

**TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**

**BÙI ANH TÀI  
NGUYỄN KHÁNH TÀI**

**XÂY DỰNG PHẦN MỀM TRÒ CHƠI  
HỖ TRỢ HỌC LẬP TRÌNH PASCAL**

**KHÓA LUẬN TỐT NGHIỆP**

**TP. HỒ CHÍ MINH – 2012**

TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN

BÙI ANH TÀI  
NGUYỄN KHÁNH TÀI

**XÂY DỰNG PHẦN MỀM TRÒ CHƠI  
HỖ TRỢ HỌC LẬP TRÌNH PASCAL**

**CHUYÊN NGÀNH: SƯ PHẠM TIN HỌC**

**KHÓA LUẬN TỐT NGHIỆP**

**GVHD: ThS LÊ ĐỨC LONG**

## LỜI CẢM ƠN

Trong những năm tháng học tập tại trường **Đại học Sư Phạm TP.HCM** chúng em đã được trang bị nền tảng kiến thức chuyên ngành đầy đủ, những kỹ năng cần thiết để vững bước trên con đường đã chọn. Luận văn tốt nghiệp giúp chúng em tổng hợp lại một cách đầy đủ những kiến thức đã được học.

Lời đầu tiên, chúng em xin chân thành cảm ơn thầy hướng dẫn **ThS. Lê Đức Long** đã tận tình hướng dẫn, giúp đỡ chúng em thực hiện luận văn này.

Chúng em xin gửi lời cảm ơn sâu sắc đến các thầy cô trong **Khoa Công Nghệ Thông Tin – Trường Đại học Sư Phạm TP.HCM** đã dạy dỗ, trang bị cho chúng em những kiến thức quý báu.

Xin cảm ơn quý Thầy Cô phản biện đã dành thời gian quan tâm đến luận văn của chúng em.

Cuối cùng, chúng con xin gửi lời cảm ơn đến gia đình, những người thân đã luôn bên cạnh, ủng hộ và giúp đỡ chúng con trong suốt những năm qua.

Tuy có những nỗ lực và cố gắng nhất định, nhưng cũng không thể tránh khỏi sai sót và khuyết điểm trong khi thực hiện báo cáo này. Mong nhận được sự đóng góp của thầy cô và bạn bè.

**Sinh viên thực hiện:**

Bùi Anh Tài

Nguyễn Khánh Tài

TP.HCM, ngày 25 tháng 4 năm 2012

## MỤC LỤC

LỜI CẢM ƠN.....	1
BẢNG DANH MỤC CÁC HÌNH VẼ.....	5
GIỚI THIỆU ĐỀ TÀI.....	9
• Mục tiêu nghiên cứu .....	9
• Qui trình nghiên cứu:.....	9
• Công cụ nghiên cứu:.....	10
• Kết quả dự kiến của đề tài: .....	10
CHƯƠNG 1: PHƯƠNG PHÁP LUẬN VÀ CƠ SỞ LÝ LUẬN.....	12
1.1 Lý thuyết trò chơi – game theory:.....	12
1.1.1 Giới thiệu:.....	12
1.1.2 Các thành phần cơ bản của game: .....	13
1.2. Trò chơi giáo dục - Educational game:.....	15
1.2.1 Khái niệm: .....	15
1.2.2 Lợi ích của trò chơi trong giáo dục: .....	15
1.2.3 Ứng dụng game trong dạy học: .....	16
1.2.4 Một số điều cần lưu ý khi sử dụng game trong giáo dục: .....	16
1.2.5 Các đặc điểm tạo nên tính hấp dẫn trong game giáo dục: .....	17
1.2.5 Quá trình thiết kế Game Educational: .....	18
1.2.6 Các dạng game thường gặp: .....	20
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ TRÒ CHƠI .....	26
2.1. Giới thiệu trò chơi Secret Garden: .....	26
2.2 Kịch bản:.....	30
2.2 Đặc tả về yêu cầu đối với phần mềm trò chơi Secret Garden:.....	40
2.2.1 Yêu cầu chức năng: .....	40
2.2.2 Yêu cầu phi chức năng: .....	40

2.2.3	Mô hình Use Case: .....	41
2.2.4	Đặc tả một số Use Case quan trọng:.....	42
2.3	Thiết kế dữ liệu: .....	48
2.4	Thiết kế xử lý:.....	51
2.4.2	Chi tiết các lớp xử lý màn hình: .....	62
2.4.3	Chi tiết các lớp xử lý đối tượng:.....	69
2.5	Thiết kế giao diện: .....	70
2.5.1	Màn hình chính:.....	71
2.5.2	Màn hình đăng ký:.....	72
2.5.3	Màn hình đăng nhập: .....	73
2.5.4	Màn hình Stage (Màn chơi):.....	74
2.5.5	Màn hình 1 scene:.....	75
<b>CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM TRÒ CHƠI.....</b>		<b>77</b>
3.1	Môi trường phát triển:.....	77
3.2	Một số màn hình và chức năng minh họa: .....	77
3.2.1	Màn hình vào game: .....	78
3.2.3	Màn hình Login : .....	80
3.2.4	Màn hình hướng dẫn:.....	81
3.2.5	Màn hình xếp hạng người chơi:.....	82
3.2.6	Màn hình Stage:.....	83
3.2.7	Màn hình các scene: .....	84
3.2.8	Màn hình hướng dẫn ở đầu mỗi scene:.....	86
3.2.9	Màn hình khi tạm dừng – Pause: .....	87
3.2.10	Màn hình câu hỏi: .....	88
3.2.11	Màn hình xem lại kiến thức:.....	90
3.2.12	Màn hình thông báo thua: .....	92

3.2.13 Màn hình chiến thắng ở level: .....	93
3.2.14 Màn hình chiến thắng ở scene: .....	94
KẾT LUẬN .....	96
• Hạn chế và khó khăn của đề tài. ....	96
• Hướng phát triển:.....	96
TÀI LIỆU THAM KHẢO .....	97
PHỤ LỤC .....	100
• Hướng dẫn sử dụng: .....	100
• Hướng dẫn chơi game Secret Garden:.....	103
• Công nghệ xử lý: .....	104

## **BẢNG DANH MỤC CÁC HÌNH VẼ**

Hình 1.1: Game Spacewar .....	12
Hình 1.2:Game cờ vua[18] .....	14
Hình 1.3: Sơ đồ thiết kế Educational game [16] .....	19
Hình 1.4:Game Timez Attack[19] .....	20
Hình 1.5:Game Timez Attack[20] .....	21
Hình 1.6:Game Sim City[21] .....	22
Hình 1.7:Game Zombies Master[22].....	23
Hình 1.8:Game AtomMate[23] .....	24
Hình 2.1:Knowledge Graph ở các mức độ theo ngữ cảnh sử dụng.....	27
Hình 2.2: Sơ đồ ý nghĩa các thành phần của Khóa học.....	27
Hình 2.3:Cấu trúc trò chơi Secret garden .....	30
Hình 2.4: Sơ đồ kịch bản trong Stage.....	31
Hình 2.5: Sơ đồ xử lý trong level 1 .....	34
Hình 2.6:Sơ đồ xử lý trong level 2 .....	35
Hình 2.7:Sơ đồ xử lý trong level 3 .....	36
Hình 2.8: Sơ đồ Use Case.....	41
Hình 2.9:Dòng sự kiện use case Register .....	42
Hình 2.11:Dòng sự kiện use case Option .....	43
Hình 2.10:Dòng sự kiện use case Guide.....	43
Hình 2.12:Dòng sự kiện use case Login.....	44
Hình 2.13:Dòng sự kiện use case New game .....	45
Hình 2.14:Dòng sự kiện use case Load game .....	46
Hình 2.15:Dòng sự kiện use case Play game.....	47
Hình 2.16: Mô hình PDM.....	48
Hình 2.17: Sơ đồ tổng quan các lớp xử lý.....	51
Hình 2.18: Các lớp xử lý dữ liệu .....	52
Hình 2.20: Lớp Stage.....	53
Hình 2.19: Lớp PI.....	53
Hình 2.21: Lớp Question .....	54
Hình 2.22: Lớp PI_Condition.....	55
Hình 2.23: Lớp Quiz.....	55

Hình 2.24: Lớp Topic .....	56
Hình 2.25: Lớp PI_Topic.....	57
Hình 2.26: Lớp Question_Type.....	57
Hình 2.27: Lớp Scene_Question .....	57
Hình 2.28:MultiChoice.....	58
Hình 2.29: Lớp Scene_Config.....	59
Hình 2.30: Lớp Scene .....	60
Hình 2.31: Lớp User.....	61
Hình 2.32: Các lớp xử lý màn hình .....	62
Hình 2.33: Lớp QuestUC.....	63
Hình 2.34: Lớp Scene .....	65
Hình 2.35: Các lớp xử lý đối tượng.....	69
Hình 2.36: Lớp Alice (bông hoa) .....	70
Hình 2.37: Sơ đồ màn hình chính.....	70
Hình 2.38:Màn hình chính.....	71
Hình 2.39:Màn hình đăng ký.....	72
Hình 2.40:Màn hình đăng nhập .....	73
Hình 2.41:Màn hình Stage.....	74
Hình 2.42:Màn hình Scene .....	75
Hình 3.1: Sơ đồ thể hiện các màn hình.....	77
Hình 3.2:Màn hình giao diện chính .....	78
Hình 3.3:Sơ đồ tình huống màn hình chính.....	78
Hình 3.4:Màn hình Register .....	79
Hình 3.5: Sơ đồ tình huống màn hình Register .....	79
Hình 3.6:Màn hình Login .....	80
Hình 3.7:Sơ đồ tình huống màn hình Login .....	80
Hình 3.8:Màn hình hướng dẫn.....	81
Hình 3.9:Sơ đồ tình huống màn hình Hướng dẫn.....	81
Hình 3.10:Màn hình Top player .....	82
Hình 3.11:Sơ đồ tình huống màn hình Top Player.....	82
Hình 3.12:Màn hình Stage.....	83
Hình 3.13: Sơ đồ tình huống màn hình Stage.....	83



Hình 3.14:Màn hình Scene1 .....	84
Hình 3.15:Màn hình scene 2.....	84
Hình 3.16:Màn hình scene 3.....	85
Hình 3.17:Sơ đồ tình huống màn hình Scene .....	85
Hình 3.18:Màn hình hướng dẫn đầu mỗi scene.....	86
Hình 3.19:Sơ đồ tình huống màn hình Hướng dẫn đầu scene .....	86
Hình 3.20:Màn hình Pause .....	87
Hình 3.21: Sơ đồ tình huống màn hình Pause .....	87
Hình 3.22:Màn hình câu hỏi Multichoice.....	88
Hình 3.23:Màn hình câu hỏi True/False .....	88
Hình 3.24:Màn hình câu hỏi Quiz .....	89
Hình 3.25:Sơ đồ tình huống màn hình câu hỏi .....	89
Hình 3.26:Màn hình kiến thức.....	90
Hình 3.27:Sơ đồ tình huống màn hình kiến thức.....	91
Hình 3.28:Màn hình thông báo thua.....	92
Hình 3.29:Sơ đồ tình huống màn hình thông báo thua.....	92
Hình 3.40:Sơ đồ tình huống màn hình thông báo thua.....	93
Hình 3.41:Sơ đồ tình huống màn hình chiến thắng ở level .....	93
Hình 3.42:Màn hình chiến thắng ở scene .....	94
Hình 3.43:Sơ đồ tình huống màn hình chiến thắng ở scene .....	94

# **GIỚI THIỆU ĐỀ TÀI**

# GIỚI THIỆU ĐỀ TÀI

Trong giáo dục ngày nay việc đổi mới phương pháp dạy học là vấn đề đang được các nhà Giáo dục đặc biệt quan tâm. Đổi mới phương pháp dạy học là thay đổi “cách dạy – cách học” nhằm phát huy tính tích cực, chủ động, sáng tạo của học sinh trong học tập.

Hiện nay có rất nhiều phương pháp dạy học tích cực, trong đó phương pháp sử dụng trò chơi trong giáo dục là phương pháp đang được sử dụng rộng rãi. Nhiều chuyên gia tâm lí ở các nước tiên tiến trên thế giới đã nghiên cứu và kết luận rằng trò chơi có tác dụng tốt trong việc rèn tư duy logic, khả năng tập trung và phối hợp, xử lý linh hoạt các tình huống trong cuộc sống, rèn tính nguyên tắc, kỉ luật... Do đó phương pháp này giúp cho người học tiếp thu kiến thức một cách “*Nhẹ nhàng, tự nhiên và hiệu quả*”.[16]

Hiện có rất nhiều phần mềm trò chơi được thiết kế nhằm hỗ trợ cho tất cả các môn học. Đối môn tin học có một số phần mềm trò chơi hỗ trợ học lập trình như Kodu Game Lab, Alice...Nhưng riêng phần lập trình cơ bản (sử dụng ngôn ngữ Pascal) vẫn chưa có phần mềm hỗ trợ. Do đó, nhóm chúng em đã chọn đề tài “Xây dựng phần mềm trò chơi hỗ trợ học lập trình Pascal”.

## **Mục tiêu nghiên cứu**

Xây dựng phần mềm hỗ trợ học lập trình Pascal dưới dạng trò chơi tương tác người - máy với tên gọi “**SECRET GARDEN**”.

Mục tiêu cụ thể bao gồm:

- Tìm hiểu các khái niệm về lý thuyết trò chơi – game theory, trò chơi giáo dục – educational game và nguyên tắc thiết kế một phần mềm trò chơi giáo dục.
- Xây dựng phần mềm trò chơi học Pascal dựa trên ý tưởng của trò chơi Plants vs.Zoomnies của PopCap Games.

## **Qui trình nghiên cứu:**

- Giai đoạn 1: Tìm hiểu đề tài.
- Giai đoạn 2: Đọc tài liệu tham khảo, nghiên cứu cơ sở lý thuyết.
- Giai đoạn 3: Phân tích và thiết kế nội dung nghiên cứu.

- Giai đoạn 4: Nghiên cứu công nghệ và viết phần mềm.
- Giai đoạn 5: Viết báo cáo.

#### ✚ Công cụ nghiên cứu:

- Công cụ nghiên cứu cơ sở lý thuyết: Các lý thuyết về thiết kế game và một số game trong giáo dục.
- Công cụ phần cứng: Máy tính.
- Công cụ phần mềm: Microsoft Visio Studio 2010, Microsoft Expression, Photoshop, PowerDesigner, Microsoft Visio.

#### ✚ Kết quả dự kiến của đề tài:

- Phần mềm trò chơi hỗ trợ học lập trình Pascal.



Hình 0.1: Game Secret Garden

- Một báo cáo về quá trình thực hiện đề tài.

Bài báo cáo luận văn của chúng em có cấu trúc như sau:

- GIỚI THIỆU ĐỀ TÀI
- CHƯƠNG 1: PHƯƠNG PHÁP LUẬN VÀ CƠ SỞ LÝ LUẬN
- CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ TRÒ CHƠI
- CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM TRÒ CHƠI
- KẾT LUẬN
- PHỤ LỤC

CHƯƠNG 1

**PHƯƠNG PHÁP LUẬN  
VÀ  
CƠ SỞ LÝ LUẬN**

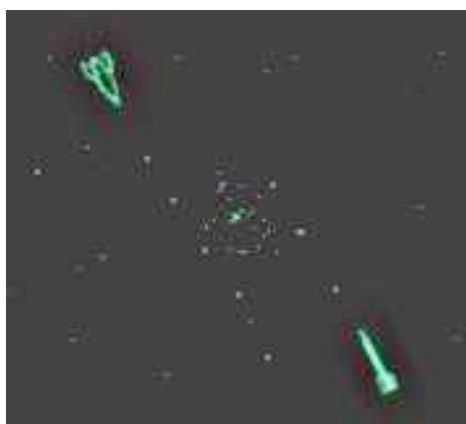
## CHƯƠNG 1: PHƯƠNG PHÁP LUẬN VÀ CƠ SỞ LÝ LUẬN.

