẽ yêu cầu rời phòng chơi hiện tại, hệ thống sẽ kiểm tra và cập nhật lại số lượng người chơi.

Các yêu cầu chức năng

|  |  |
| --- | --- |
| Mã yêu cầu | REQ10 |
| Tên chức năng | Rời khỏi phòng |
| Đối tượng sử dụng | Người chơi |
| Tiền điều kiện | Đăng nhập thành công vào ứng dụng |
| Cách xử lý | Bước 1: Người chơi chọn nút Rời phòng chơi  Bước 2: Hệ thống hiển thị xác nhận có muốn rời phòng chơi hay không.  Bước 3: Nếu người chơi là chủ phòng, khi chọn rời phòng chơi. Ứng dụng sẽ tiến hành xóa bỏ phòng chơi đó. Ngược lại, nếu người chơi là người thách đấu đã tham gia. Ứng dụng sẽ cập nhật lại trạng thái người chơi và không xóa bỏ phòng chơi. |
| Kết quả | Rời phòng chơi thành công hoặc thất bại |
| Ghi chú |  |

## 3.1.5 CÁC YÊU CẦU PHI CHỨC NĂNG

### 3.1.5.1 YÊU CẦU THỰC THI

* Tốc độ xử lý của hệ thống phải nhanh chóng với thời gian đáp ứng đặt vé trễ nhất là 5 giây và chính xác 99%.
* Âm thanh phải thực hiện đúng với thao tác của người chơi theo từng trường hợp.
* Khi người dùng thực hiện các thao tác di chuyển đĩa thì phải đáp ứng đúng yêu cầu của người chơi
* Dữ liệu nhập vào của người chởi được kiểm tra để đảm bảo đúng kiểu.
* Ứng dụng phải chuyển đổi giữa các chức năng tốt và tức thời.
* Khi gặp vấn đề trong việc tải dữ liệu phải thông báo cho người biết để chờ hoặc khởi động lại ứng dụng.

### **3.1.5.2 YÊU CẦU AN TOÀN**

* Ứng dụng hoạt động độc lập và không liên kết với các ứng dụng gây hại khác.
* Ứng dụng không ảnh hưởng đến các thành phần và dữ liệu trong thiết bị với hệ điều hành đang sử dụng.
* Người chơi quản lý tốt tài khoản của mình có thể xóa thành tích cá nhân của mình nhưng không xóa thành tích của người chơi khác được.

### **3.1.5.3 YÊU CẦU BẢO MẬT**

### **3.1.5.4 CÁC ĐẶC ĐIỂM CHẤT LƯỢNG PHẦN MỀM**

* Có tính hướng đối tượng và được cài đặt theo module thiết kế trước đó để thuận tiện cho công việc kiểm thử
* Ứng dụng phải có tính tái sử dụng các module về âm thanh.
* Giao diện của ứng dụng phải thân thiện, dễ sử dụng, không gây khó chịu cho người sử dụng khi họ sử dụng trong thời gian dài.
* Cơ sở dữ liệu có thể chứa được dữ liệu của người chơi ít nhất là 100 người.
* Ứng dụng phải đảm bảo được sự chính xác, nhanh chóng và hiệu quả khi thực hiện các chức năng.
* Ứng dụng phải có tính có thể bảo trì trong tương lai.
* Mã nguồn phải được viết rõ ràng. Dùng các từ có ý nghĩa, ngắn gọn và có tính gợi nhớ để đặt tên cho biến. Các hàm hoắc các xử lý phức tạp cần phải có chú thích.
* Phần mềm và tài liệu phải có sự thống nhất với nhau.
* Tài liệu cần phải mô tả các chức năng về xử lý, cũng như các truy xuất CSDL.
* Đảm bảo thực hiện chính xác, nhanh chóng và hiệu quả các chức năng đã đặt ra.
* Ứng dụng có thể bảo trì trong tương lai.
* Đối với phạm vi lập trình: có chú thích các dòng lệnh, các hàm quan trọng trong mã nguồn. Trung bình tối thiểu 1 dòng chú thích trên 20 dòng lệnh.

### **3.1.5.5 CÁC QUY TÁC NGHIỆP VỤ**

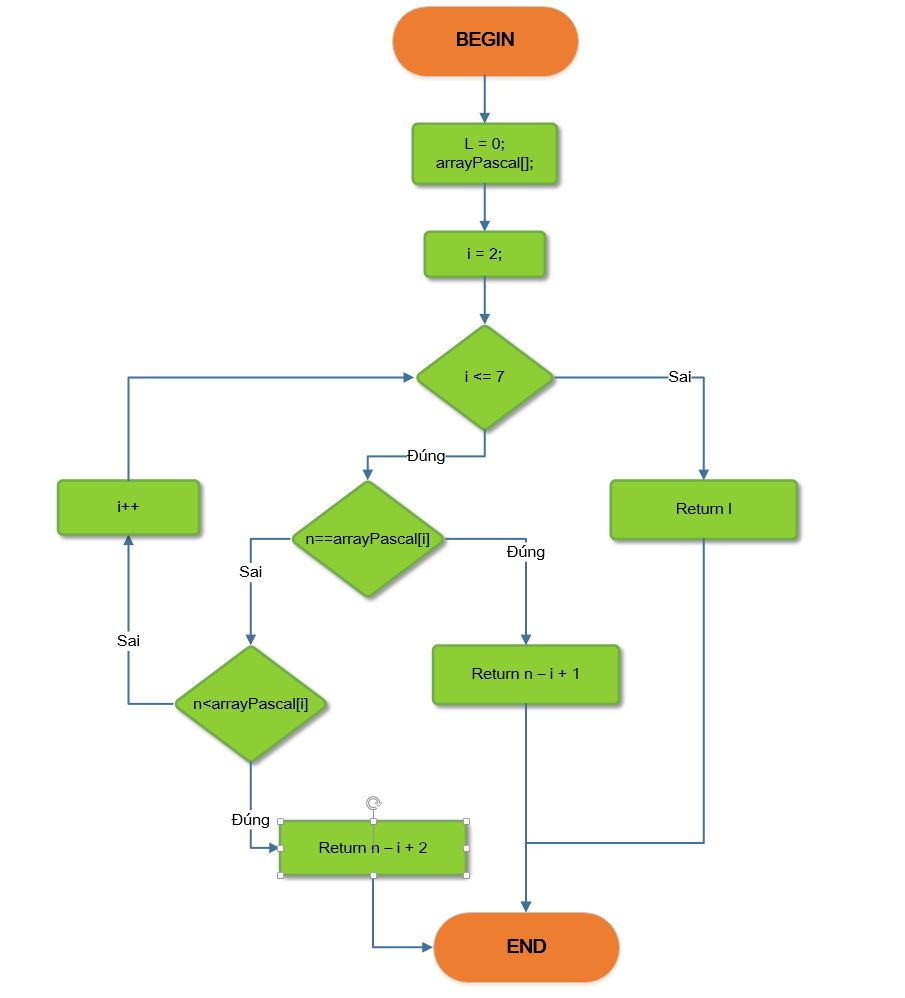
* Người sử dụng cần phải tham khảo tài liệu hướng dẫn sử dụng để có thể nắm rõ các chức năng cũng như cách thức thực hiện chúng.
* Khi xảy ra lỗi, người dùng cần liên hệ đến người phát triển ứng dụng.

## 3.1.6 CÁC YÊU CẦU KHÁC

* Yêu cầu về phong tục – văn hóa: Trò chơi Chinh phục tháp Hà nội có giao diện phù hợp với mọi người dùng khi sử dụng, không có hình ảnh hay tiêu ngữ không phù hợp với những điểm riêng của mỗi dân tộc, thành phần trong xã hội.
* Yêu cầu về pháp luật: Trò chơi Chinh phục tháp Hà nội phù hợp quy tắc pháp luật của nước Việt Nam, không tạo ra với mục đích trái với ứng dụng.
* Khả năng phục hồi và chịu lỗi.

## 3.1.7 XÂY DỰNG GIẢI THUẬT

#### **3.1.7.1 GIẢI THUẬT TÌM HỆ SỐ CHIA TỐI ƯU**



*Hình : Lưu đồ giải thuật xác định hệ số chia tối ưu*

Giải thuật này giải quyết bài toán tháp hà nội 4 cọc.

#### **3.1.7.2 GIẢI THUẬT TRỢ GIÚP VỚI 3 CỌC**

Nếu n = 1 hoặc n = 2 thì bài toán được giải quyết ngay.