### 1.1 Lý thuyết trò chơi – game theory:

#### 1.1.1 Giới thiệu:

Trò chơi máy tính – **PC game (gọi tắt là game)** là hình thức giải trí được ra đời từ rất lâu và nó đã xâm nhập vào đời sống con người một cách nhanh chóng. Game là một trong những thể loại của **video game** được chơi trên các máy tính cá nhân. **Video game-** được hiểu là một dạng trò chơi điện tử liên quan đến tính tương tác với một giao diện người sử dụng để tạo ra một phản hồi hình ảnh trên một thiết bị hiển thị (*video*). [13]

Nhiều game máy tính sơ khai đã chạy trên hệ thống máy chủ của các trường đại học ở nước Mỹ và được một số cá nhân lập trình trong thời gian rảnh rỗi. Tuy nhiên, do máy tính thời đó khá hạn chế nên những game loại này rất ít và dễ bị quên lãng. Vào năm 1961, một nhóm sinh viên tại Học viện công nghệ Massachusetts, trong đó có Steve Russell, đã phải mất tới 200 giờ để thiết kế trò chơi *Spacewar* trên máy tính DEC PDP-1 thuộc loại "đời mới" nhất hồi đó. Hai người tham gia sẽ đấu với nhau, mỗi người điều khiển một con tàu vũ trụ có khả năng phóng tên lửa. Một lỗ đen ở giữa sẽ tạo ra trường trọng lực lớn để gây khó khăn cho các game thủ. Trò chơi nhanh chóng được phổ biến trên các máy DEC thế hệ mới và được "bán" thông qua mạng. Được giới thiệu tại Phòng trưng bày khoa học tại Học viện công nghệ Massachusetts năm 1962, đây được đánh giá là game có ảnh hưởng và phổ biến đầu tiên trên thế giới.



Hình 1.1: Game Spacewar[17]

Sau khi **Game Spacewar** thô sơ ra đời, các nhà phát triển game nhận thấy cần phải có một lý thuyết để xây dựng, phát triển game hấp dẫn hơn và họ đã dựa vào lý thuyết về game (Game Theory) để xây dựng thế hệ game sau này.

**Game Theory** là một nhánh của toán học ứng dụng. Ngành này nghiên cứu các tình huống chiến thuật của một trò chơi, trong đó các đối thủ lựa chọn các hành động, chiến thuật khác nhau để giành được chiến thắng. Ngày nay, nhờ vào lý thuyết này game đã được phát triển rất mạnh mẽ và có rất nhiều thể loại: hành động, đối kháng, chiến thuật... với nhiều mục đích khác nhau, từ giải trí đơn thuần cho đến dạy cho người học những kỹ năng, ngôn ngữ, môn học...

### **1.1.2 Các thành phần cơ bản của game:**

Một game gồm có 4 thành phần cơ bản là: người chơi – Player, chiến lược – Strategy, thu hoạch – Payoff, luật chơi – rule.[1]

#### **a. Người chơi hay đấu thủ (player):**

- Người chơi có thể cá nhân hay một tập thể.
- Người chơi là tác nhân có thể quyết định và nhận các kết quả tương ứng với quyết định của mình.

#### **b. Chiến lược (strategy):**

- Mỗi người chơi có các kế hoạch hành động khả thi cho riêng mình, nó là một chuỗi các hành động (bước đi) sao cho khi chơi đạt được nhiều kết quả nhất hay giành được chiến thắng.

#### **c. Thu hoạch (payoff):**

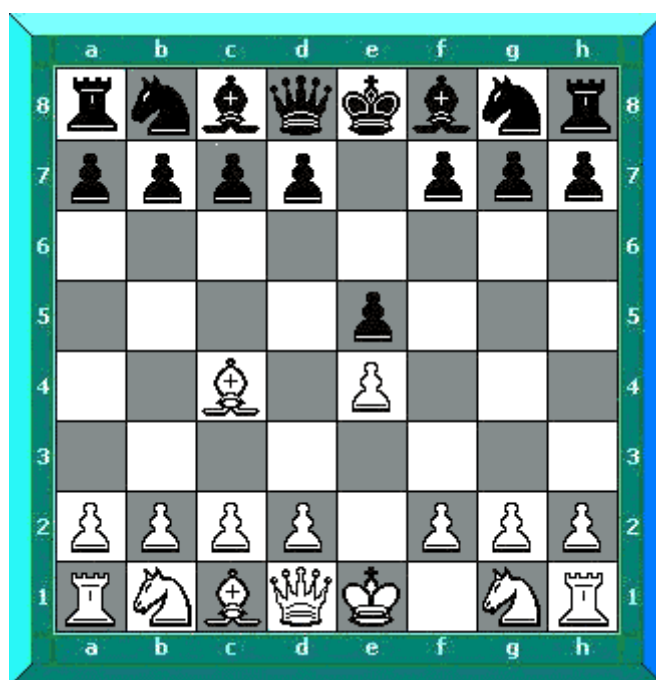
- Mỗi người chơi sẽ chọn một chiến lược cho riêng mình và khi đó cuối trò chơi mỗi người sẽ thu được một kết quả, điều đó gọi là thu hoạch.
- Người chơi được xem là chơi giỏi nếu với các chiến lược của mình đạt được thu hoạch tối đa.

#### **d. Luật chơi (rule):**

- Tập hợp tất cả quy tắc, cách thức người chơi phải hành động theo.

Các thành phần trong game Cờ vua:

<http://www.trochoivui.com/game/5936/choi-co-vua-8.html>



Hình 1.2:Game cờ vua[18]

**- Người chơi:**

+ Có thể là hai đấu thủ, hoặc giữa một đấu thủ với máy vi tính.

**-Chiến lược:**

+ Mỗi “nước đi” người chơi có thể chọn lựa trong 16 quân cờ. Từng quân cờ sẽ có quy định riêng về cách đi.

+ Người chơi sẽ phối hợp các nước đi trên từng quân cờ để loại bỏ quân cờ của đấu thủ.

**-Thu hoạch:**

+ Là kết quả đạt được sau mỗi “nước đi” của người chơi (loại bỏ được quân cờ của đối phương).

+Đến khi loại bỏ được quân cờ “Vua” của đối phương thì coi như người chơi dành được chiến thắng

**-Luật chơi:**

+ Mỗi đấu thủ sẽ lần lượt đi các quân của mình sau khi đối phương đã đi xong một nước và phải tuân thủ theo quy định riêng của từng quân cờ.



## **1.2. Trò chơi giáo dục - Educational game:**

Trò chơi từ lâu đã trở thành một phần của giáo dục. Và ngày nay với sự tiến bộ của công nghệ mới, game gần đây đã nổi lên như một công cụ giảng dạy mới. Các nhà tâm lý đã chứng minh rằng “Trò chơi có thể được thiết kế phù hợp với các mục tiêu khác nhau”. [16]

### **1.2.1 Khái niệm:**

Educational game là một hình thức game được thiết kế dành riêng cho việc học, đây là hình thức kết hợp giữa “**vừa học vừa chơi**” . [16]

Hay Educational game là sự phối hợp của nội dung giáo dục, các nguyên tắc học tập và trò chơi máy tính. [16]

Game giáo dục là chương trình được thiết kế để thúc đẩy quá trình học tập bằng các kết hợp trò chơi vào dạy học. Là công cụ học tập được sử dụng trong quá trình giảng dạy và học tập. [16]

### **1.2.2 Lợi ích của trò chơi trong giáo dục:**

Trò chơi sẽ mang lại cho học sinh sự thư giãn, giải trí và tâm lý thoải mái trong khi học. Khi tham gia trò chơi học sinh sẽ học một cách say mê, đây là điều hiếm thấy nếu chúng ta sử dụng các phương pháp dạy học truyền thống để truyền đạt cho kiến thức cho học sinh. Sau đây là một số lợi ích của game:

- Tạo động lực học tập cho học sinh thông qua các thách thức, cạnh tranh trong game.
- Trong game có thể kiểm tra được khả năng vận dụng kiến thức của học sinh.
- Tạo điều kiện củng cố và mở rộng kiến thức cho học sinh.
- Tăng cường phát triển kỹ năng và khả năng tập trung của học sinh.
- Thu hút sự tham gia của học sinh.
- Giúp học sinh tiếp nhận tiếp nhận thông tin dễ dàng hơn.

### 1.2.3 Ứng dụng game trong dạy học:

Educational game đã trở thành một phần của giáo dục trong nhiều thập kỷ qua. Với tiến bộ của công nghệ gần đây thì Educational game được xem như là một công cụ giảng dạy đặc lực của giáo viên. Và sau đây là một số tình huống có thể ứng dụng Educational game:

- **Sử dụng như một công cụ rèn luyện:** game ở đây được xem như một phần mềm hỗ trợ người chơi thực hiện các bài thực hành. Người giáo viên có thể sử dụng game để hỗ trợ việc tự học của học sinh, giúp học sinh ôn tập, gợi nhớ lại kiến thức.

- **Dùng để dạy kỹ năng làm việc nhóm:** giống như những phần mềm mô phỏng, một số game có thể hướng dẫn kỹ các kỹ năng làm việc nhóm cơ bản cho người chơi. Ngoài ra, một vài game có thể được chơi chung bởi nhiều người thông qua mạng internet. Do đó, tạo cơ hội cho người chơi rèn luyện và nâng cao kỹ năng phối hợp cũng như làm việc nhóm của mình.

- **Dùng để khen thưởng, khích lệ học sinh:** có lẽ việc sử dụng phổ biến nhất của game là để thưởng cho công việc tốt. Như bạn biết việc khen thưởng, khích lệ, động viên học sinh trong quá trình tiếp thu kiến thức là rất cần thiết. Điều này giúp cho học sinh có cảm giác vui sướng, có động lực học tập hơn rất nhiều. Do đó khi cần thiết bạn nên sử dụng game để động viên tinh thần học sinh.[10]

### 1.2.4 Một số điều cần lưu ý khi sử dụng game trong giáo dục:

- **Sử dụng trò chơi một cách hợp lý:** một số nhà giáo dục cho rằng đa số hiện nay việc sử dụng trò chơi trên máy tính đang bị lạm dụng, sai mục đích và sử dụng không thích hợp. Do đó chúng ta phải lựa chọn và sử dụng hợp lý các trò chơi để các học sinh có hứng thú học tập chứ không chỉ để giải trí đơn thuần.

- **Tác động đến tất cả học sinh:** bạn phải đảm bảo rằng khi tiến hành các game chơi trong khi dạy học thì tất cả các học sinh trong lớp đều phải tham gia và hiểu rõ được ý nghĩa cũng như vai trò của trò chơi.

- **Nhấn mạnh những nội dung kỹ năng:** trước khi học sinh bắt đầu chơi, hãy chắc chắn các học sinh của bạn hiểu được luật chơi và các hoạt động chính trong trò

chơi. Và sinh viên nên tự mình nhận ra những quy luật chung trong game với những gợi ý ban đầu của giáo viên.[10]

### **1.2.5 Các đặc điểm tạo nên tính hấp dẫn trong game giáo dục:**

#### **a. Phải có mục tiêu rõ ràng:**

Một trò chơi được đánh giá là tốt khi bạn xây dựng được mục tiêu rõ ràng. Lúc này người chơi sẽ biết được họ học được cái gì, và vận dụng những kiến thức đã học như thế nào trong trò chơi.

#### **b. Phải chứa nhiều kiến thức và cơ hội thực hành:**

Trong game và mô phỏng, người học được xem, được trình bày với tập hợp các kiến thức và có cơ hội thực hành. Lúc này việc học rất nhẹ nhàng, kiến thức được truyền đạt từ một thế giới đầy màu sắc, đa dạng chứ không trừu tượng, khô khan như trong sách vở.

#### **c. Thường xuyên khen thưởng, khích lệ người chơi:**

Một trò chơi hấp dẫn là trò chơi thường xuyên khích lệ người chơi thông qua các điểm số, vật thưởng hoặc các hình thức khác. Nhờ vào các yếu tố khích lệ này, người chơi sẽ cố gắng đạt được điểm thưởng bằng cách cố gắng tìm hiểu, thu thập thông tin về các cách chơi, chiến lược trong trò chơi.

#### **d. Game phải gần gũi ,liên hệ với thực tế:**

Đối với game giáo dục thì nội dung của game rất quan trọng, do đó nó cần gắn liền với thực tế, gần gũi với người chơi. Càng gần gũi với đời sống của người chơi thì game càng lôi cuốn, hấp dẫn người chơi.

#### **e. Phải có thời gian giới hạn trong trò chơi:**

Để luôn cuốn hút người chơi thì cần có giới hạn thời gian trong game, nếu như không có giới hạn về thời gian thì người chơi sẽ chơi mãi, chơi hoài. Từ đó gây sự nhàm chán, không tạo được sự hứng thú trong khi chơi.

#### **f. Cần có gợi ý, giúp đỡ :**

Cần cung cấp phần giúp đỡ, hỗ trợ người chơi bằng các gợi ý hoặc chỉ dẫn trong từng phần chơi. Đối với game giáo dục chúng ta cần có phần hỗ trợ kiến thức cho người chơi, giúp người học có thể vượt qua các khó khăn về kiến thức trong khi chơi.

#### **h. Tạo được tính đối kháng cao:**

Để tạo tính hấp dẫn cần có sự đối kháng trong game, được thể hiện qua:

- + Sự xung đột của trò chơi.
- + Sự cạnh tranh giữa người chơi với máy tính, giữa những người chơi với nhau.
- + Nhân vật đối lập, phản biện.

#### **g. Xây giao diện đẹp:**

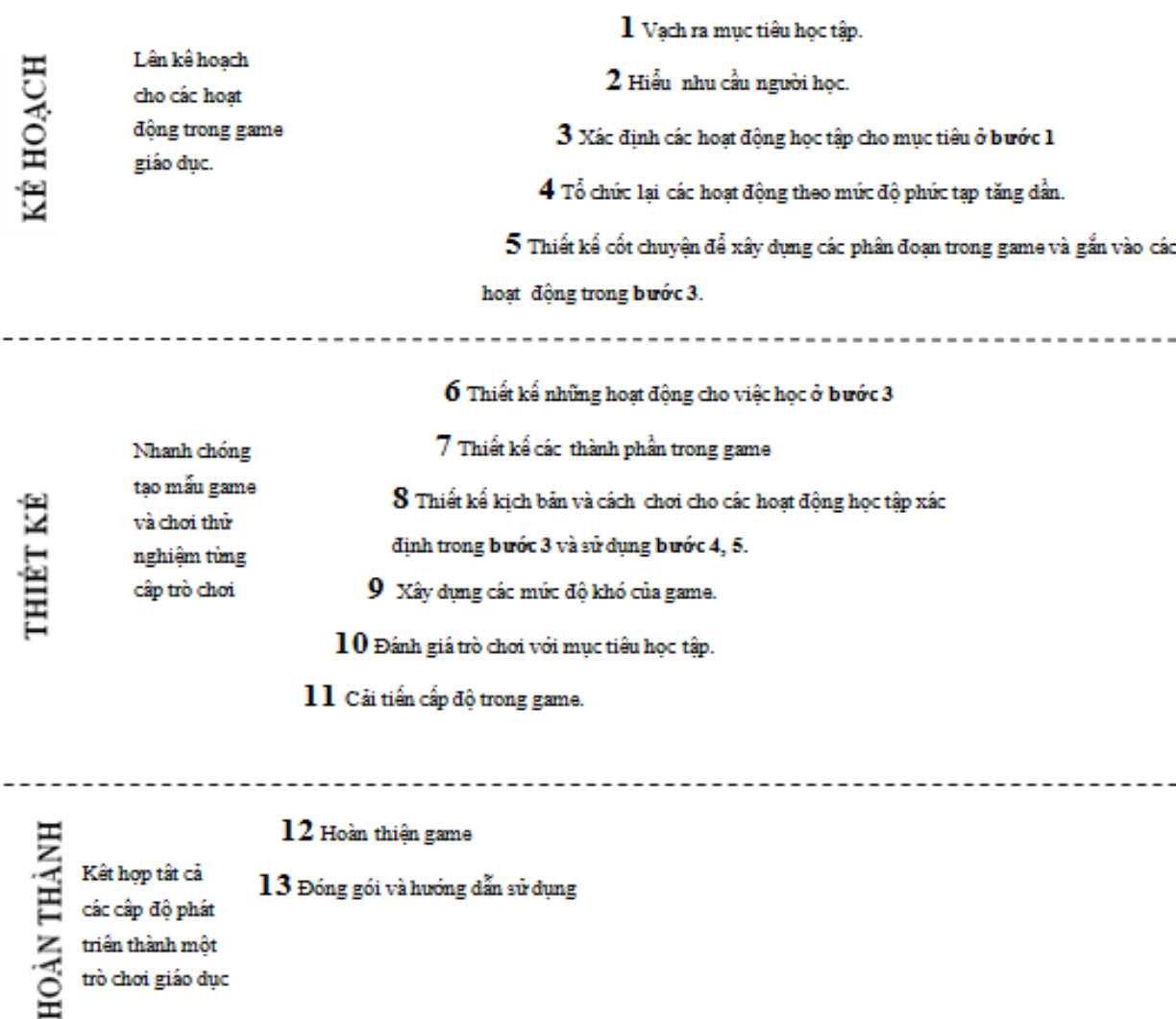
Đây là đặc điểm không thể thiếu đối với bất kỳ một game hấp dẫn nào, người chơi có thích thú hay không phụ thuộc rất nhiều vào đặc điểm này.

#### **1.2.5 Quá trình thiết kế Game Educational:**

Các thể loại game thường lấy ý tưởng từ những cuốn tiểu thuyết, kịch bản của các bộ phim, truyện tranh ... Còn đối với Educational game thì ý tưởng chính phụ thuộc vào nội dung học tập mà tác giả muốn truyền đạt cho người chơi. Do đó thiết kế Educational game khác biệt rất nhiều so với thiết kế trò chơi máy tính thông thường bởi vì nó chủ yếu liên quan đến phương pháp sư phạm. Khi thiết kế trò chơi giáo dục, người giáo viên thường xây dựng trò chơi bằng cách xác định mục tiêu học tập hoặc kết quả học tập đạt được sau khi chơi hơn là dựa vào yếu tố hấp dẫn, lôi cuốn của trò chơi.

Educational game chủ yếu được thiết kế để hỗ trợ người học đạt được mục tiêu học tập tuy nhiên đã là trò chơi thì cần có tính thú vị, hấp dẫn để thu hút người chơi. Do đó trò chơi này thường đặt giáo dục lên hàng đầu và yếu tố giải trí đứng ở vị trí thứ hai. Quá trình tạo một Educational game cần sự kết chặt chẽ giữa người giáo viên và đội ngũ phát triển phần mềm. Đây là một quá trình lặp đi lặp lại các bước (thiết kế, viết game, thử nghiệm và phát hành).[16]

Sơ đồ về thiết kế một Educational game:



Hình 1.3: Sơ đồ thiết kế Educational game [16]

## 1.2.6 Các dạng game thường gặp:

### a. Game hành động:

Thể loại này thường được thiết kế để dạy một môn học cụ thể. Trong đó người chơi thường sẽ nhập vai trở thành một nhân vật trong game. Và để vượt qua các thử thách, trở ngại của từng màn chơi trong game thì họ phải vận dụng kiến thức để giải đáp các câu hỏi, cũng như bài tập trong game đưa ra. Khi giải đáp đúng câu hỏi, bạn sẽ được cung cấp một số công cụ để chống lại kẻ thù trong game. Thể loại game này yêu cầu người chơi sự phản xạ nhanh chóng, chính xác và đúng thời điểm để vượt qua các trở ngại.

Ví dụ: Game **Timez Attack**:



Hình 1.4: Game Timez Attack[19]

- Nhà sản xuất: công ty **Bigbrain**.
- Mục đích: Dạy học môn toán cho trẻ em.
- Bản quyền: Miễn phí.
- Link dowload: <http://www.bigbrainz.com>.
- Chơi online: [http://www.bigbrainz.com/online/select\\_school.php](http://www.bigbrainz.com/online/select_school.php) .

Với trò chơi Timez Attack, được thiết kế trong một môi trường đồ họa phong phú, các em sẽ say mê với câu chuyện trong khung cảnh hấp dẫn và đầy thách thức. Các em không chỉ được học môn toán mà các em còn được trải qua những thời khắc phiêu lưu cuốn hút. Giúp các em có được tư duy nhạy bén, tinh thần dũng cảm và có được những giây phút giải trí vui chơi trong khi ôn luyện và học tập.

## **b. Game phiêu lưu, mạo hiểm:**

Ở dạng game này điểm được chú ý chính là cốt chuyện của game, tốc độ game này thường chậm hơn so với game hành động. Cách thức của game là người chơi phải thu thập thông tin, giải quyết các mã số và bí ẩn để thăng tiến trong trò chơi. Mục tiêu thường gặp ở thể loại trò chơi này là giúp người chơi hiểu được phong tục tập quán, lịch sử của một quốc gia hay một vùng nào đó.

Ví dụ: Game **Ansel & Clair's Adventure's in Africa**



**Hình 1.5: Game Timez Attack[20]**

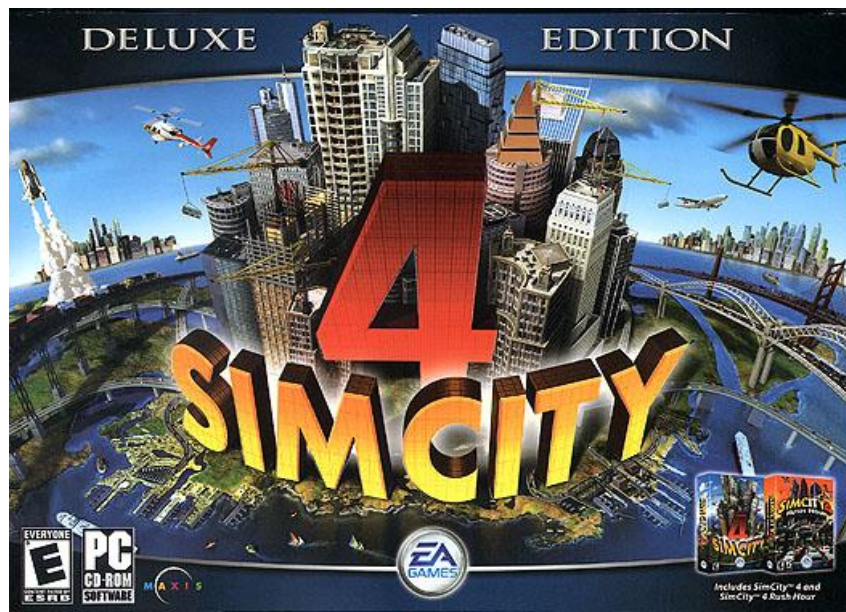
- Nhà sản xuất: Kid Inc
- Mục đích: Dạy học môn địa lý – mô tả về tự nhiên.
- Bản quyền: Có phí.
- Link download: <http://www.bigbrainz.com>.

Là một trò giáo dục phiêu lưu dạy cho trẻ em về ba khu vực chính của Châu Phi: sông Nile, sa mạc Sahara, và Serengeti. Mỗi khu vực có một số loài động vật, yếu tố văn hóa, đặc điểm của vùng đất để học sinh để tìm hiểu. Câu chuyện được mô tả như một cuộc khám phá của một người nước ngoài và một robot đang xâm nhập vào trái đất để tìm hiểu về châu Phi.

### c. Game mô phỏng:

Đây là thể loại mô tả, tái hiện lại các hoạt động trong thế giới thực và có các mục đích khác nhau như: đào tạo, phân tích, dự đoán các hoạt động. Một số loại game mô phỏng nổi tiếng như là game chiến tranh, game kinh doanh hay game nhập vai, đối với lĩnh vực giáo dục thì loại game nhập vai là game chúng ta thường nhận thấy nhất ở thể loại này.

Ví dụ: Game **Sim city**.



Hình 1.6: Game Sim City[21]

- Nhà sản xuất: Maxis, một chi nhánh của Electronic Arts.
- Mục đích: Dạy người chơi cách xây dựng và quản lý một thành phố.
- Bản quyền: Miễn phí
- Link dowload: <http://download123.vn/sim-city-2000-884-download.aspx>

Game cho phép người chơi tạo ra một vùng đất bằng công cụ biến đổi địa hình, sau đó thiết kế và xây dựng các khu vực định cư để phát triển thành phố. Người chơi có thể quy hoạch các khu vực như thương mại, dân cư, và công nghiệp cũng như xây dựng. Và duy trì hoạt động của các dịch vụ công cộng, hệ thống giao thông và công trình công cộng. Thành công của người chơi là phải quản lý được về mặt tài chính, môi trường, và chất lượng cuộc sống cho cư dân thành phố.



#### d. Game đối kháng:

Thể loại game này tập trung vào trận chiến 1-1 giữa hai nhân vật, một trong hai đối thủ này có thể do máy điều khiển. Nếu chơi game này người học sẽ đóng vai trò là một trong 2 nhân vật đối kháng, họ sẽ giải đáp câu đố liên quan đến một môn học nào đó, nếu trả lời nhanh và đủ số lượng câu hỏi thì sẽ đánh bại được kẻ thù.

Ví dụ: Game Zombie Master



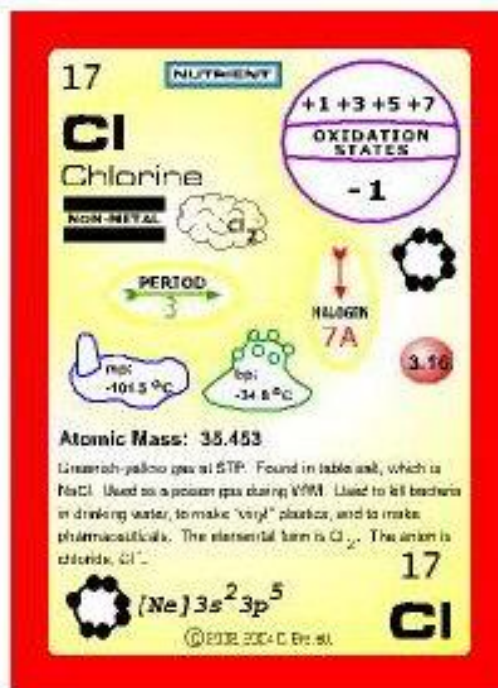
Hình 1.7: Game Zombies Master[22]

- Nhà sản xuất: <http://www.pappumathgames.com>
- Mục đích: Dạy học môn toán
- Bản quyền: Miễn phí
- Link chơi online: <http://www.pappumathgames.com/zombie-master-bt/>

### e. Game dạng thẻ - Card game:

Một hình thức chơi games thông qua các thẻ chơi. Các dạng Flash card games được sử dụng giảng dạy trong toán học, sinh học và một số trò chơi để cải thiện trí nhớ.

Sau đây là một ví dụ game được sử dụng giảng dạy trong hóa học.



Hình 1.8: Game AtomMate[23]

- Nhà sản xuất: công ty DuPont
- Mục đích: Dạy học mô hóa
- Bản quyền: Có phí.
- Link chơi online: <http://boardgamegeek.com/boardgame/22681/atommate>

**AtomMate** là một **card game** hóa học bao gồm một tầng 49 thẻ bài với tên, biểu tượng và định nghĩa về các chất trong bảng tuần hoàn. Trò chơi này được sử dụng để tìm hiểu các vấn đề có liên quan đến các hợp chất hóa học, được thiết kế cho học sinh từ 10 tuổi trở lên.

CHƯƠNG 2

**PHÂN TÍCH VÀ THIẾT  
KẾ TRÒ CHƠI**

## **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ TRÒ CHƠI.**

### **2.1. Giới thiệu trò chơi Secret Garden:**

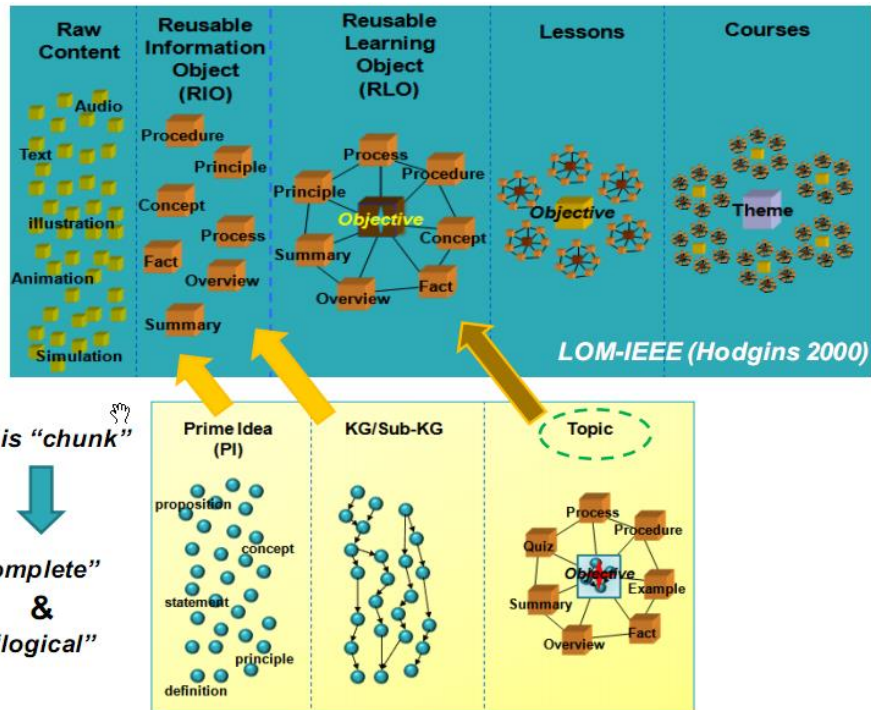
• **Ý tưởng:** Game của chúng em được xây dựng trên ý tưởng trò chơi đối kháng **Plants and Zoobies**. Game sẽ đưa người chơi vào cuộc hành trình ở một khu vườn bí mật – “**Secret Garden**” với nhiều khó khăn, thách thức mà người tham gia phải vượt qua. **Secret Garden** được thể hiện qua nhiều màn chơi với các nhân vật là: bông hoa ẩn chứa kiến thức, người chơi có vai trò là người trồng các bông hoa này trong khu vườn, bên cạnh đó còn có nhân vật đối kháng sẽ ăn mất những bông hoa. Và để vượt qua từng màn chơi thì người tham gia phải vận dụng kiến thức đã học về ngôn ngữ lập trình Pascal để trả lời số lượng câu đố quy định ở mỗi màn chơi.

- **Đối tượng:** Dành cho những biết và yêu thích ngôn ngữ lập trình Pascal.
- **Phạm vi:** Kiến thức tin học phổ thông – Tin học lớp 11.
- **Mục tiêu:** Game sẽ giúp cho người dùng tự học, cũng như ôn tập kiến thức về ngôn ngữ Pascal.