Giả sử đã biết cách giải bài toán với n – 1 đĩa. Giải bài toán cho n đĩa như sau:

Chuyển n – 1 đĩa trên cùng từ cọc A sang cọc B (theo giả thuyết đã biết cách giải)

Chuyển đĩa thứ n (đĩa dưới cùng trên cọc A) từ cọc A sang cọc C (Bài toán 1 đĩa)

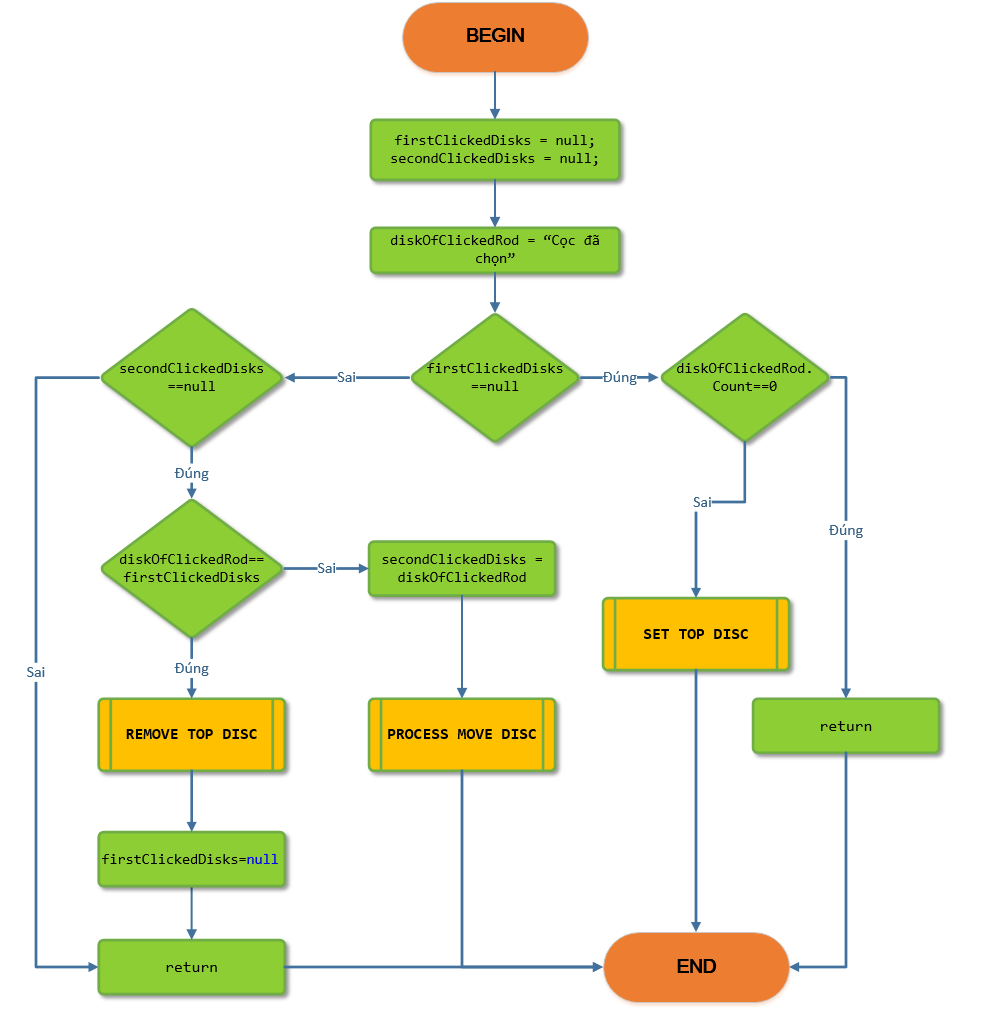
Chuyển n – 1 đĩa từ cọc B sang cọc C ( theo giả thuyết biết cách giải)

Như vậy, lời giải bài toán rất đơn giản: giải bài toán n đĩa được đưa về n – 1 đĩa và bài toán 1 đĩa. Ta kí hiệu L(n) là số lần chuyển đĩa tối ưu trong bài toán tháp Hà Nội với n đĩa và 3 cọc.

Khi đó, L(n) = 2n – 1 là số lần chuyển đĩa tối ưu.

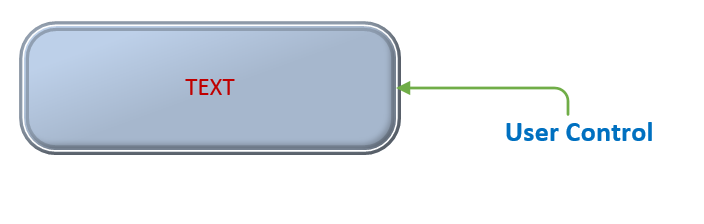
#### **3.1.7.3 GIẢI THUẬT TRỢ GIÚP VỚI 4 CỌC**

#### **3.1.7.4 KIỂM TRA ĐĨA CÓ THỂ DI CHUYỂN ĐƯỢC**



*Hình : Lưu đồ kiểm tra đĩa có thể di chuyển được*

Mục đích chính là kiểm tra khi người chơi chạm vào vị trí các cọc nào đó giải thuật sẽ được gọi để kiểm tra di chuyển của người chơi có hợp lệ hay không. Để hiểu rõ hơn về cấu tạo của một đĩa ta sẽ tìm hiểu cấu trúc cụ thể của nó như sau:



*Hình : Kiến trúc của 1 đĩa*

Mỗi đĩa sẽ bao gồm hai phần: Rectagle (quyết định độ chiều rộng và chiều cao của mỗi đĩa) và TextBlock (hiển thị giá trị số thứ tự của đĩa) được nằm trong lớp UserControl. Khi bắt đầu trò chơi, người chơi sẽ chọn số đĩa tương ứng và ứng dụng sẽ gọi hàm khởi tạo và thêm vào cọc A với số đĩa tương ứng. Ứng dụng sẽ chỉ kiểm tra sự kiện chạm vào cọc thứ nhất và cọc thứ hai cho nên phần sử lý sự kiện chạm các đĩa sẽ không cài đặt ở đây. Mục đích của việc tạo một đối tượng đĩa trong lớp UserControl nhằm để thêm nhiều đĩa vào một cọc bất kì.

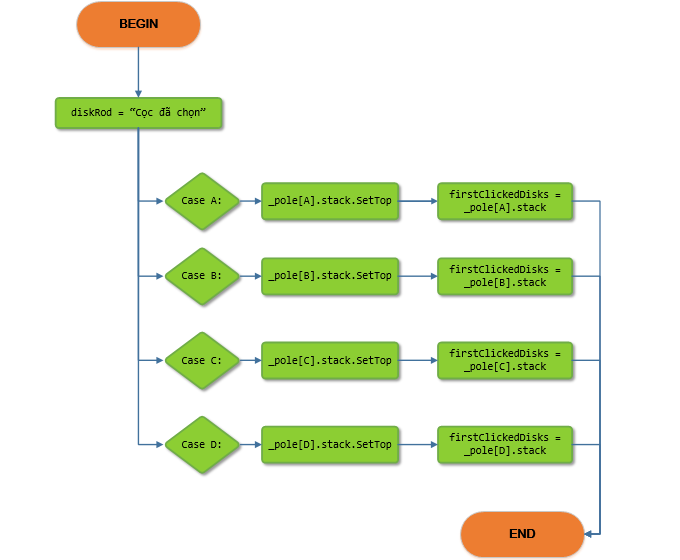
Quay trở lại với giải thuật, vấn đề di chuyển được mô tả như sau:

Nếu firstClickDisc bằng null nghĩa là người chơi chọn lần thứ nhất thì tiến hành di chuyển đĩa lên trên vị trí cao nhất của mỗi cọc.

Ngược lại kiểm tra secondClickDisc bằng null thì kiểm tra một trong hai trạng thái.

* Trạng thái a: Người chơi chọn đúng cọc lúc đầu, thì bỏ chọn đĩa (di chuyển đĩa lại vị trí ban đầu) và không thêm đĩa vào bất cứ cọc nào khác
* Trạng thái b: Người chơi chọn cọc thứ hai khác cọc thứ nhất thì gọi hàm con “Process Move Disc”. Ở trạng thái này ứng dụng sẽ không kiểm tra đĩa nhỏ có nằm trên đĩa lớn hay không mà sẽ để cho hàm con xử lý.