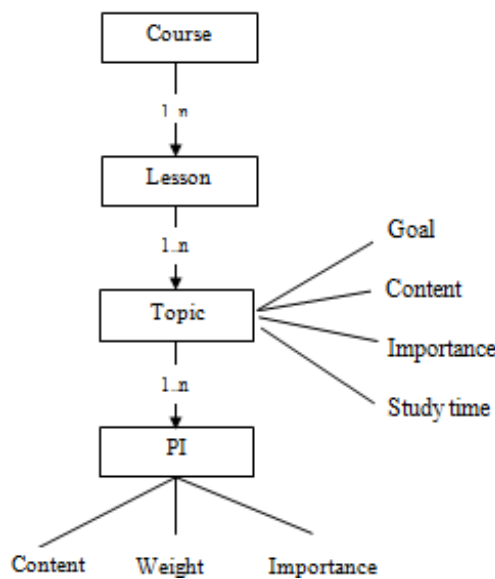
#### **• Các thành phần trong trò chơi:**

##### **- Kiến thức:**

+ Kiến thức trong trò chơi được xây dựng dưới dạng **Knowledge Graph** (tạm dịch là đồ thị tri thức) – viết tắt là **KG**. Ý tưởng cơ bản của mô hình là gắn kết tính sự phạm vào trong quá trình thiết kế và xây dựng nội dung học tập. KG được biểu diễn như sau: **Graph = core content knowledge: node = prime idea**(the smallest unit of learning knowledge) – viết tắt là **PI**; **arc = hard-condition/necessary-condition**(relationship between the prime ideas).



**Hình 2.1: Knowledge Graph ở các mức độ theo ngữ cảnh sử dụng**  
 + PI, là một tập các đơn vị kiến thức học nhỏ nhất – tạm dịch là ý giảng chính, một đơn thể kiến thức (compact knowledge). KG, là sự liên kết các PI thông qua một thứ tự logic về sự bắt buộc phải học trước một kiến thức khác. Giả sử các PI và mối liên hệ giữa chúng được định nghĩa một cách đầy đủ và hợp lý bởi các chuyên gia sư phạm và chuyên gia nội dung.  
 + Sơ đồ tổng quát các thành phần của một khóa học:



**Hình 2.2: Sơ đồ ý nghĩa các thành phần của Khóa học**

**Mô tả:**

1	<b>Course</b>	<i>Khóa học</i> được hiểu là một phân hay một môn học cụ thể được gắn với những yêu cầu về mặt mục tiêu, kiến thức, kỹ năng, và thái độ. Nó được biên soạn từ một tập các thành phần nhỏ hơn gọi là <i>bài học</i> .															
2	<b>Lesson</b>	<i>Bài học</i> , được tổ chức để hoàn tất một trong những mục tiêu chung của một môn học, hay một tập các mục tiêu liên quan															
3	<b>Topic</b>	<p><i>Chủ đề/ đề mục</i> được thiết kế cho việc tự học. Nó là một bài giảng Online hoặc Offline được xác định trên một lượng kiến thức cụ thể để hoàn thành một mục tiêu ở mức thấp, đơn giản. Vì vậy nó cũng được xem như là thành phần nội dung học tập nhỏ nhất cần chuyên tải đến người học. Thành phần của Topic có những thành phần sau</p> <table border="1" data-bbox="576 1151 1355 1361"> <thead> <tr> <th>STT</th> <th>Thành phần</th> <th>Mô tả</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Goal</td> <td>Mục tiêu chủ đề</td> </tr> <tr> <td>2</td> <td>Content</td> <td>Nội dung của Topic</td> </tr> <tr> <td>3</td> <td>Importance</td> <td>Độ quan trọng của chủ đề</td> </tr> <tr> <td>4</td> <td>Study time</td> <td>Thời gian học một chủ đề</td> </tr> </tbody> </table>	STT	Thành phần	Mô tả	1	Goal	Mục tiêu chủ đề	2	Content	Nội dung của Topic	3	Importance	Độ quan trọng của chủ đề	4	Study time	Thời gian học một chủ đề
STT	Thành phần	Mô tả															
1	Goal	Mục tiêu chủ đề															
2	Content	Nội dung của Topic															
3	Importance	Độ quan trọng của chủ đề															
4	Study time	Thời gian học một chủ đề															
4	<b>PI</b>	<p><i>Y chính</i> là tập các đơn vị kiến thức nhỏ nhất. Mỗi Topic có thể có nhiều PI. Mỗi PI có những thành phần sau</p> <table border="1" data-bbox="576 1509 1355 1794"> <thead> <tr> <th>STT</th> <th>Thành phần</th> <th>Mô tả</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Content</td> <td>Nội dung của PI</td> </tr> <tr> <td>2</td> <td>Weight</td> <td>Trọng số - dùng để sắp xếp thứ tự của PI, PI nào cần học trước, PI nào học sau.</td> </tr> <tr> <td>3</td> <td>Importance</td> <td>Độ quan trọng - Dùng để xác định PI nào quan trọng hơn.</td> </tr> </tbody> </table>	STT	Thành phần	Mô tả	1	Content	Nội dung của PI	2	Weight	Trọng số - dùng để sắp xếp thứ tự của PI, PI nào cần học trước, PI nào học sau.	3	Importance	Độ quan trọng - Dùng để xác định PI nào quan trọng hơn.			
STT	Thành phần	Mô tả															
1	Content	Nội dung của PI															
2	Weight	Trọng số - dùng để sắp xếp thứ tự của PI, PI nào cần học trước, PI nào học sau.															
3	Importance	Độ quan trọng - Dùng để xác định PI nào quan trọng hơn.															

[6][7][8]

(Xem chi tiết nội dung kiến thức tin học lớp 11 được tổ chức theo dạng KG trong phần phụ lục )

### **- Nhân vật:**

+ **Nhân vật chính:** Người chơi có nhiệm vụ là trồng các bông hoa trong khu vườn và sẽ chọn các bông hoa để giải quyết các vấn đề như: câu hỏi, câu đố, bài tập...

+ **Nhân vật đối kháng:** con ốc sên, con sâu, con rùa... sẽ tấn công khu vườn của người chơi và không cho người chơi có điều kiện trả lời các câu hỏi bằng cách sẽ ăn các cây hoa mà người chơi trồng được.

+ **Nhân vật hỗ trợ :** mặt trời, ngôi sao, hạt giống... Người chơi sẽ bắt lấy các vật phẩm này khi chúng xuất hiện để trồng hoa cho khu vườn của mình. Ngoài ra còn có người hướng dẫn sẽ giúp đỡ người chơi trong suốt cuộc hành trình. Người này có nhiệm vụ đưa ra các lưu ý, hướng dẫn, nhắc nhở ở từng chặng đường.

### **-Chiến lược:**

+ Ở mỗi màn chơi, hệ thống sẽ cung cấp cho người chơi một số nhân vật hỗ trợ đã quy định trước với thời gian xuất hiện bị giới hạn, người chơi sẽ nhấp chọn các nhân vật này để trồng các bông hoa trong khu vườn.

+ Khi đã trồng được một số bông hoa người chơi có thể chọn lựa hoặc là nhấp vào các bông hoa để trả lời câu hỏi hoặc sẽ tiếp tục nhấp chọn các nhân vật hỗ trợ.

+ Người chơi có thể sử dụng điểm thưởng của mình để có các giúp đỡ: mua thời gian để tạm dừng sự tấn công của các nhân vật đối kháng hoặc có thể xem lại kiến thức trong lúc trả lời câu hỏi.

### **-Thu hoạch:**

+ Người chơi sẽ nhận được điểm thưởng và vật thưởng nếu như trả lời đúng các câu hỏi.

+ Với chiến lược tốt người chơi sẽ nhanh chóng trả lời đủ số câu hỏi để qua các màn chơi.

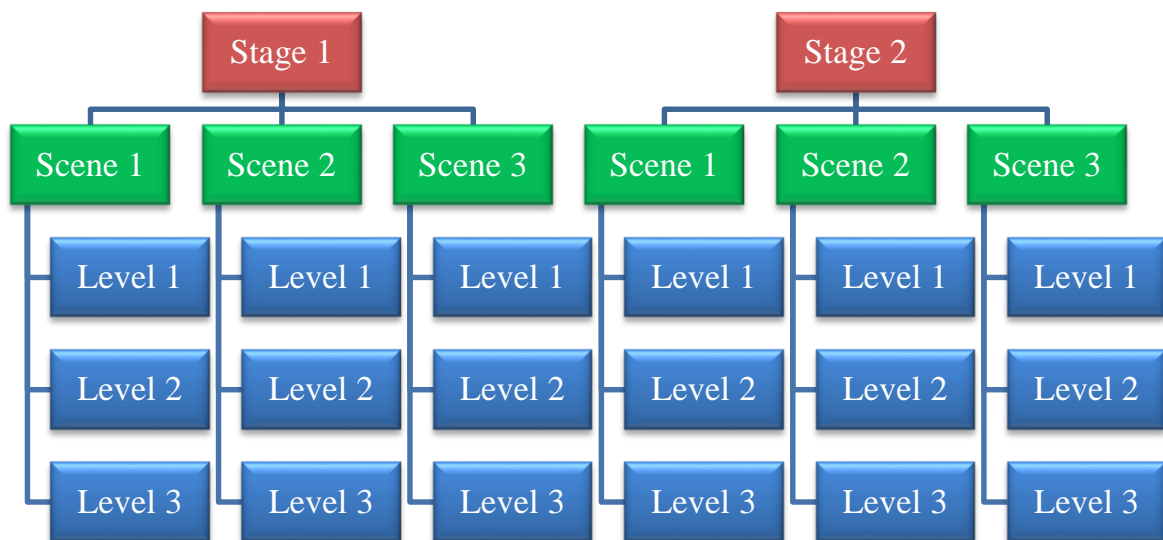
+ Người chơi sẽ học được kiến thức thông qua giải quyết các vấn đề ở mỗi màn chơi.

### **-Luật chơi:**

- + Muốn vượt qua mỗi màn chơi người chơi phải trả lời đủ số câu hỏi quy định trong từng màn.
- + Để trả lời các câu hỏi người chơi phải trồng đủ số bông yêu cầu trong từng màn chơi.
- + Khi người chơi để các nhân vật đối kháng ăn hết các cây bông và tấn công vào nhà của mình coi như thua cuộc.

### **2.2Kịch bản:**

Trò chơi **Secret Garden** dự kiến sẽ có 2 màn chơi (stage), trong mỗi màn chơi được chia làm 3 phân cảnh (scene) và mỗi scene sẽ được chia nhỏ thành 3 cấp độ khác nhau.



Hình 2.3:Cấu trúc trò chơi Secret garden

- **Kịch bản trong mỗi màn chơi như sau:**

- ✓ **Stage 1:**

- Nhân vật chính:

- + Người chơi trồng các bông hoa.

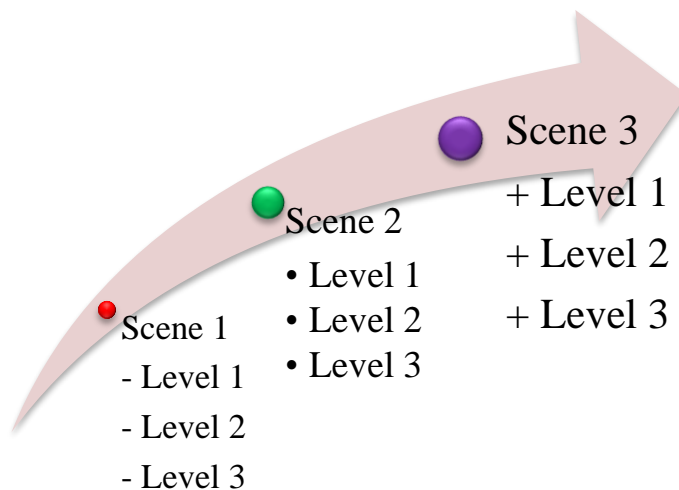
- + Các bông hoa chứa các câu đố và có chức năng ngăn cản đường đi của nhân vật đối kháng.

- Nhân vật đối kháng: Sẽ tấn công và ăn các bông hoa.

- Nhân vật hỗ trợ: Xuất hiện với thời gian cố định.



- Lượng kiến thức: Các câu đố lý thuyết có dạng multichoice, true/false, quiz.
- **Stage 2**: Tương tự như stage 1 nhưng có một số thay đổi sau:
  - Nhân vật chính:
    - + Các bông hoa có khả năng đánh bại nhân vật đối kháng.
  - Nhân vật đối kháng: Tăng về số lượng cũng như tốc độ tấn công.
  - Nhân vật hỗ trợ: Có thêm các vật thưởng dành cho người chơi như:
    - + Phân bón – chăm sóc sức sống của các cây hoa, thuốc trừ sâu – tiêu diệt nhân vật đối kháng.
  - Lượng kiến thức: Bổ sung thêm các câu đố về bài tập, yêu cầu người chơi phải suy nghĩ lâu hơn.
- ✓ **Kịch bản từng scene**: Do cách xây dựng kịch bản trong mỗi scene ở các stage là khá giống nhau nên chúng em xin trình kịch bản các scene trong stage 1.



**Hình 2.4: Sơ đồ kịch bản trong Stage**

- **Scene 1:**

- Input - Đầu vào.

- + Khung cảnh trò chơi là vào ban ngày.

- + Nhân vật đối kháng là 1 loài. ( Con Ốc sên)

- + Nhân vật hỗ trợ: mặt trời, bông hoa, đồng hồ thời gian (20 giây).

- + Kiến thức: Câu hỏi dạng True/False.

- Output – Đầu ra.

- + Vượt qua được 9 câu đố.

- Có 3 cấp độ là: Level 1, level 2, level 3. Kịch bản chi tiết của mỗi level sẽ có 3 thành phần: Input – Đầu vào, Output – Đầu ra, và Process – Xử lý.

- ✚ Level 1:

- Input – Đầu vào:

- + Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 100.

- + Mặt trời: Sẽ xuất hiện trong thời gian là 10 giây.

- + Ốc sên có sức sống là 100. Xuất hiện 1 con tại vị trí biên màn chơi.

- + Mức độ câu hỏi thuộc loại dễ.

- Output – Đầu ra:

- + Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

- Process – Xử lý: ( Xem hình 14: Sơ đồ xử lý level 1)

- ✚ Level 2:

- Input – Đầu vào:

- + Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 100.

- + **Ốc sên có sức sống là 100. Xuất hiện 1 lần 5 con tại vị trí biên màn hình.**

- + **Mức độ câu hỏi thuộc loại trung bình.**

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

- Process – Xử lý:( Xem hình 15: Sơ đồ xử lý level 2)

 Level 3:

- Input – Đầu vào:

+ Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 100.

+ **Mặt trời: Sẽ xuất hiện trong thời gian là 8 giây.**

+ **Ốc sên có sức sống là 100. Xuất hiện ngẫu nhiên trong khoảng cách từ biên màn hình đến hàng bông đầu tiên.**

+ **Mức độ câu hỏi thuộc loại khó.**

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

- Process – Xử lý:( Xem hình 16:Sơ đồ xử lý level 3)



Hình 2.5: Sơ đồ xử lý trong level 1



Hình 2.6:Sơ đồ xử lý trong level 2



Hình 2.7: Sơ đồ xử lý trong level 3

- **Scene 2:**

- Input - Đầu vào.

- + **Khung cảnh trò chơi là vào ban đêm.**

- + **Nhân vật đối kháng là 2 loài. (Con Ốc sên, Con sâu)**

- + **Nhân vật hỗ trợ: Ngôi sao, bông hoa (thay đổi loại bông), đồng hồ thời gian.(15 giây)**

- + **Kiến thức: Câu hỏi dạng True/false và Multichoice.**

- Output – Đầu ra.

- + Vượt qua được 9 câu đố.