1. Lưu đồ con “SET TOP DISC”



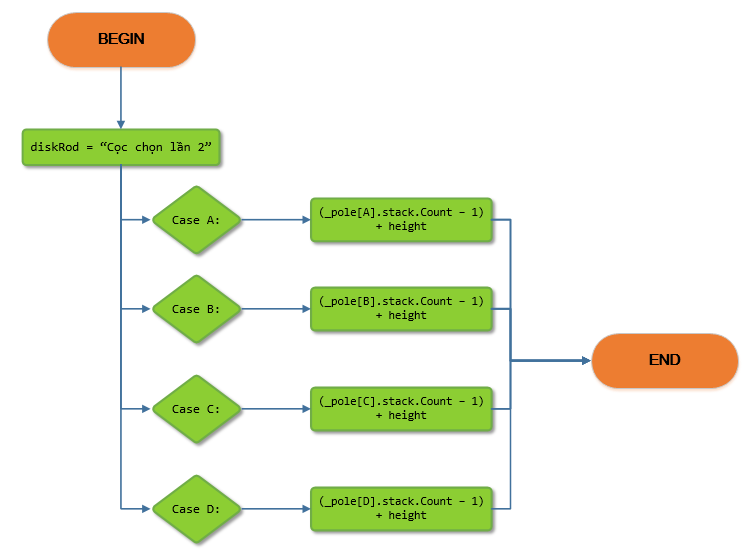
*Hình : Lưu đồ di chuyển dĩa lên trên mỗi cọc khi chọn lần đầu (chọn đĩa)*

1. Lưu đồ con “REMOVE TOP DISC”

Khi người chơi nhấp lần thứ nhất vào 1 cọc bất kì thì sẽ lưu lại tên cọc đã nhấp chọn lần 1 hiểu như là hành động CHỌN ĐĨA. Sau đó kiểm tra nếu người người chơi nhấp lần 2 đúng với cọc lần 1 trước đó thì xem hành động đó là BỎ CHỌN ĐĨA.

Ý tưởng để chuyển đĩa về vị trí ban đầu như sau: Vị trí của đĩa sẽ được tính bằng công thức:

x = Chiều cao cọc – (số lượng đĩa trong cọc - 1) \* Chiều cao mỗi đĩa)

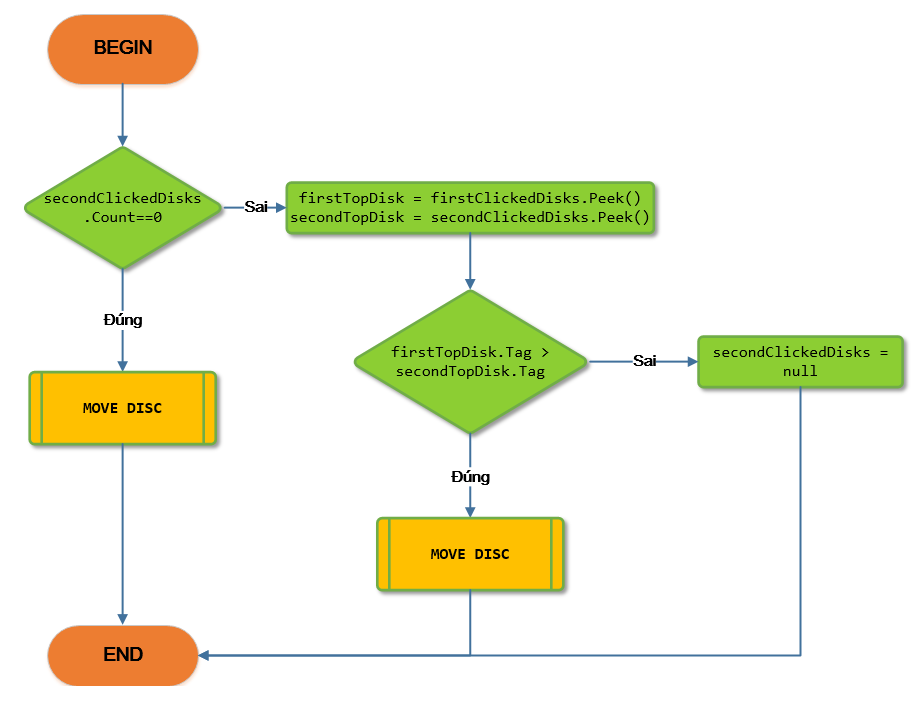


*Hình : Lưu đồ di chuyển đĩa về vị trị ban đầu (bỏ chọn)*

1. Lưu đồ con “PROCESS MOVE DISC”

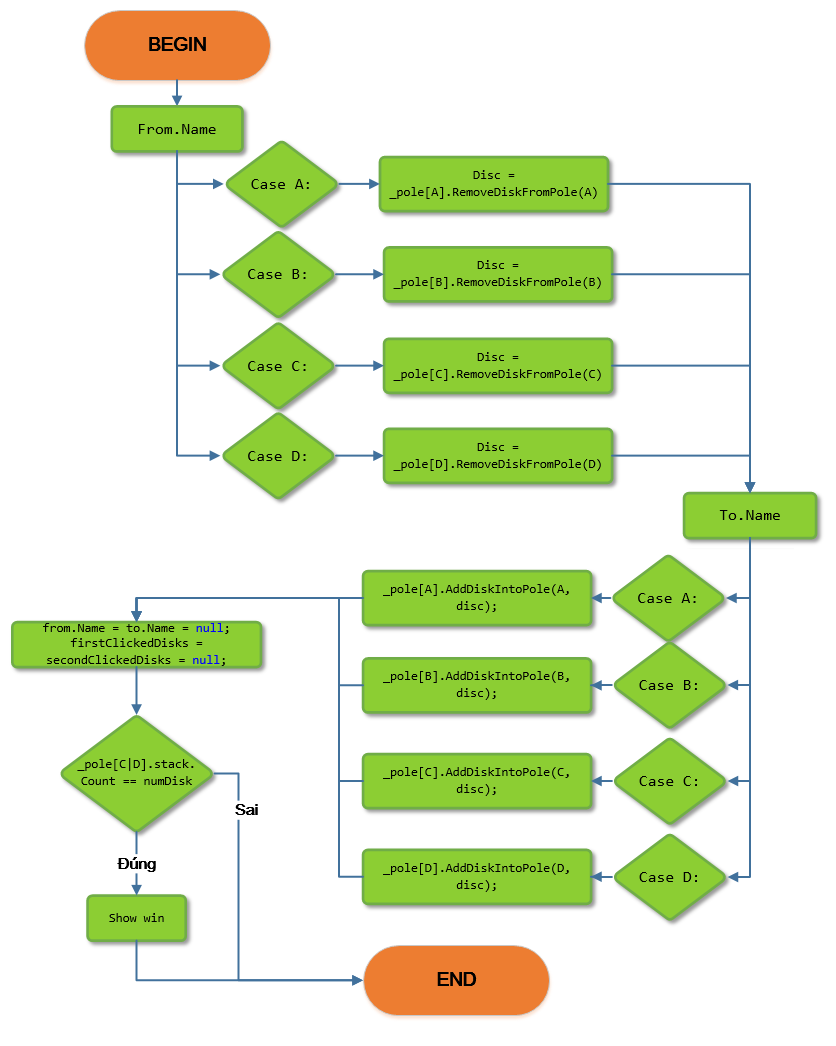
Trong phần xử lý lưu đồ tiến trình con này, mục đích để kiểm tra các trường hợp di chuyển đĩa tại lần chọn thứ 2, cụ thể:

* Nếu lần chọn thứ 2 chọn vào cọc không có đĩa (null) thì không cần kiểm tra đĩa nhỏ có nằm trên đĩa lớn hay không mà di chuyển đĩa từ cọc nguồn sang cọc đích.
* Ngược lại nếu tồn tại đĩa trên cọc chọn thứ 2 thì kiểm tra nếu thỏa quy tắc đĩa nhỏ nằm trên đĩa lớn thì tiến hành chuyển đĩa. Nếu không hợp lệ thì gán gán lại giá trị chọn cọc thứ 2 là null (chưa chọn cọc thứ 2 có thể chọn lại cọc thứ 2)

**

*Hình : Lưu đồ di kiểm tra di chuyển tiến trình di chuyển*

1. Lưu đồ con “MOVE DISC”

**

*Hình : Lưu đồ di chuyển đĩa từ nguồn tới đích*

#### **3.1.7.5 GIẢI THUẬT THÀNH TÍCH CÁ NHÂN**

#### **3.1.7.6 GIẢI THUẬT THÀNH TÍCH TRỰC TUYẾN**

#### **3.1.7.7 KIỂM TRA TRẠNG THÁI THẮNG CUỘC**

## 3.2 THIẾT KẾ PHẦN MỀM

### **3.2.1 Giới thiệu**

#### **3.2.1.1 Mục đích**

Tài liệu thiết kế phần mềm này mô tả thiết kế kiến ​​trúc và thiết kế hệ thống cho trò chơi Chinh phục tháp Hà nội.

Tài liệu được thiết kế để thông báo cho các bên liên quan về các chi tiết của thiết kế và quá trình thiết kế.

Tài liệu cung cấp mô tả đầy đủ về thiết kế của hệ thống phần mềm từ đó giúp nhà phát triển có cái nhìn khái quát đến chi tiết về ứng dụng sẽ được xây dựng.

#### **3.2.1.2 Phạm vi**

Mục đích chính của dự án này là phát triển một trò chơi dựa trên cơ sở thuật toán đệ quy, cụ thể là bài toán tháp Hà nội.

Với việc phát triển ứng dụng trên thiết bị di dộng, đặc biệt là trên nền tảng Windows Phone sẽ đem lại sự hứng thú và mới mẽ. Hơn nữa, việc kết hợp chơi trực tuyến sẽ mang trò chơi trở nên hấp dẫn và sẽ có nhiểu người chơi biết đến.

Để thực hiện những mục tiêu của dự án cũng sẽ bao gồm một máy dùng cho việc xử lý các yêu cầu dữ liệu, và có giao diện để người dùng để truy cập trên internet ở bất kì mọi nơi để chơi trực tuyến ứng dụng trên thiết bị của mình.

#### **3.2.1.3 Tổng quan về tài liệu**

Tài liệu này gồm có 5 phần cơ bản: Giới thiệu, Tổng quan hệ thống, Kiến trúc hệ thống, Thiết kế dữ liệu và Thiết kế theo chức năng

* Nhóm lập trình viên nên đọc phần thiết kế theo chức năng.
* Nhóm thiết kế viên chỉ cần đọc phần kiến trúc hệ thống và thiết kế dữ liệu.
* Nhóm kiểm thử viên đọc phần Thiết kế theo chức năng
* Nhóm người quản lí cần đọc Kiến trúc hệ thống, Thiết kế dữ liệu, Thiết kế theo chức năng.

### **3.2.2 Tổng quan hệ thống**

Các thiết bị di dộng ngày càng được phát triển mạnh mẽ đã thúc đẩy sự phát triển các trò chơi trên nền tảng này. Đặc biêt, có sự phát triển mạnh mẽ của hệ điều hành Windows Phone. Các trò chơi đa phần mang lĩnh vực về giải trí. Trò chơi chinh phục tháp Hà nội này được phát triển không những mang lại sự giải trí cao cho người chơi mà còn đem lại những bài học bổ ích về những bài toán đã được giải quyết trên thực tiễn.

### **3.2.3 Kiến trúc hệ thống**

#### **3.2.3.1 Thiết kế kiến trúc**

Ứng dụng được thiết kế theo mô hình sau:



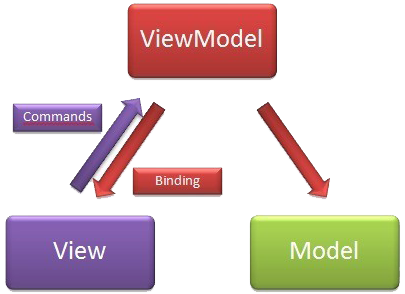
Các đặc điểm của WCF, mỗi phương pháp có ưu khuyết điểm riêng của nó. Điều đặc biệt về WCF là luôn có hơn một cách để giải quyết vấn đề trong WCF, và ta không nhất thiết phải chọn duy nhất một phương pháp nào. Có 3 phương pháp được sử dụng khi phát triển dịch vụ WCF như sau:

* Phương pháp khai báo
* Phương pháp lập trình trực tiếp
* Phương pháp sử dụng tập tin cấu hình.

Lập trình khai báo được thông qua các thuộc tính. Những thuộc tính này được sử dụng để định nghĩa các contract và xác định hành xử của dịch vụ. Chúng được sử dụng để xác định thêm các tham số để thay đổi các chi tiết của contract và hành xử dịch vụ.

Thuộc tính ServiceContract dùng để quy định là giao diện này định nghĩa các chức năng của một dịch vụ. Thuộc tính OperationContract được sử dụng ở các hàm để quy định rằng hàm này được khai báo là một phần của dịch vụ. Đó là tất cả những gì cần để tạo ra một dịch vụ WCF.

Thêm vào đó, ta không nhất thiết phải sử dụng các giao diện (interface) khi cài đặt một dịch vụ, điều này cũng giống như việc ta không cần sử dụng giao diện để định nghĩa một lớp. Tuy vậy ta nhất thiết phải chỉ rõ phần nào thuộc về dịch vụ. Ta có thể định nghĩa những phần khác cần cho giao diện, nhưng chỉ những phương thức có gắn thuộc tính [OperationContract].

**

MVVM là từ viết tắt của Model-View-ViewModel Desgin parttern. Là mô hình sử dụng khá phổ biến hiện nay, từ những trò chơi Windows Phone cho đến những trò chơi web, Windows 8. MVVM giúp cho việc cập nhật, binding dữ liệu được xuyên suốt trong quá trình tương tác, ta có thể hiểu đơn giản là với kiến trúc này chúng ta chỉ việc tương tác với dữ liệu thì phía giao diện sẽ tự động thay đổi. Với kiến trúc này sẽ tách biệt giữa kiến trúc dữ liệu, quá trình xử lý và giao diện trò chơi.

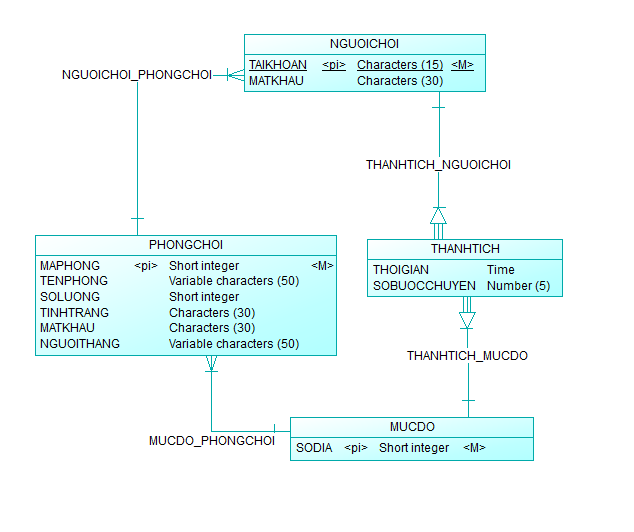
* **Model**: Chứa kiến trúc dữ liệu của trò chơi, trong Model chúng ta sẽ định nghĩa những lớp cấu tạo nên trò chơi, những cấu trúc dữ liệu cần thiết.
* **ViewModel**: Chứa dữ liệu để đổ ra ngoài giao diện, ví dụ như “danh sách các phòng chơi hiện đang có trên máy chủ” chẳng hạn. Ngoài ra các thao tác trên dữ liệu như thêm, sửa, xóa, cập nhật … cũng được xử lý trong ViewModel.
* **View**: Giao diện trò chơi

#### **3.2.3.2 MÔ TẢ SỰ PHÂN RÃ**

### **3.2.4 THIẾT KẾ DỮ LIỆU**

#### **3.2.4.1 MÔ TẢ DỮ LIỆU**

Mô hình CDM:



#### **3.2.4.2 TỪ ĐIỂN DỮ LIỆU**

BẢNG MỨC ĐỘ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên** | **Kiểu** | **Kích thước** | **Khóa chính** | **Duy nhất** | **NOT NULL** | **Khóa Ngoại** | **Diễn giải** |
| 1 | SODIA | INTERGER | 6 | x | x | x |  | Số đĩa |

BẢNG PHÒNG CHƠI

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên** | **Kiểu** | **Kích thước** | **Khóa chính** | **Duy nhất** | **NOT NULL** | **Khóa Ngoại** | **Diễn giải** |
| 1 | MAPHONG | INTERGER | 1 | x | x | x |  | Mã phòng |
| 2 | SOLUONG | INTERGER | 1 |  |  |  |  | Số lượng |
| 3 | TINHTRANG | INTERGER | 1 |  |  |  |  | Tình trạng |
| 4 | MATKHAU | VARCHAR | 32 |  |  |  |  | Mật khẩu |
| 5 | MUCDO | INTERGER | 1 |  |  |  | x | Mức độ |