- Có 3 cấp độ là: Level 1, level 2, level 3. Kịch bản chi tiết của mỗi level sẽ có 3 thành phần: Input – Đầu vào, Output – Đầu ra, và Process – Xử lý. Do thành phần xử lý khá giống với các cấp độ ở scene 1 nên chúng em chỉ trình 2 thành phần là Input, và output.

- ✚ Level 1:

- Input – Đầu vào:

- + Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 150.

- + Ngôi sao: Sẽ xuất hiện trong thời gian là 10 giây.

- + Ốc sên có sức sống là 100 và con sâu có sức sống là 150. Các con vật xuất hiện lần lượt từng con tại vị trí biên màn hình.

- + Mức độ câu hỏi thuộc loại dễ.

- Output – Đầu ra:

- + Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

- ✚ Level 2:

- Input – Đầu vào:

- + Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 150.

- + Ốc sên có sức sống là 100 và con sâu có sức sống là 150. Xuất hiện với số lượng nhiều hơn. Một lần 5 con đối với mỗi nhân vật.

+ **Mức độ câu hỏi thuộc loại trung bình.**

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

✚ Level 3:

- Input – Đầu vào:

+ Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 150.

+ Ngôi sao: Sẽ xuất hiện trong thời gian là 8 giây.

+ **Ốc sên có sức sống là 100 và con sâu có sức sống là 150. Các con vật xuất hiện lần lượt ngẫu nhiên từ vị trí biên màn chơi đến hàng bông đầu tiên.**

+ **Mức độ câu hỏi thuộc loại khó.**

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

• **Scene 3:**

- Input - Đầu vào.

+ **Khung cảnh trò chơi là vào ban ngày tại bờ sông.**

+ **Nhân vật đối kháng là 3 loài. (Con Ốc sên, con sâu, con rùa)**

+ **Nhân vật hỗ trợ: Ngôi sao, bông hoa (thay đổi loại bông), đồng hồ thời gian.(15 giây)**

+ **Kiến thức: Câu hỏi dạng True/false, Multichoice và Quiz**

- Output – Đầu ra.

+ Vượt qua được 9 câu đố.

- Có 3 cấp độ là: Level 1, level 2, level 3. Kịch bản chi tiết của mỗi level sẽ có 3 thành phần: Input – Đầu vào, Output – Đầu ra, và Process – Xử lý (tương tự các level ở scene 1).

✚ Level 1:

- Input – Đầu vào:



+ Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 200.

+ Mặt trời: Sẽ xuất hiện trong thời gian là 15 giây.

+ Ốc sên có sức sống là 100 và con sâu có sức sống là 150, con rùa có sức sống là 200. Các con vật xuất hiện lần lượt từng con tại vị trí biên màn hình.

+ Mức độ câu hỏi thuộc loại dễ.

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

### **Level 2:**

- Input – Đầu vào:

+ Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 200.

+ Ốc sên có sức sống là 100 và con sâu có sức sống là 150, con rùa có sức sống là 200. . Xuất hiện với số lượng nhiều hơn. Một lần 5 con đối với mỗi nhân vật.

+ Mức độ câu hỏi thuộc loại trung bình.

- Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

### ✚ Level 3:

#### - Input – Đầu vào:

+ Bông hoa: Mỗi cây trồng được tương ứng là 5 điểm. Và cây hoa có sức sống là 150.

+ Ngôi sao: Sẽ xuất hiện trong thời gian là 8 giây.

+ Ốc sên có sức sống là 100, con sâu có sức sống là 150, con rùa có sức sống là 200. Các con vật xuất hiện lần lượt ngẫu nhiên từ vị trí biên màn chơi đến hàng bông đầu tiên.

+ Mức độ câu hỏi thuộc loại khó.

#### - Output – Đầu ra:

+ Trả lời đúng 3 câu hỏi, số điểm phải lớn hơn 0 và chưa bị con vật nào tấn công vào nhà.

## 2.2 Đặc tả về yêu cầu đối với phần mềm trò chơi Secret Garden:

### 2.2.1 Yêu cầu chức năng:

#### - Đối với phía người dùng – người chơi:

STT	Yêu cầu chức năng
1	Tạo một tài khoản mới để tham gia trò chơi.
2	Lưu thông tin, điểm số và các màn chơi mà người chơi đã tham gia.
3	Cho phép người dùng tham gia lại các màn chơi cũ để nâng điểm số hoặc chơi tiếp tục các màn chơi kế tiếp.
4	Trong lúc đang chơi người dùng có thể lựa chọn Pause (tạm dừng), Resume (tiếp tục), Exit (thoát khỏi game).
5	Người dùng được học tập thông qua trả lời các câu đố hoặc có thể dừng game để ôn tập lại kiến thức.

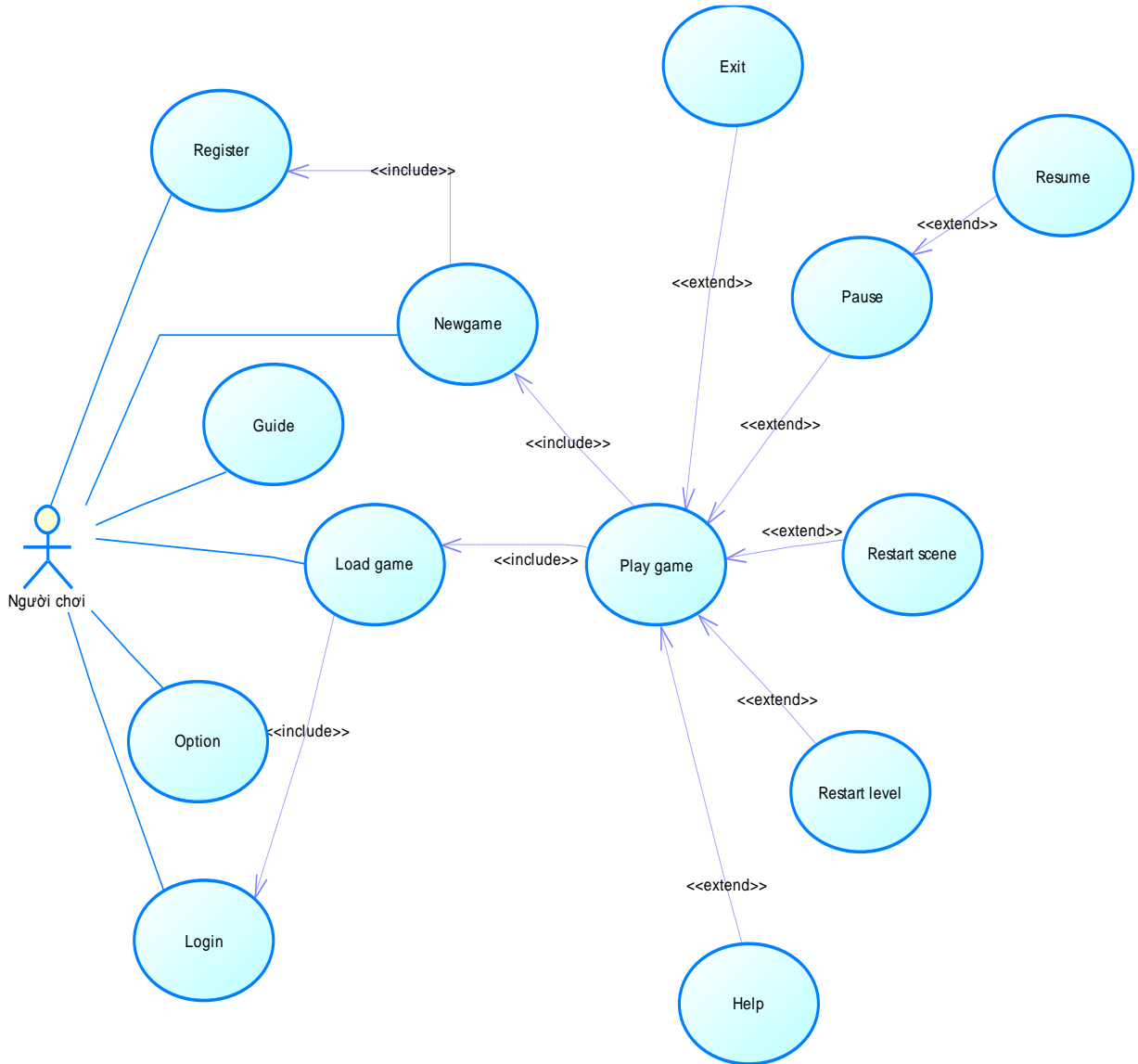
#### - Đối với phía hệ thống:

STT	Yêu cầu chức năng
1	Phân cấp các cấp độ (level), cảnh chơi (scene) trong từng màn chơi (stage). Cụ thể là có 3 scene trong mỗi stage, với mỗi scene lại có 3 level.
2	Mỗi màn chơi sẽ ấn định các nhân vật, số lượng - mức độ - loại câu hỏi khác nhau.
3	Cập nhật thông tin, điểm số người dùng.
4	Lưu trữ trạng thái của các màn chơi đối với mỗi người dùng.

### 2.2.2 Yêu cầu phi chức năng:

STT	Yêu cầu chức năng
1	Cho phép cấu hình thông số trong game: âm thanh, số lượng cũng như tốc độ các nhân vật... Thông qua việc điều chỉnh tập tin *.xml.
2	Thông kê danh sách những người chơi đạt điểm cao nhất. (Top 5)
3	Hỗ trợ cài đặt trên môi trường Window XP trở lên.
4	Cơ sở dữ liệu lưu trữ ở dạng file XML.

### 2.2.3 Mô hình Use Case:



**Hình 2.8: Sơ đồ Use Case**

- Danh sách các Actor của mô hình:**

STT	Actor	Ý nghĩa
1	Player	Người tham gia vào trò chơi.

- Danh sách các Use case của mô hình:**

STT	Use case	Ý nghĩa
1	Register	Đăng ký một người chơi mới.
2	Guide	Xem hướng dẫn về game.
3	Option	Tùy chỉnh thông số trong game.
4	Login	Đăng nhập vào game.
5	New game	Tạo mới trò chơi.
6	Load game	Tiếp tục chơi các màn chơi trong game.
7	Play game	Chơi trò chơi.
8	Exit	Thoát khỏi trò chơi.
9	Pause	Tạm dừng trò chơi.
10	Resume	Tiếp tục chơi sau khi tạm dừng.
11	Restart scene	Chơi lại ở cấp độ scene 1.
12	Restart level	Chơi lại level 1 ở đầu mỗi scene.
13	Help	Trợ giúp trong lúc tham gia trò chơi.

## 2.2.4 Đặc tả một số Use Case quan trọng:

### a) Register:

#### + Tóm tắt:

- Cho phép người dùng tạo một tài khoản cho mình trước khi chơi game.

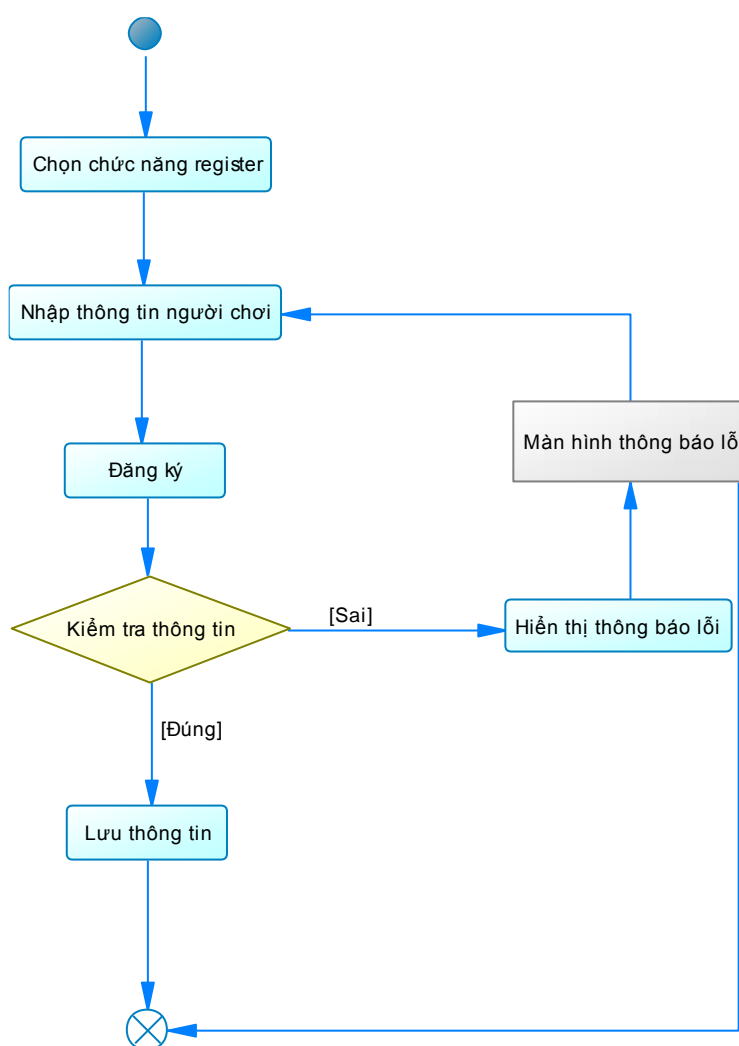
#### + Điều kiện tiên quyết :

- Không có.

#### + Kết quả :

- Sau khi thực hiện thành công hệ thống sẽ lưu thông tin người dùng.

#### + Dòng sự kiện:



Hình 2.9:Dòng sự kiện use case Register

**b) Guide:**

**+ Tóm tắt**

- Cho phép người dùng xem thông tin giới thiệu về trò chơi.

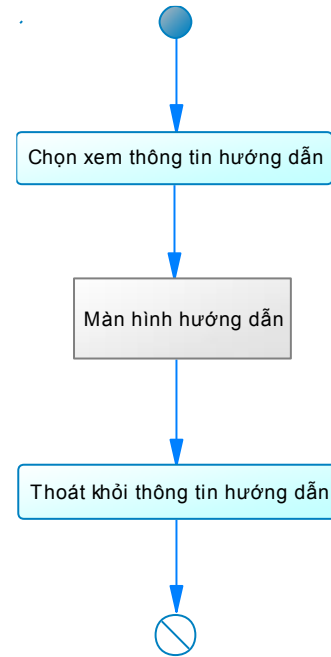
**+ Điều kiện tiên quyết :**

- Không có.

**+ Kết quả :**

- Không có

**+ Dòng sự kiện:**



**Hình 2.10:Dòng sự kiện use**

**c) Option:**

**+ Tóm tắt**

- Cho phép người dùng chỉnh thông số liên quan đến trò chơi (âm thanh)

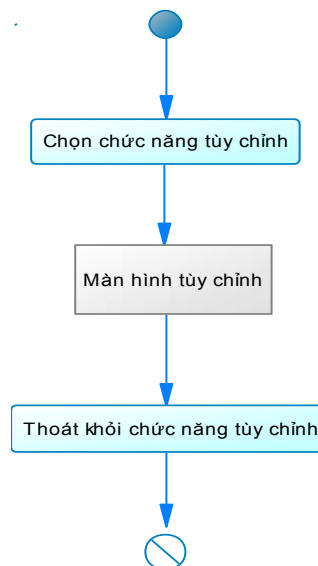
**+ Điều kiện tiên quyết :**

- Không có.

**+ Kết quả :**

- Các thông số trong game sẽ thay đổi theo tùy chỉnh của người dùng.

**+ Dòng sự kiện:**



**Hình 2.11:Dòng sự kiện use case Option**

**d) Login:**

**✚ Tóm tắt:**

- Thực hiện chức năng đăng nhập vào trò chơi đối với người dùng.