BẢNG NGƯỜI CHƠI

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên** | **Kiểu** | **Kích thước** | **Khóa chính** | **Duy nhất** | **NOT NULL** | **Khóa Ngoại** | **Diễn giải** |
| 1 | TAIKHOAN | VARCHAR | 40 | X | X | X |  | Tên đăng nhập |
| 2 | MATKHAU | VARCHAR | 32 |  |  |  |  | Mật khẩu |

BẢNG THÀNH TÍCH

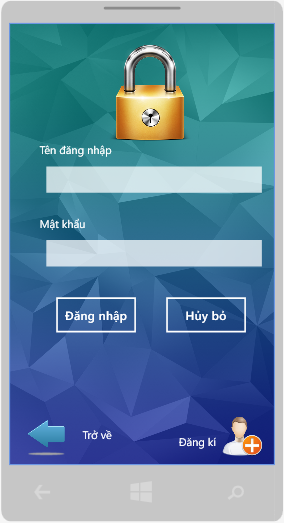
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Tên** | **Kiểu** | **Kích thước** | **Khóa chính** | **Duy nhất** | **NOT NULL** | **Khóa Ngoại** | **Diễn giải** |
| 1 | SOBUOCCHUYEN | INTERGER | 2 |  |  |  |  | Số bước chuyển |
| 2 | THOIGIAN | TIME |  |  |  |  |  | Thời gian |
| 3 | SODIA | INTERGER | 2 | X |  |  | X | Số dĩa |
| 4 | TAIKHOAN | VARCHAR | 40 |  |  |  |  | Tài khoản |

### **3.2.5 THIẾT KẾ THEO CHỨC NĂNG**

#### **3.2.5.1 CHỨC NĂNG ĐĂNG NHẬP**

**- Mục đích**: cho phép người chơi thách đấu trực tuyến có thể vào phòng chơi.

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | TextBox | txtUsername | Null |  |
| 2 | TextBox | txtPassword | Null |  |
| 3 | Button | btnDangnhap | Đăng nhập |  |
| 4 | Button | btnHuybo | Hủy bỏ |  |
| 5 | Button | btnBack | Trở về |  |
| 6 | Button | btnDangki | Đăng kí |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | NGUOICHOI | x |  |  |  |

**- Cách xử lý:**

Bước 1: Người chơi chọn thách đấu trực tuyến, tiếp theo hệ thông sẽ hiện thị trang dăng nhập.

Bước 2: Người chơi nhập vào thông tin bao gồm: tên đăng nhập và mật khẩu.

Bước 3: Hệ thống kiểm tra tên đăng nhập và mật khẩu nếu hợp lệ thì điều hướng sang trang danh sách phòng chơi để người chơi lựa chọn. Ngược lại thông báo cho người chơi nếu nhập sai tên đăng nhập hoặc mật khẩu.

**- Hàm/sự kiện**:

Sự kiện Tap chọn trên nút đăng kí

Sự kiện hiển thị bàn phím nhập kí tự cho Textbox

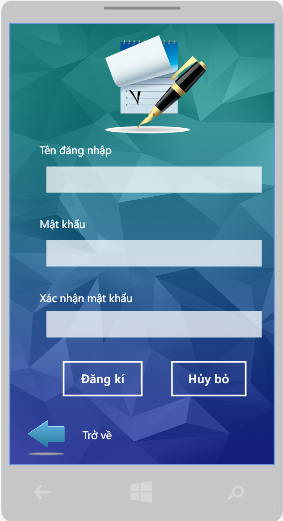
**- Các ràng buộc**:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Điều Khiển** | **Ràng Buộc** |
| 1 | username | Username không được để trống |
| 2 | password | Password không được để trống |

#### **3.2.5.2 CHỨC NĂNG ĐĂNG KÍ**

**- Mục đích**: cho phép người chơi đăng kí để sử dụng đăng nhập chơi thách đấu trực tuyến

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | TextBox | txtUsername | Null |  |
| 2 | TextBox | txtPassword | Null |  |
| 3 | Button | txtRepassword | Null |  |
| 4 | Button | btnDangki | Đăng kí |  |
| 5 | Button | btnHuybo | Hủy bỏ |  |
| 6 | Button | btnBack | Trở về |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | NGUOICHOI |  | x |  |  |

**- Cách xử lý:**

Bước 1: Người chơi chọn đăng kí mới một tài khoản, trò chơi sẽ gọi trang đăng kí.

Bước 2: Người chơi nhập thông tin tài khoản, mật khẩu, xác nhận mật khẩu.

Bước 3: Người chơi chọn đăng kí

Bước 4: Kiểm tra cú pháp hợp lệ của tài khoản và mật khẩu. Nếu không hợp lệ, trò chơi ra thông báo yêu cầu nhập lại thông tin, ngược lại, chuyển sang bước kế tiếp.

Bước 5: Trò chơi kiểm tra sự tồn tại của tài khoản trong CSDL. Nếu đã tồn tại tài khoản này, đưa ra thông báo người chơi không thể đăng kí tài khoản này, yêu cầu đổi tài khoản khác, ngược lại khởi tạo tài khoản.

Bước 6: Thông báo khởi tạo tài khoản thành công, chuyển đến trang đăng nhập.

**- Hàm/sự kiện**:

Sự kiện chọn nút lệnh đăng kí, trò chơi sẽ thực hiện kiểm tra thông tin hợp lệ của tài khoản, mật khẩu.

Sự kiện xác nhận khởi tạo tài khoản thành công và chuyển về trang đăng nhập

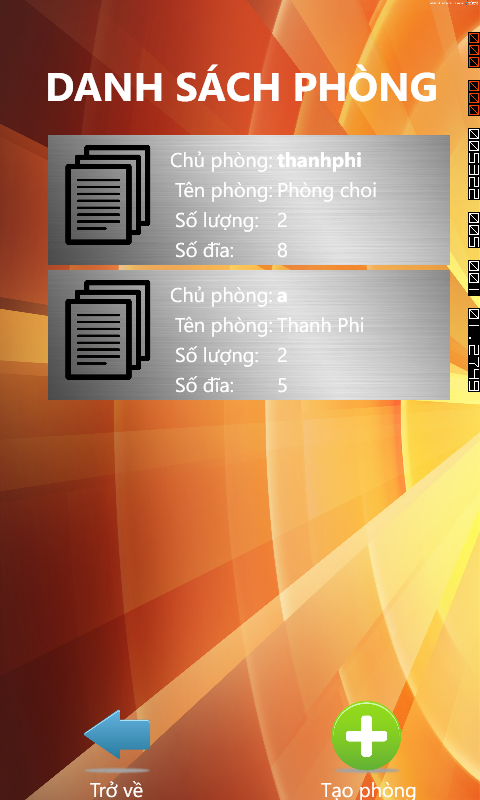
**- Các ràng buộc**:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Điều Khiển** | **Ràng Buộc** |
| 1 | username | Username không được để trống |
| 2 | password | Password không được để trống |
| 3 | repassword | Repassword không được để trống và phải giống với Password đã nhập trước đó. |

**3.2.5.3 CHỨC NĂNG HIỂN THỊ DANH SÁCH PHÒNG CHƠI**

**- Mục đích**: hiển thị danh sách để người chơi chọn phòng chơi mà mình muốn tham gia

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | WrapPanel | Wrap |  | Chứa thông tin 1 phòng chơi |
| 2 | TextBlock | Text |  | Chứa thông tin: chủ phòng, tên phòng, số lượng |
| 3 | Image | imgPhongchoi |  | Hiển thị hình ảnh: logo phòng, phòng bảo mật |
| 4 | Image | ImgTaophong |  | Hiển thị button để người chơi có thể tạo được phòng mới |
| 5 | Image | imgBack |  | Điều hướng về trang trước đó |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | PHONGCHOI | x |  |  |  |

**- Cách xử lý:**

Bước 1: sau khi đăng nhập thành công, người chơi được đưa đến giao diện danh sách phòng chơi, nếu muốn tham gia vào phòng nào thì chạm chọn phòng chơi đó.

Bước 2: Ứng dụng kiểm tra nếu số lượng người chơi còn là 1 và hiển thị mật khẩu phòng (nếu phòng có thiết đặt mật khẩu) để người chơi nhập vao, các yêu cầu này được gửi về máy chủ, máy chủ xử lí, truy xuất thông tin rồi gửi kết quả trở về máy của người chơi.

Bước 3: Thông báo phòng đầy nếu đủ số lượng người chơi. Ngược lại cho phép người chơi mới vào phòng.

**- Hàm/sự kiện**:

Sự kiện kiểm tra phòng có bảo mật

Sự kiện kiểm tra phòng đã đủ số lượng người chơi

**- Các ràng buộc**:

**3.2.5.4 Chức năng Tạo phòng mới**

**- Mục đích**: cho phép người chơi tạo phòng chơi mới.

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | TextBox | txtChuphong | Tên chủ phòng với tài khoản đăng nhập | Tên chủ phòng không được phép chỉnh sửa |
| 2 | TextBox | txtSoluong | 2 | Không được chỉnh sửa |
| 3 | PasswordBox | txtPassword | Null |  |
| 4 | ListPicker | listSodia | 3 | Hiển thị danh sách từ 3 đến 10 dĩa cho người chơi lựa chọn |
| 5 | Image | imgOk | Null | Lưu lại thông tin phòng đã tạo |
| 6 | Image | imgBack | Null | Điều hướng về trang trước đó |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | PHONGCHOI | x | x |  |  |
| 2 | MUCDO | x |  |  |  |

**- Cách xử lý:**

Bước 1: Sau khi chạm chọn tạo phòng, phòng chơi sẽ được khởi tạo với các thông tin mặc định: tên phòng, chủ phòng là tên tài khoản người chơi, mật khẩu của phòng không có (phòng chơi công cộng), số lượng người chơi mặc định là 2 người, số dĩa là 3 đến 10 dĩa.

Bước 2: Người chơi chọn vào nút Tạo thì ứng dụng lưu lại thông tin phòng và điều hướng người chơi vào phòng. Hiển thị lên danh sách phòng chơi.

**- Hàm/sự kiện**:

Sự kiện tạo mới phòng với các thông tin mặc định

Sự kiện chạm chọn các chức năng: trở về, thoát khỏi phòng chơi.

**- Các ràng buộc**:

**3.2.5.5 Chức năng Lựa chọn cọc**

**- Mục đích**: hiển thị giao diện cho người chơi chọn số lượng cọc để chơi

**- Giao diện**:



**3.2.5.6 Chức năng Chơi game**

**- Mục đích**: hiển thị giao diện cho người chơi di chuyển đĩa theo ý muốn

**- Giao diện**:



Giao diện sau khi chọn “Bắt đầu”



Giao diện khi chiến thắng:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Canvas | cavasRodA | Null | Lưu các đĩa ở cọc A |
| 2 | Canvas | cavasRodB | Null | Lưu các đĩa ở cọc B |
| 3 | Canvas | cavasRodC | Null | Lưu các đĩa ở cọc C |
| 4 | TextBlock | txtThoigian | Thời gian 00:00:00 | Cập nhật thời gian khi người chơi nhấp bắt đầu |
| 5 | TextBlock | txtSodia | Số lần chuyển: 0 | Mỗi lần di chuyển đúng thì số lần chuyển sẽ tăng lên 1 đơn vị |
| 6 | Button | btnStart | Bắt đầu | Bắt đầu đếm thời gian |
| 7 | Button | btnHuongdan | Hướng dẫn | Điều hướng đến trang hướng dẫn |
| 8 | ListPicker | listPickerSodia | 3 | Hiển thị danh sách dĩa được chọn từ 3 đến 10 dĩa |
| 9 | Canvas | Thongbao | Visibility | Hiển thị khi người chơi thắng cuộc |
| 10 | Button | Next | Số lượng dĩa + 1 | Cho phép người chơi chơi ở mức độ tiếp theo |
| 11 | Image | imgBack |  | Trở về trang giao diện chính |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | MUCDO | x |  |  |  |
| 2 | THANHTICH | x | x | x |  |

**- Cách xử lý:**

Bước 1: Giao diện hiển thị số lượng dĩa để người chơi lựa chọn.

Bước 2: Người chơi nhấp chọn “Bắt đầu” để tiến hành vào trò chơi

Bước 3: Người chơi “Chọn” dĩa cần di chuyển từ cọc bất kì sang cọc mong muốn. Ứng dụng sẽ kiểm tra sự kiện di chuyển dĩa có hợp lý không. Sự kiện hợp lý là mỗi lần chuyển được 1 dĩa và dĩa nhỏ phải nằm trên dĩa lớn hơn.

Bước 4: Cập nhập số lần di chuyển đúng và kiểm tra xem trạng thái thắng cuộc của người chơi gồm số dĩa cọc C phải đúng bằng với số lượng dĩa đã chọn trước đó.

Bước 5: Hiển thị giao diện thắng cuộc cho người chơi và cập nhật thành tích nếu thời gian di chuyển ngắn hơn thời gian lưu trữ trong dữ liệu. Nếu cấp độ đó chưa từng chơi thì thêm mới vào.

**- Hàm/sự kiện**:

Sự kiện chạm trên cọc bao gồm cọc A, cọc B, cọc C.

Sự kiện chuyển dĩa hợp lệ và không hợp lệ

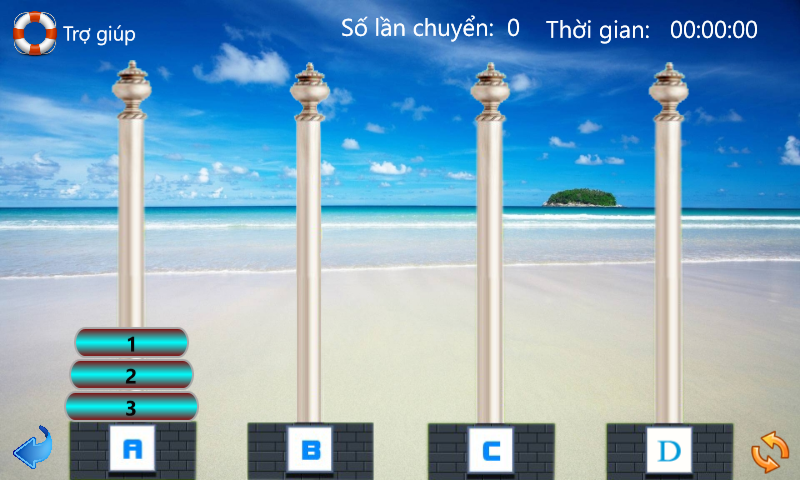
**- Các ràng buộc**:



Sau khi chọn bắt đầu giao diện trò chơi sẽ hiển thị:



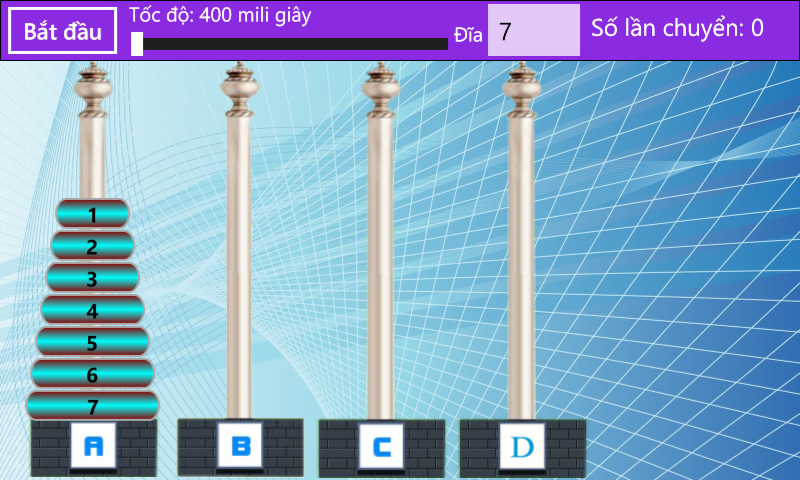
Ngoài ra, người chơi có thể chọn chơi lại bằng cách nhấn vào biểu tượng bên phải góc màn hình:



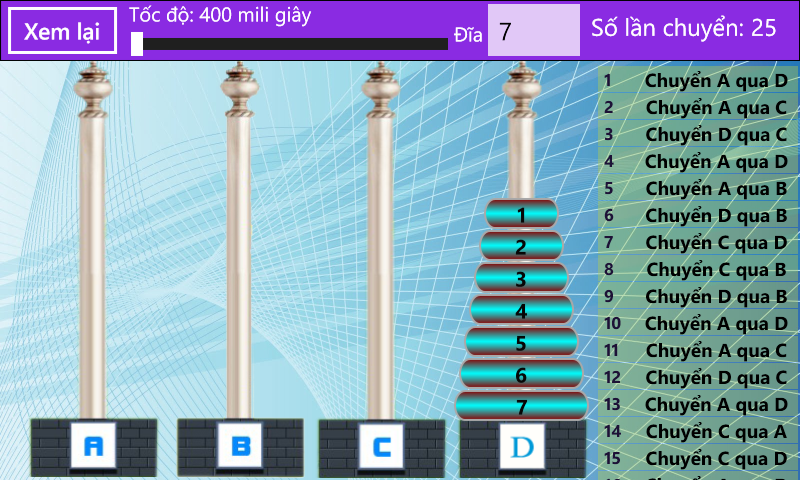
**3.2.5.7 Chức năng Trợ giúp**

**- Mục đích**: sẽ hiển thị cách thức di chuyển dĩa đã được tính toán trước đó cho người chơi có thể tham khảo và tiến hành chơi lại.