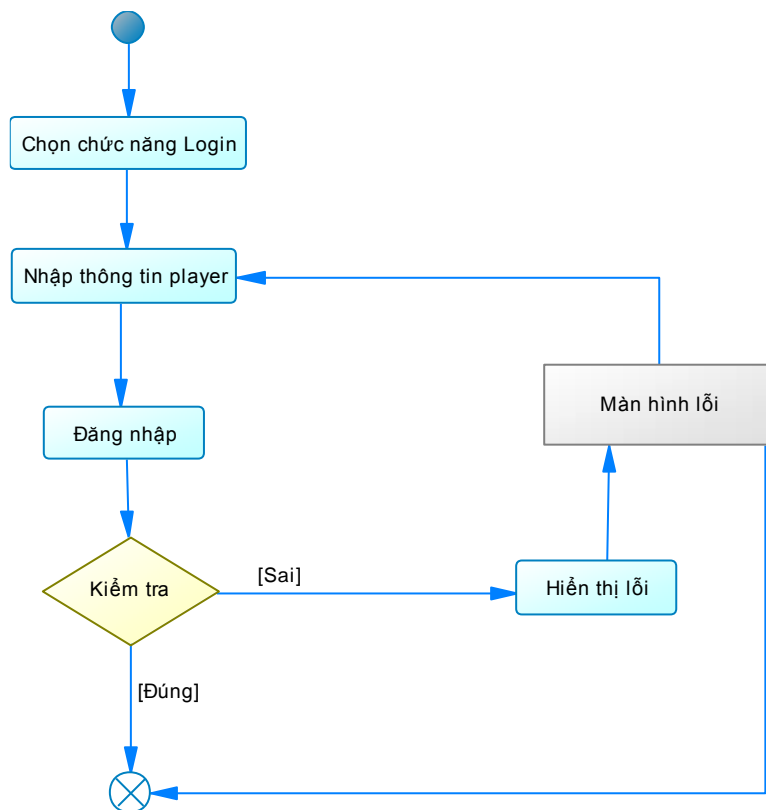
**✚ Điều kiện tiên quyết :**

- Không có.

**✚ Kết quả :**

- Cho phép người dùng chơi game.

**✚ Dòng sự kiện:**



**Hình 2.12:Dòng sự kiện use case Login**

**e. New game:**

**+ Tóm tắt:**

- Cho phép chơi một game mới (Bắt đầu chơi game ở màn chơi đầu tiên)

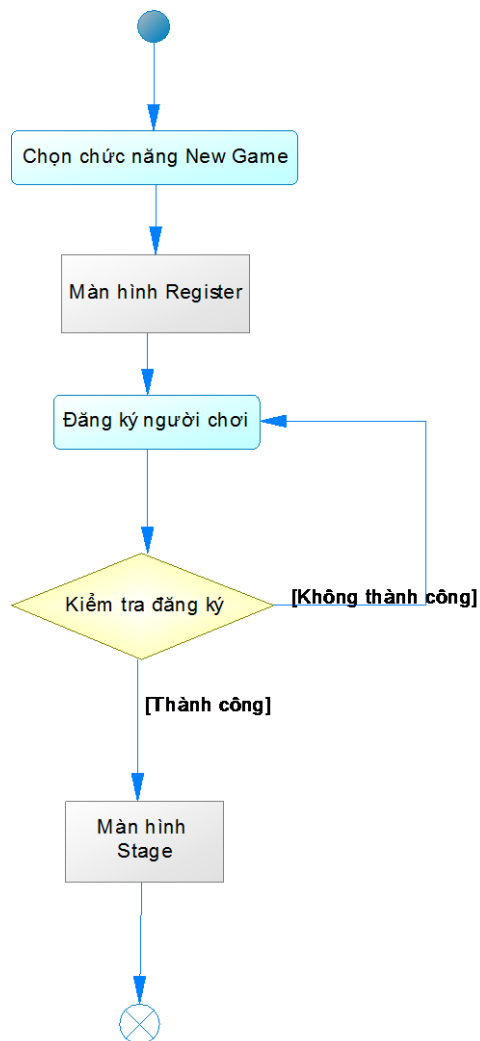
**+ Điều kiện tiên quyết :**

- Phải đăng ký là một người chơi của game.

**+ Kết quả :**

- Không có

**+ Dòng sự kiện:**



**Hình 2.13:Dòng sự kiện use case New game**

**f. Load game:**

**+ Tóm tắt:**

- Cho phép người dùng tiếp tục chơi các màn chơi ở lần chơi trước.

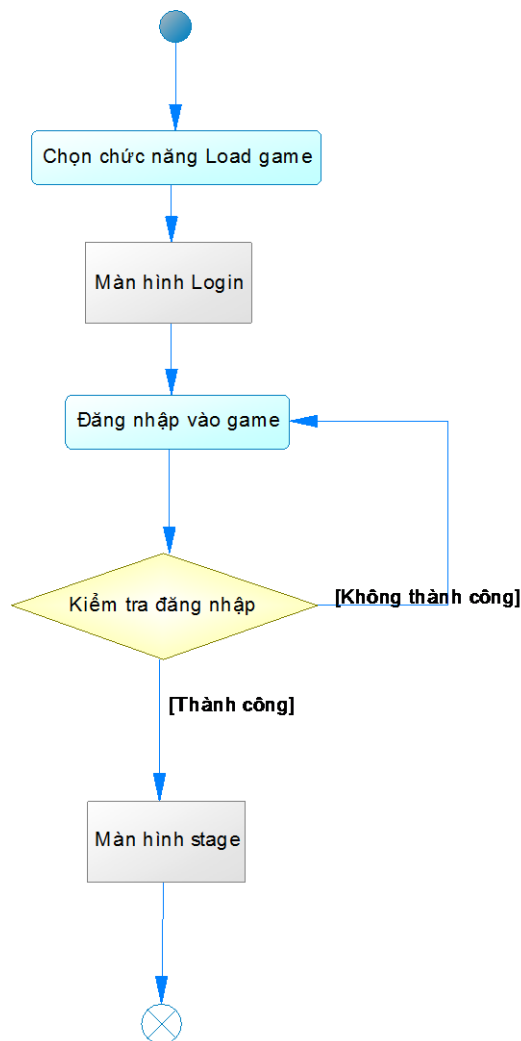
**+ Điều kiện tiên quyết :**

- Phải đăng nhập vào hệ thống trước khi thực hiện chức năng này

**+ Kết quả :**

- Không có

**+ Dòng sự kiện:**



**Hình 2.14: Dòng sự kiện use case Load game**



**g) Play game:**

**+ Tóm tắt:**

- Đây là chức năng chung của 2 use case: Load game và New game.

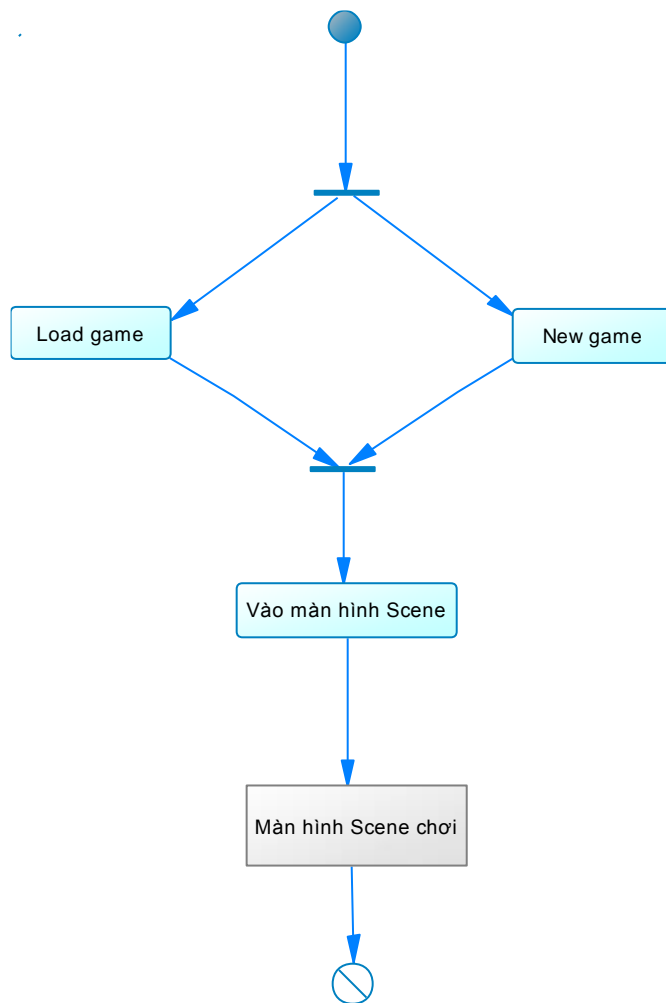
**+ Điều kiện tiên quyết :**

- Phải thực hiện một trong 2 chức năng: Load game và New game.

**+ Kết quả :**

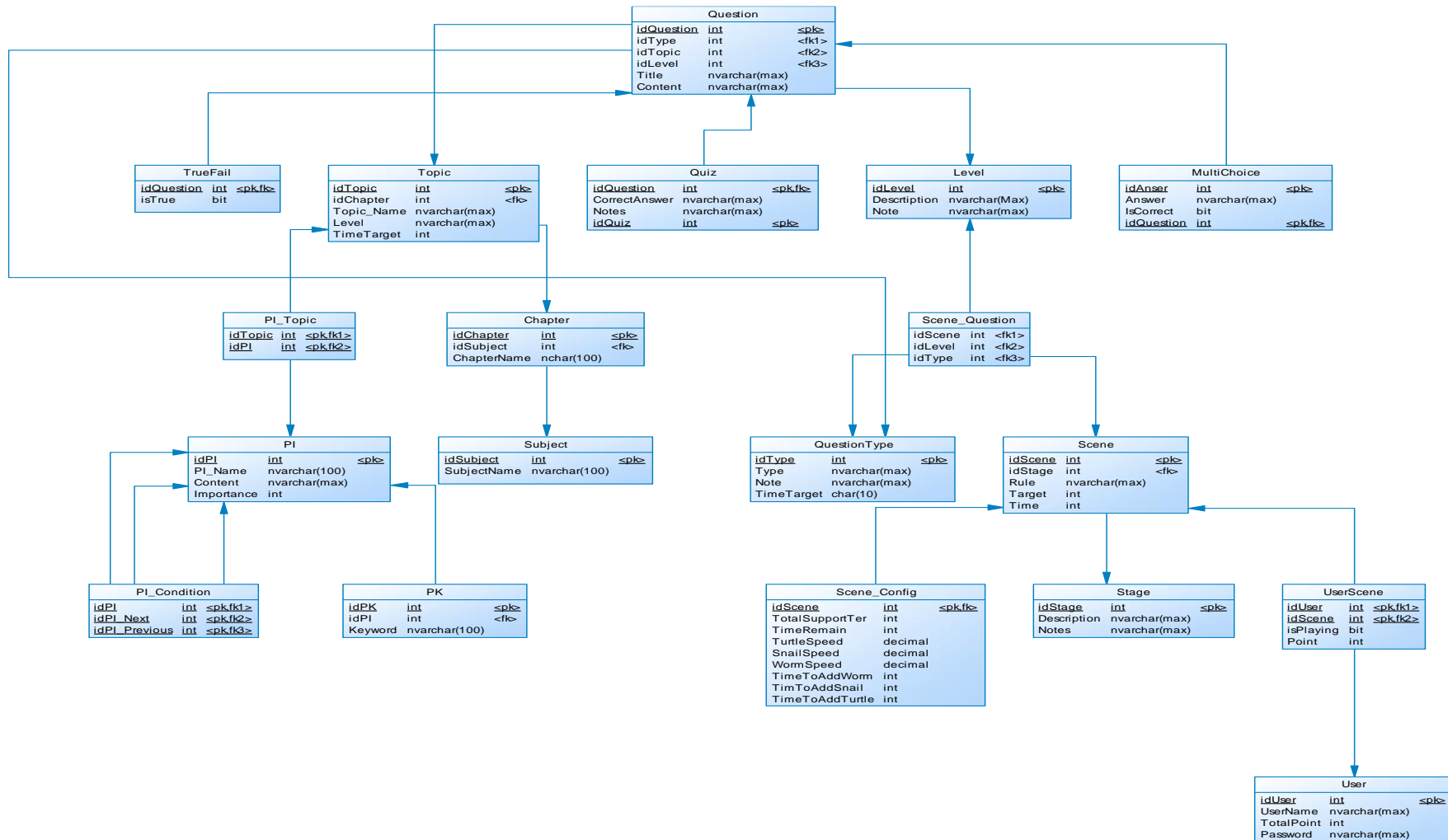
- Không có.

**+ Dòng sự kiện:**



**Hình 2.15:Dòng sự kiện use case Play game**

### 2.3 Thiết kế dữ liệu:



Hình 2.16: Mô hình PDM

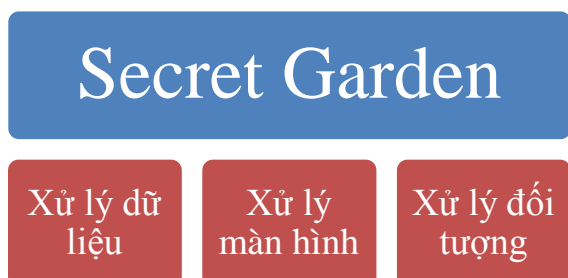
Mô tả chi tiết các bảng:

Scene				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idScene	Int	X	X	Mã cảnh chơi
idStage	int		X	Mã màn chơi
Rule	string		X	Luật chơi
Target	Int		X	Số câu trả lời đúng cần đạt
Time	Int			Thời gian chơi của scene
Stage				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idStage	int		X	Mã màn chơi
Description	string		X	Mô tả về stage
Note	String		X	Ghi chú
User				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idUser	int		X	Mã màn chơi
UserName	string		X	Mô tả về stage
TotalPoint	String		X	Ghi chú
Password	String			Mật khẩu
User Scene				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idUser	int		X	Mã người chơi
idScene	int		X	Mã scene
isPlaying	bit		X	Cho biết là người chơi có đang chơi scene đó hay không
Point	int			Điểm của scene.
Scene Config				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idScene	Int	X	X	Mã Scene
TotalSupporter	int		X	Số nhân vật hỗ trợ giúp tạo nhanh cây bông
TimeRemain	Int		X	Thời gian tạm dừng khi sử dụng đồng hồ
TurtleSpeed	double		X	Tốc độ của con rùa
SnailSpeed	double			Tốc độ của con ốc sên
WormSpeed	double			Tốc độ của con sâu
TimeToAddSnail	Int			Thời gian xuất hiện con ốc sên
TimeToAddWorm	Int			Thời gian xuất hiện con sâu
TimeToAddTurtle	Int			Thời gian xuất hiện con rùa
Question				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idQuestion	Int	X	X	Mã câu hỏi
Content	String		X	Nội dung câu hỏi
idTopic	Int		X	Mã chủ đề của câu hỏi
idType	Int		X	Loại câu hỏi
idLevel	Int			Mức độ khó của câu hỏi
TrueFalse				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idQuestion	Int	X	X	Mã câu hỏi
isTrue	bit		X	Trả về kết quả của câu hỏi (đúng hay sai)
PI_TOPIC				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idTopic	Int	X	X	Mã chủ đề
idPI	int		X	Mã PI

Quiz				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idQuestion	Int	X	X	Mã câu hỏi
idQuiz	Int			Mã câu Quiz
CorrectAnswer	String			Đáp án đúng
Note	String		X	Ghi chú
Chapter				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idChapter	Int	X	X	Mã chương
idSubject	int		X	Mã môn
ChapterName	string		X	Tên của chương
Multichoice				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idQuestion	Int	X	X	Mã câu hỏi
idAnswer	Int			Mã đáp án
IsCorrect	bit			Trả về kết quả của câu hỏi (đúng hay sai)
Answer	String		X	Đáp án
Question_Scene				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idScene	Int	X	X	Mã scene
idType	Int		X	Loại câu hỏi
idLevel	Int			Mức độ khó của câu hỏi
PI				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idPI	Int	X	X	Mã PI
PI_Name	string		X	Tên PI
Content	String		X	Nội dung của PI
Importance	Int			Độ quan trọng của PI
PI_Condition:				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idPI	Int	X	X	Mã PI
IdPI_Next	Int		X	Mã PI trước
IdPI_Previous	Int		X	Mã PI sau
Question Type				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idType	Int	X	X	Mã loại câu hỏi
Type	String		X	Loại câu hỏi
Note	string		X	Ghi chú
TimeTarget	Int		X	Thời gian dự kiến
Level				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idLevel	Int	X	X	Mã mức độ khó
Level	String		X	Mô tả mức độ khó
Note	string		X	Ghi chú
Topic				
Thuộc tính	Kiểu	Key	Not Null	Diễn giải
idTopic	Int	X	X	Mã chủ đề
idChapter	String		X	Mã chương của chủ đề
Level	string		X	Mức độ khó của chủ đề
TimeTarget	Int			Thời gian học dự kiến của chủ đề

## 2.4 Thiết kế xử lý:

### ✚ Tổng quan về các lớp xử lý:



**Hình 2.17:** Sơ đồ tổng quan các lớp xử lý

### ✚ Giải thích:

Lớp xử lý dữ liệu dùng để lưu thông tin, cập nhật dữ liệu của các đối tượng trong game như *người chơi, câu hỏi, cấu hình màn chơi, cấu hình câu hỏi theo màn chơi...*

Lớp xử lý màn hình: dùng để hiển thị những dữ liệu như câu hỏi, trợ giúp ... và hình dạng của nhân vật trong trò chơi ví dụ như khi cây bông trả lời đúng, khi hai nhân vật tiếp xúc nhau...