**- Giao diện**:



Giao diện khi đăng xử lý di chuyển với 4 cọc:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Canvas | cavasRodA | Null | Lưu các đĩa ở cọc A |
| 2 | Canvas | cavasRodB | Null | Lưu các đĩa ở cọc B |
| 3 | Canvas | cavasRodC | Null | Lưu các đĩa ở cọc C |
| 4 | TextBlock | txtThoigian | Thời gian 00:00:00 | Cập nhật thời gian khi người chơi nhấp bắt đầu |
| 5 | TextBlock | txtSodia | Số lần chuyển: 0 | Mỗi lần di chuyển đúng thì số lần chuyển sẽ tăng lên 1 đơn vị |
| 6 | Button | btnStart | Bắt đầu | Bắt đầu đếm thời gian |
| 7 | Button | btnHuongdan | Hướng dẫn | Điều hướng đến trang hướng dẫn |
| 8 | ListPicker | listPickerSodia | 3 | Hiển thị danh sách dĩa được chọn từ 3 đến 10 dĩa |
| 9 | Canvas | Thongbao | Visibility | Hiển thị khi người chơi thắng cuộc |
| 10 | Button | Next | Số lượng dĩa + 1 | Cho phép người chơi chơi ở mức độ tiếp theo |
| 11 | Image | imgBack |  | Trở về trang giao diện chính |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | MUCDO | x |  |  |  |

**- Cách xử lý:**

Bước 1: Người chơi chọn số lượng dĩa cần trợ giúp

Bước 2: Người chơi chọn bắt đầu và hệ thống sẽ tự chuyển tất cả các dĩa từ cọc A sang cọc C.

Bước 3: Thông báo hoàn tất sau khi đã chuyển.

**- Hàm/sự kiện**:

Hàm tính toán giải thuật đệ quy để giải quyết bài toán Tháp Hà Nội.

Hàm tính thời gian di chuyển để quyết định tốc độ di chuyển.

**- Các ràng buộc**:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Điều Khiển** | **Ràng Buộc** |
|  |  |  |

**3.2.5.8 Chức năng Hướng dẫn**

**- Mục đích**: hướng dẫn cách thức chơi cung như quy luật về trò chơi bao gồm các phần Chơi đơn, Thách đấu trực tuyến

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Image | imgMain | Null |  |
| 2 | Image | imgChoidon | Null |  |
| 3 | Image | imgThachdau | Null |  |
| 4 | Image | imgBackground | Null |  |
| 5 | Image | imgFullBackground | Null |  |
| 6 | TextBlock | txtChoidon | Chơi đơn |  |
| 7 | TextBlock | txtThachdau | Thách đấu |  |
| 8 | TextBlock | txtHuongdan | Hướng dẫn |  |

**- Dữ liệu được sử dụng**:

Không có

**- Cách xử lý:**

Người chơi di cần di chuyền chạm và giữ màn hình kéo lên hoặc xuống để có thể xem tất cả thông tin hướng dẫn cách chơi và giới thiệ về ứng dụng.

* 1. Chơi đơn
  2. Thách đấu trực tuyến
  3. Bảng xếp hạng thành tích

**- Hàm/sự kiện**:

Sự kiện chạm và kéo để lướt xem chi tiết

**- Các ràng buộc**:

**3.2.5.9 Chức năng Bật/tắt âm thanh**

**- Mục đích**: cho người chơi tùy chọn âm thanh

**- Giao diện**

**- Các thành phần trong giao diện**:

**- Dữ liệu được sử dụng**:

**- Cách xử lý:**

Người chơi chọn vào hình ảnh có biểu tượng âm thanh. Nếu âm thanh đang bật thì sẽ tắt và ngược lại sẽ bật lại âm thanh.

**- Hàm/sự kiện**:

Sự kiện chạm vào hình ảnh

**- Các ràng buộc**:

**3.2.5.10 Chức năng Bảng xếp hạng cá nhân**

**- Mục đích**: hiển thị thành tích cá nhân của người chơi đơn với các mức độ khác nhau tương ứng với số dĩa đã chơi

**- Giao diện**:



*Hình : Giao diện thành tích cá nhân với số cọc là 3*

Mặc định giá trị được lấy từ CSDL với số đĩa từ 3 đĩa đến 10 đĩa. Số bước được mặc đinh là 0, thời gian được mặc định là --:--:-- và ngày lập mặc định là --/--/---

*Hình : Giao diện thành tích cá nhân với số cọc là 4*

**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Image | imgXoathanhtich | Null |  |
| 2 | ListBox | listThanhtich | Null | Giá trị của listbox sẽ phụ thuộc từ CSDL, nếu ở mức độ nào đó người chơi chưa thực hiện thì hiện thị là 0 lần chuyển và thời gian 00:00:00 |
| 3 |  |  |  |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | THANHTICH | x |  |  |  |

**- Cách xử lý:**

Bước 1: Người chơi chọn vào thành tích cá nhân

Bước 2: Ứng dụng sẽ hiển thị các thành tích mới nhất của người chơi. Nội dung hiển thị bao gồm mức độ (số dĩa), thời gian chuyển, số bước chuyển, ngày lập kỉ lục. Nếu thành tích mới lập thấp hơn thành tích đã được lưu trong CSDL thì sẽ không cập nhật thành tích.

**- Hàm/sự kiện**:

Sự kiện làm xóa bảng thành tích, sự kiện này sẽ thiết đặt các giá trị ủa bảng thành tích cá nhân về giá trị mặc định là 0 với mỗi cấp độ từ 3 đến 10

**- Các ràng buộc**:

**3.2.5.11 Chức năng Thành tích thách đấu trực tuyến**

**- Mục đích**: cho phép người chơi xem thành tích thách đấu trực tuyến của mình ở mỗi phòng và cấp độ khác nhau

**- Giao diện**:

**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 |  | x |  |  |  |

**- Cách xử lý:**

**- Hàm/sự kiện**:

**- Các ràng buộc**:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Điều Khiển** | **Ràng Buộc** |
|  |  |  |

**3.2.5.12 Chức năng Giao diện chính**

**- Mục đích**: hiển thị giao diện chính cho phép người chơi bắt đầu lựa chọn các tính năng của ứng dụng

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Image | imgThachdautructuyen |  | Điều hướng đến trang đăng nhập |
| 2 | Image | imgChoidon |  | Điều hướng đến trang chơi đơn |
| 3 | Image | imgThanhtich |  | Điều hướng đến thành tích |
| 4 | Image | imgGioithieu |  | Điều hướng đến trang giới thiệu |
| 5 | Image | imgAmthanh |  | Bật/tắt âm thành |
| 6 | Image | imgCaidat |  | Hiển thị cài đặt về cường độ âm thanh |

**- Dữ liệu được sử dụng**:

**- Cách xử lý:**

Bước 1: Người chơi chọn vào chức năng xem thành tích thách đấu trực tuyến

Bước 2: Hệ thống truy vấn cơ sở dữ liệu và hiển thị danh sách các phòng mà người chơi đã thắng trước đó.