Lớp xử lý đối tượng: dùng để xử lý những nhân vật có trong trò chơi ví dụ như chuyển động, di chuyển, cập nhật hình dạng của nhân vật.

## 2.4.1 Chi tiết các lớp xử lý dữ liệu:

# Xử lý dữ liệu

Topic: Xử lý chủ đề

PI: xử lý những kiến thức

PI\_Topic: xử lý Những kiến thức liên quan đến chủ đề

QuestionType: Các loại câu hỏi

Level: Mức độ khó của câu hỏi

Scene\_Question: Những câu hỏi theo màn chơi

Question: Xử lý Các câu hỏi

TrueFalse: Câu hỏi dạng đúng sai

Multichoice: Câu hỏi dạng nhiều đáp án

Quiz: Câu hỏi dạng câu đố

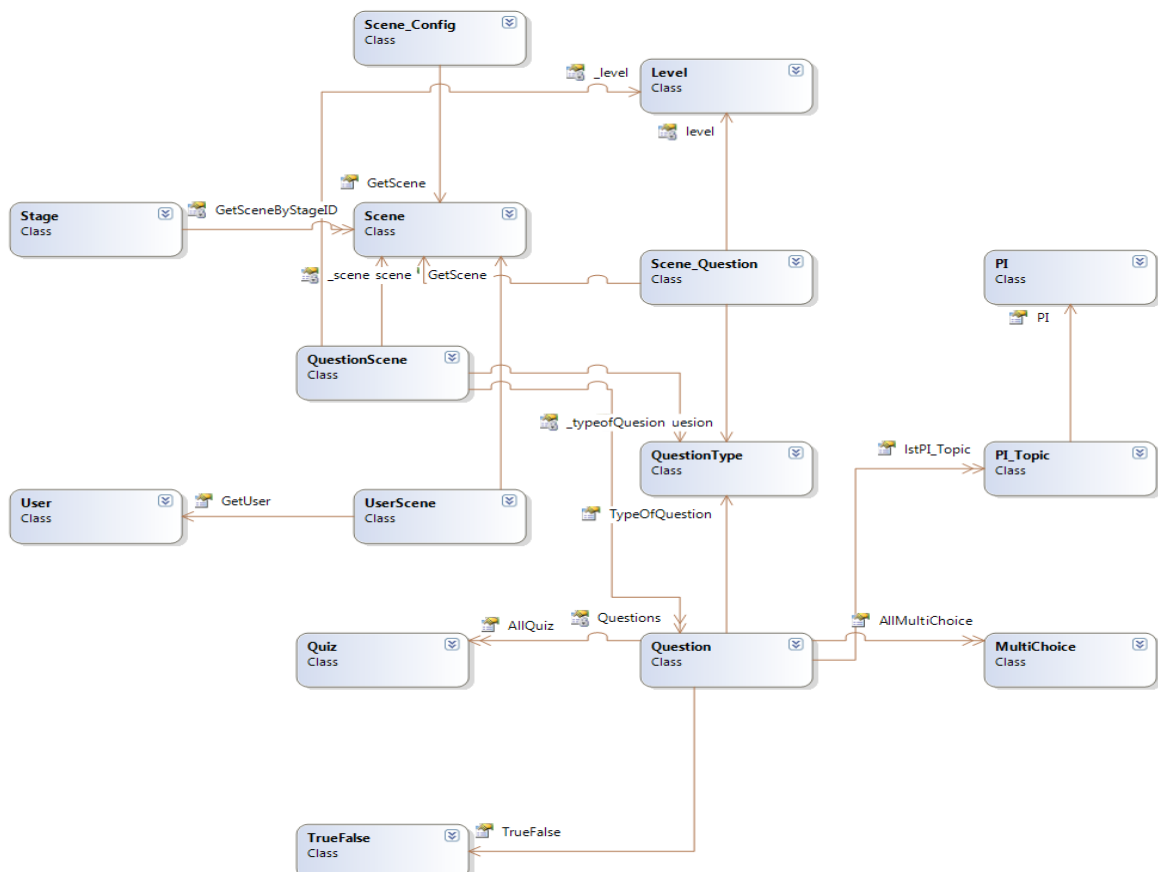
User: Xử lý liên quan đến người chơi

Stage: những stage có trong game

Scene: Xử lý liên quan đến màn chơi

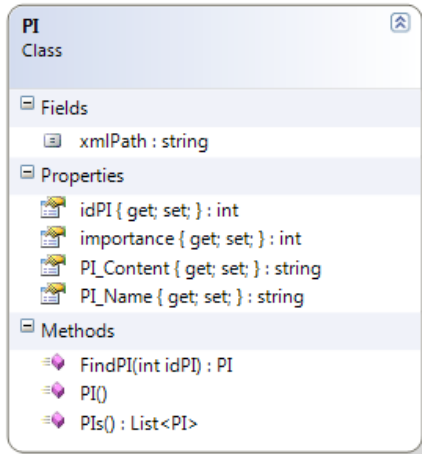
User\_Scene: Xử lý liên quan đến những màn chơi của chơi

Scene\_Config: Tùy chỉnh một số thuộc tính của màn chơi



Hình 2.18: Các lớp xử lý dữ liệu

- Lớp PI:Lấy thông tin của kiến thức để người chơi học và ôn tập.

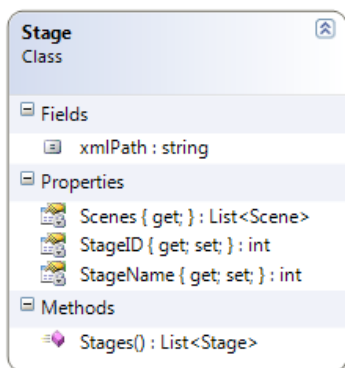


**Hình 2.19: Lớp PI**

Thuộc tính			
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng
xmlPath	const string	private	Đường dẫn file xml
idPI	int	public	Mã PI
PI_Content	string	public	Nội dung của PI
importance	int	public	Độ quan trọng
PI_Name	string	public	Tên của PI
Phương thức			

Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
PIs	public	NULL	List<PI>	Trả về tất cả những PI
FindPI	public	int idPI	PI	Trả về PI tương ứng với mã PI (idPI)

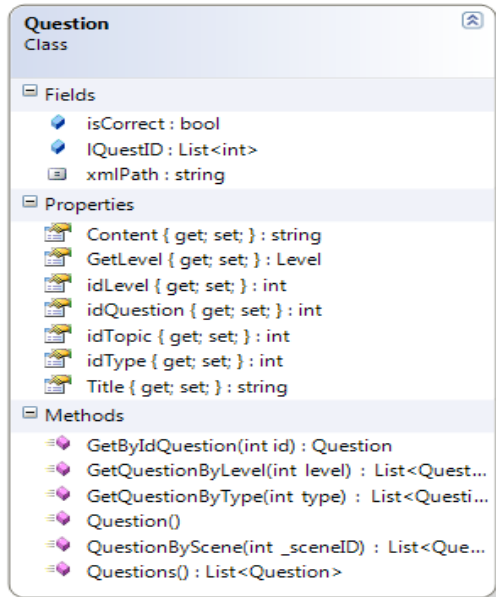
- Lớp Stage:Lấy thông tin của màn chơi.



Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Stages	public	NULL	List<Stage>	Lấy tất cả những Stage
Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
StageID	int	public	Mã Màn chơi	
StageName	string	public	Tên của Màn chơi	
GetSceneByStageID	List<Scene>	public	Trả về những Scene tương ứng với mã Stage(StageID)	

**Hình 2.20: Lớp Stage**

- Lớp Question:Giúp lấy thông tin câu hỏi



**Hình 2.21: Lớp Question**

Danh sách thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
isCorrect	Bool	public	Trả về true hay false tùy theo người chơi trả lời câu hỏi	
IQestID	List<int>	private	Danh sách những mã câu hỏi mà người chơi đã trả lời.	
xmlPath	String	private	Đường dẫn của file xml chứa câu hỏi	
Content	String	public	Nội dung của câu hỏi	
GetLevel	Level(class)	public	Trả về mức độ khó của câu hỏi	
idLevel	Int	public	Mã mức độ khó của câu hỏi	
idQuestion	Int	public	Mã câu hỏi	
idTopic	Int	public	Mã Chủ đề của câu hỏi	
idType	Int	public	Mã loại câu hỏi	
Title	String	public	Tiêu đề của câu hỏi	
AllMultiChoice	List<MultiChoice>	public	Trả về những câu hỏi dạng nhiều lựa chọn theo mã câu hỏi	
AllQuiz	List<Quiz>	public	Trả về những câu hỏi dạng Quiz theo mã câu hỏi	
TrueFalse	TrueFalse	public	Trả về những câu hỏi đúng sai theo mã câu hỏi	
Danh sách phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
GetById	public	int id	Question	Câu hỏi theo SceneID
GetByIdLevel	public	int level	Level	Danh sách câu hỏi theo mức độ
GetByType	public	int type	Type	Danh sách câu hỏi theo loại
QuestionByScene	public	int _sceneID	List<Question>	Danh sách những câu hỏi theo SceneID
Questions	public	NULL	List<Question>	Trả về tất cả những câu hỏi



- Lớp PI\_Condition:Giúp sắp xếp thông tin PI theo điều kiện trước sau .

**PI\_Condition**  
Class

Properties

- idPI { get; set; }...
- idPI\_Next { get; set; }...
- idPI\_Previous { ...

Thuộc tính			
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng
xmlPath	const string	private	Đường dẫn file xml
idPI	int	public	Mã PI
idPI_Previous	string	public	Mã PI trước đó
idPI_Next	int	public	Mã PI tiếp theo

**Hình 2.22: Lớp PI\_Condition**

- Lớp Quiz:Giúp lấy thông tin câu hỏi dạng câu đố.

**Quiz**  
Class

Fields


- xmlPath : string

Properties

- CorrectAnswer { get; set; } : string
- idQuestion { get; set; } : int
- idQuiz { get; set; } : int
- Note { get; set; } : string

Methods

- AllQuiz() : List<Quiz>
- GetByIdQuest(int id) : List<Quiz>
- Quiz()
- Quiz(XElement xelement)



Để mô tả < kiểu số thực > với miền giá trị từ  $2.9 \times 10^{-39}$   
→  $1.7 \times 10^{+38}$  ta dùng từ khóa nào?

real

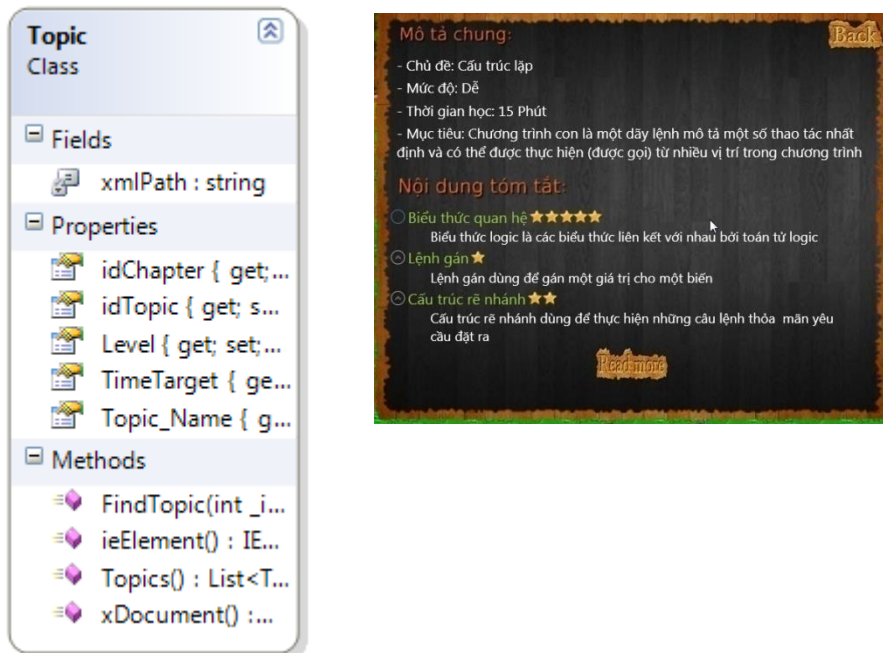
Ok Cancel

**Hình 2.23: Lớp Quiz**

## Mô tả

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
idQuestion	int	public	Mã câu hỏi	
CorrectAnswer	string	public	Đáp án đúng	
Note	string	public	Ghi chú	
idQuiz	int	public	Mã Quiz	
xmlPath	const string	private	Đường dẫn file xml	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
AllQuiz	public	NULL	List<Quiz>	Tất cả những Quiz
GetByIdQuest	public	int id	List<Quiz>	Trả về Quiz theo mã câu hỏi

✚ Lớp Topic: Lấy ra thông tin về chủ đề của câu hỏi.