**- Hàm/sự kiện**:

**- Các ràng buộc**:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Điều Khiển** | **Ràng Buộc** |
|  |  |  |

**3.2.5.13 Chức năng Thoát khỏi phòng chơi**

**- Mục đích**: cho phép người chơi tùy chọn rời phòng chơi.

**- Giao diện**:

**- Các thành phần trong giao diện**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Loại Điều Khiển | Tên Điều Khiển | Giá Trị Mặc Định | Lưu ý |
| 1 | Canvan | canvasThongbao | Bạn có chắc chắn muốn rời khỏi phòng chơi hay không |  |
| 2 | Image | imgOK | Xác nhận |  |
| 3 | Image | imgHuybo | Hủy bỏ |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | TÊN BẢNG | Phương thức | | | |
| Truy vấn | Thêm | Sửa | Xoá |
| 1 | PHONG | x |  |  | x |

**- Cách xử lý:**

Bước 1: Người chơi chọn vào chức năng rời khỏi phòng chơi

Bước 2: Nếu đồng ý thoát thì chọn đồng ý, ngược lại chọn hủy

Bước 3: Khi chủ phòng thoát khỏi phòng chơi thì xóa phòng chơi và hiển thị lại giao diện danh sách phòng chơi.

**- Hàm/sự kiện**:

Sự kiện chạm vào hình ảnh để rời khỏi phòng chơi

**- Các ràng buộc**:

## 3.3 KIỂM THỬ PHẦN MỀM

### Giới thiệu kế hoạch kiểm thử

#### Mục tiêu

Mục đích chính của kiểm thử là phát hiện ra các lỗi phần mềm để từ đó khắc phục và sửa chữa. Việc kiểm thử không thể khẳng định được rằng các chức năng của sản phẩm đúng trong mọi điều kiện, mà chỉ có thể khẳng định rằng nó không hoạt động đúng trong những điều kiện cụ thể.

Tài liệu kế hoạch kiểm thử được dùng để:

* Xác định các thành phần dự án cần được kiểm thử.
* Liệt kê những yêu cầu kiểm thử
* Định nghĩa những phương pháp, chiến lược kiểm thử nên sử dụng trong quá trình kiểm thử
* Tổng hợp, đánh giá, báo cáo những kết quả, tài liệu có được sau khi thực hiện kiểm thử

#### Phạm vi

Phạm vi của kiểm thử phần mềm thường bao gồm việc kiểm tra mã, thực hiện các mã trong môi trường và điều kiện khác nhau, và việc kiểm thử các khía cạnh của mã: nó có làm đúng nhiệm vụ của nó hay không, và nó có làm những gì cần phải làm hay không. Trong môi trường phát triển phần mềm hiện nay, một đội kiểm thử có thể tách biệt với đội phát triển. Các thành viên trong đội kiểm thử giữ các vai trò khác nhau. Các thông tin thu được từ kiểm thử có thể được sử dụng để điều chỉnh quá trình phát triển phần mềm.

Quá trình kiểm thử sẽ được thực hiện qua các công đoạn:

* Kiểm thử thiết kế: tiến hành kiểm thử bản thiết kế có khớp với đặc tả yêu cầu.
* Kiểm thử cài đặt: kiểm thử hộp trắng, tìm và sửa lỗi, kiểm thử đơn vị.
* Kiểm thử chức năng: kiểm thử hộp đen, kiểm tra các chức năng có xử lí đúng dữ liệu.
* Kiểm thử chấp nhận: kiểm tra các chức năng của chương trình có đáp ứng đặc tả yêu cầu.
* Các loại kiểm thử sẽ sử dụng:
  + Kiểm thử đơn vị.
  + Kiểm thử chức năng.
  + Kiểm thử cài đặt.
  + Kiểm thử tích hợp.

### Chi tiết kế hoạch kiểm thử

#### Các tính năng sẽ được kiểm thử

#### Các tính năng sẽ không được kiểm thử

#### Cách tiếp cận

Kiểm thử chức năng của phần mềm:

* Phân tích mã nguồn với công cụ Visual Studio 2012 Professional Server Pack 1, kiểm tra các lớp, phương thức, viết testcase, tài liệu kết quả.
* Kiểm thử chức năng để kiểm tra thực thi của các chức năng trên nền tảng hệ điều hành Windows Phone 8.0, viết testcase và viết tài liệu kết quả.

Kiểm thử giao diện:

* Phân tích bằng thiết kế giao diện, nguyên tắc hoạt động, thử nghiệm giao diện thực tế, viết tài liệu kết quả.
* Sử dụng thời gian lâu, thử các điều khiển trên giao diện để tìm lỗi, viết testcase, tài liệu kết quả.

Kiểm thử các yêu cầu phi chức năng:

* Sử dụng các chức năng, ghi nhận thời gian phản ứng, viết tài liệu kết quả.
* Kiểm tra tương tác, kiểm tra các điều kiện sử dụng chức năng chương trình, kiểm tra CSDL, viết tài liệu kết quả.
* Thử nghiệm đóng góp phần mềm, tiến hành chạy ứng dụng trong môi trường giả lập Windows Phone Emulator để đánh giá khả năng ứng dụng của phần mềm.
* Trong phần này, kỹ thuật và tiêu chuẩn đánh giá là những nội dung chính cần quan tâm.

Ngoài ra, các công cụ sau sẽ được sử dụng để tham gia vào quá trình kiểm thử hộp trắng và hộp đen:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên công cụ | Phiên bản | Nhà sản xuất |
| 1 | Visual Studio 2012 | Professional Server Pack 1 | Microsoft |
| 2 | Microsoft Expression Blend | Version 4 | Microsoft |
| 3 | Windows Phone SDK | Version 8.0 | Microsoft |
| 4 | SQL Server 2012 |  | Microsoft |

#### Tiêu chí kiểm thử thành công/thất bại

Tiêu chí kiểm thử thành công:

* Kết quả thực tế khi thực thi một chức năng khớp với kết quả mong muốn.
* Lỗi phát sinh trong quá trình kiểm thử phải nằm trong các lỗi được dự đoán trước.
* Chức năng đáp ứng yêu cầu đặc tả
* Thời gian gọi và hiển thị mỗi trang chức năng tối đa 5 giây

Tiêu chí kiểm thử thất bại:

* Quá trình kiểm thử phát sinh lỗi không nằm trong kết quả mong muốn
* Thao tác của một chức năng bị sai khác với bản thiết kế.

#### Tiêu chí đình chỉ và yêu cầu bắt đầu lại

* Trong quá trình kiểm thử một chức năng, bị thoát ra khỏi trò chơi hoặc người chơi không thao tác được với các nút lệnh.
* Với các chức năng thuộc phần chơi trực tuyến, khi phát sinh lỗi không kết nối được với máy chủ hoặc máy chủ không phản hồi quá khoảng 10 giây.

#### Sản phẩm bàn giao kiểm thử

* Tài liệu kế hoạch kiểm thử, trường hợp kiểm thử.
* Môi trường kiểm thử.
* Các đặc tả kiểm thử.

### Quản lí kiểm thử

#### Các hoạt động/công việc được lập kế hoạch; sự tiến hành kiểm thử

#### Môi trường

Chinh phục tháp Hà nội sẽ được kiểm thử trên bộ giả lập Windows Phone Emulator có thông số phần cứng giả lập như sau:

* 1. Độ phân giải màn hình 480x800
  2. Màn hình nhận diện 4 điểm tiếp xúc
  3. Camera ít nhất 5MP
  4. 512MB RAM
  5. Flash 8GB

#### Tài nguyên và sự cấp phát chúng

Tài nguyên:

* Mã nguồn chương trình
* Phần cứng: laptop giả lập thiết bị WP
* Phần mềm: WPE, VS
* Nhân lực: Lập trình viên, kiểm thử viên

#### Kế hoạch, dự đoán và chi phí

#### Các rủi ro

* Tỷ lệ hoàn tất chuẩn bị môi trường kiểm thử
* Chưa bao quát hết các trường hợp, rủi ro bởi mã kiểm thử
* Mức tin tưởng của tester vào sản phẩm
* Mốc thời gian quan trọng
* Tester thiếu đào tạo về kĩ năng kiểm thử hoặc ít chú trọng vai trò kiểm thử.
* Về mặt kĩ thuật kiểm thử: không tồn tại thuật toán tổng quát có thể chứng minh sự đúng đắn hoàn toàn của bất kỳ một chương trình nào. Do đó, kết quả kiểm thử không khẳng định chương trình không có lỗi mà chỉ đưa ra mức độ tin cậy mà người dùng có thể sử dụng chương trình.

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## 4.1 KẾT QUẢ ĐẠT ĐƯỢC

### **4.1.1 Lý thuyết**

### **4.1.2 Chương trình**

### **4.1.3 Thực tế**

## 4.2 HẠN CHẾ

## 4.3 HƯỚNG PHÁT TRIỂN