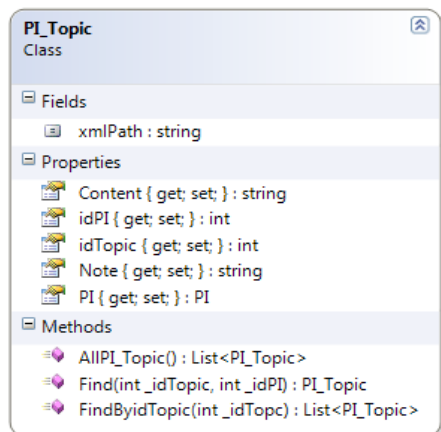


Hình 2.24: Lớp Topic

**Mô tả**

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idTopic	int	public	Trả về chủ mã chủ đề	
idChapter	int	public	Trả về mã chương tương ứng	
Topic_Name	string	public	Tên của chủ đề	
Level	string	public	Trả về thông tin mức độ khó	
TimeTarget	int	public	Thời gian học dự kiến của chủ đề	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Topics	public	NULL	List<Topic>	Trả về tất cả chủ đề
FindTopic	public	int id	Topic	Trả về chủ đề tương ứng với mã chủ đề(idTopic)

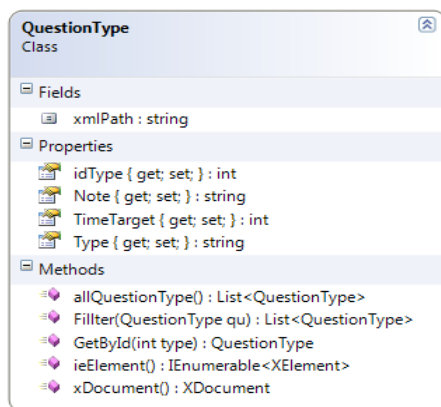
**✚ Lớp PI\_Topic:** Dùng để lấy thông tin những PI theo Topic



Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idPI	int	public	Mã PI	
idTopic	int	public	Mã chủ đề	
Content	string	public	Nội dung	
PI	PI	public	Trả về PI tương ứng với mã PI	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Find	public	int_idTopic, int_idPI	PI_Topic	Trả về PI_Topic theo mã Topic và mã PI
FindByIdTopic	public	int_idTopic	List<PI_Topic>	Trả về những PI_Topic theo mã Topic
AllPI_Topic	public	NULL	List<PI_Topic>	Trả về tất cả những PI_Topic

**Hình 2.25: Lớp PI\_Topic**

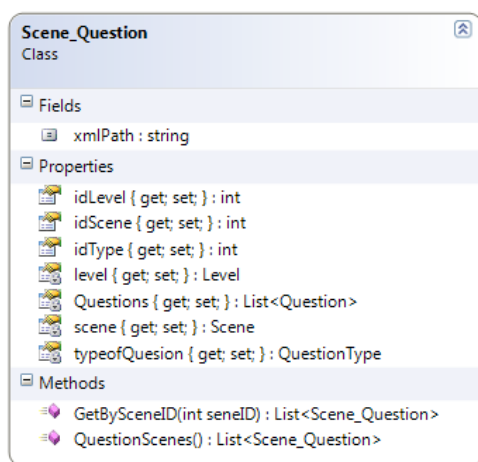
**✚ Lớp Question\_Type:** Lấy thông tin về loại của câu hỏi



Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idType	int	public	Mã loại câu hỏi	
Type	tring	public	Tên loại câu hỏi	
Note	string	public	Ghi chú	
TimeTarget	int	public	Thời gian dự kiến trả lời của loại câu hỏi	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
allQuestionType	public	NULL	List<QuestionType>	Lấy tất cả những loại câu hỏi
GetById	public	int id	QuestionType	Lọc loại câu hỏi theo mã câu hỏi
Fillter	public	NULL	List<QuestionType>	Lọc loại câu hỏi theo điều kiện cho của loại câu hỏi

**Hình 2.26: Lớp Question\_Type**

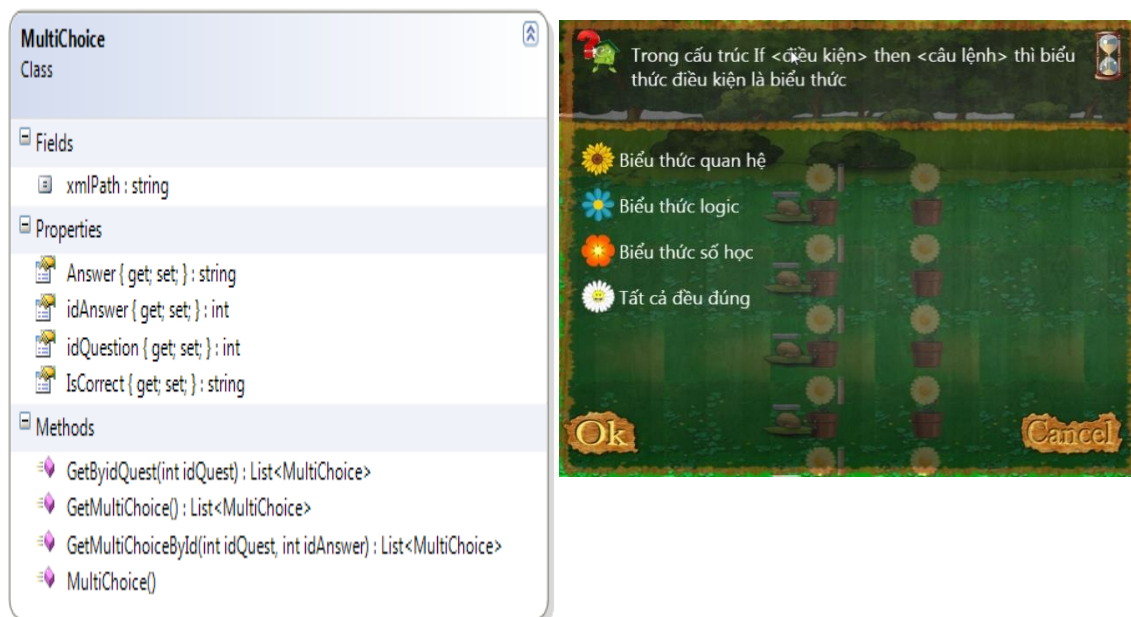
**✚ Lớp Scene\_Question:** Cấu hình loại câu hỏi, mức độ khó theo tùy màn chơi



Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idScene	int	public	Mã Scene	
idLevel	int	public	Mã mức độ khó	
idType	int	public	Mã loại câu hỏi	
typeofQuestion	QuestionType	public	Loại câu hỏi	
scene	Scene	public	Scene tương ứng	
level	Level	public	Mức độ khó của câu hỏi	
Questions	List<Question>	public	Danh sách những câu hỏi được lọc theo scene	
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
QuestionScenes	public	NULL	List<Scene_Question>	Lấy tất cả những cấu hình câu hỏi
QuestionSceneBySceneID	public	int id	QuestionType	Lấy ra những cấu hình về câu hỏi (mức độ, loại câu hỏi) theo Scene

**Hình 2.27: Lớp Scene\_Question**

✚ MultiChoice:Xử lý việc lấy câu hỏi dạng nhiều đáp án.

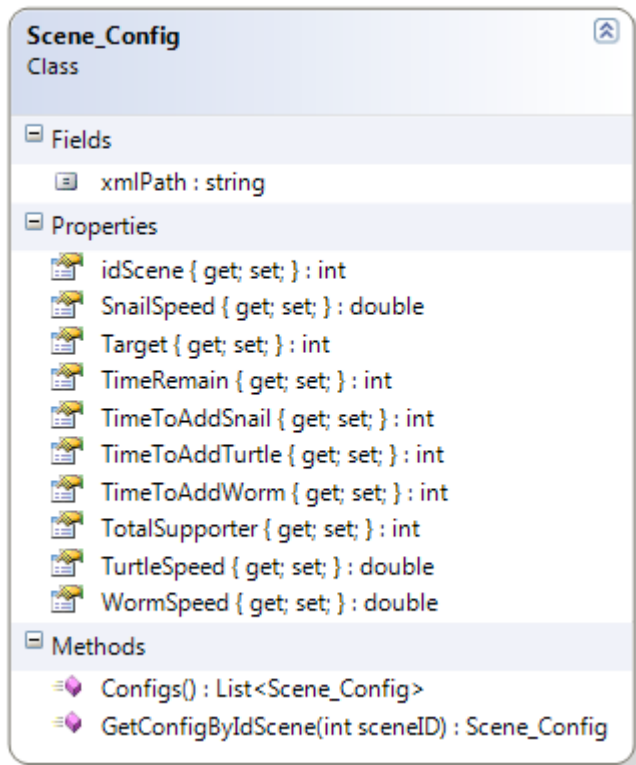


Hình 2.28:MultiChoice

Mô tả

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idAnswer	Int	Public	Mã đáp án	
Answer	string	public	Đáp án của câu hỏi	
IsCorrect	string	public	Trả về kết quả đúng hay sai tương ứng với đáp án của câu hỏi này	
idQuestion	int	public	Mã câu hỏi	
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
GetMultiChoice	public	NULL	List<MultiChoice>	Trả về tất cả những câu hỏi nhiều lựa chọn
GetById	public	int idQuest int idAnswer	List<MultiChoice>	Trả về câu hỏi nhiều lựa chọn tương ứng với mã câu hỏi nhiều lựa chọn
GetByQuestionIDS	public	NULL	List<MultiChoice>	Trả về những câu hỏi nhiều lựa chọn tương ứng với mã câu hỏi

- ✚ Lớp Scene\_Config:Giúp lấy thông tin cấu hình của màn chơi như tốc độ di chuyển của nhân vật đối kháng,số câu trả lời đúng cần đạt để qua màn.

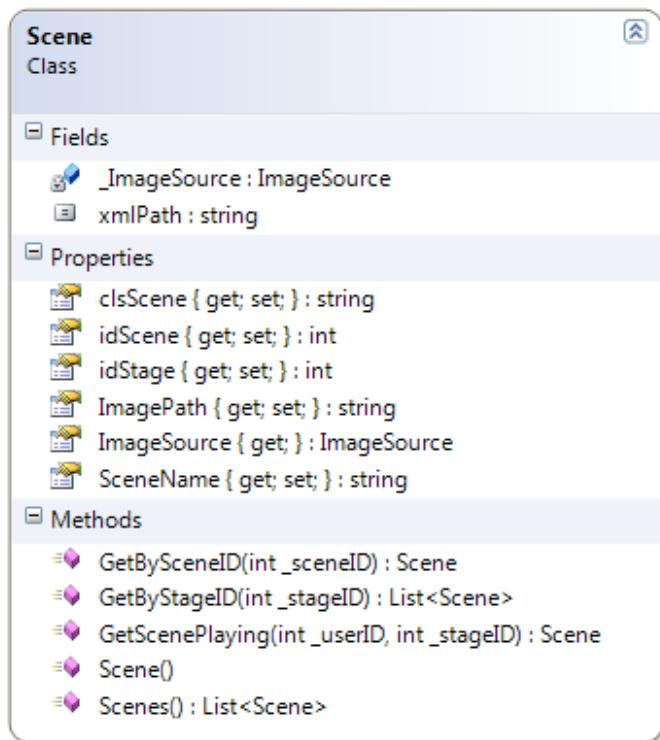


**Hình 2.29: Lớp Scene\_Config**

**Mô tả**

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idScene	int	public	Mã Scene	
TotalSupporter	int	public	Số lượng nhân vật hỗ trợ để tạo nhanh cây bông	
TimeRemain	int	public	Thời gian tạm dừng trò chơi khi nhấp vào đồng hồ cát	
TimeToAddSnail	int	public	Thời gian xuất hiện ốc sên	
TimeToAddWorm	int	public	Thời gian xuất hiện con Sâu	
TimeToAddTurtle	int	public	Thời gian xuất hiện con Rùa	
Target	int	public	Số câu hỏi đúng cần đạt để qua một mức độ của cảnh chơi	
WormSpeed	double	public	Tốc độ di chuyển của con Sâu	
SnailSpeed	double	public	Tốc độ ốc sên	
TurtleSpeed	double	public	Tốc độ di chuyển của con rùa	
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Configs	public	NULL	List<Scene_Question>	Lấy tất cả những câu hình
GetConfigByIdScene	public	int sceneID	Scene_Config	Trả về những câu hình tương ứng với Scene

- ✚ Lớp Scene: Lấy thông tin về màn chơi, thông tin về những màn chơi người chơi đã vượt qua., lấy những Scene theo StageID.



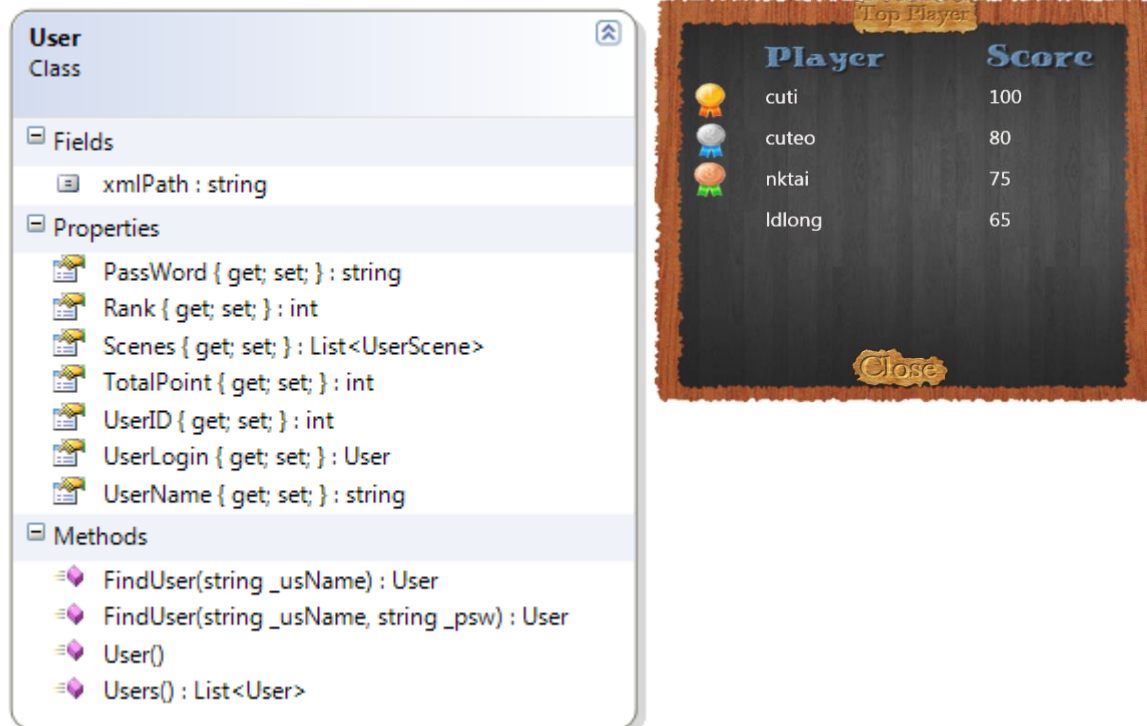
**Hình 2.30: Lớp Scene**

Mô tả:

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
idScene	int	public	Mã Scene	
idStage	int	public	Mã Stage	
clsScene	string	public	Lớp tương ứng với scene	
SceneName	string	public	Tên Scene	
ImagePath	string	public	Đường dẫn hình	
ImageSource	ImageSource	public	Hình đại diện cho Scene	
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Scenes	public	NULL	List<Scene>	Lấy tất cả những Scene
GetScenePlaying	public	int userID, int stageID	Scene	Trả về Scene đang chơi
GetBySceneID	public	int sceneID	Scene	Lấy thông tin Scene tương ứng với SceneID
GetByStageID	public	int stageID	List<Scene>	Lấy những Scene tương ứng với StageID



- ✚ Lớp User:Giải thích:Xử lý liên quan đến người chơi như cập nhật điểm,cảnh đang chơi...



**Hình 2.31: Lớp User**

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
xmlPath	const string	private	Đường dẫn file xml	
UserID	int	public	Mã người chơi	
UserName	string	public	Tên người chơi	
PassWord	string	public	Mật khẩu	
TotalPoint	int	public	Tổng điểm	
Rank	int	public	Xếp hạng	
UserLogin	User	public	Trả về Người chơi đang đăng nhập	
ScenePlayed	List<UserScene>	public	Những UserScene đã chơi tương ứng với mã User	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
FindUser	public	string _usName, string _psw	User	Tìm kiếm người chơi theo tên đăng nhập và mật khẩu
Users	public	NULL	List<User>	Lấy tất cả những người chơi
FindUser	public	string _usName	User	Tìm kiếm người chơi theo tên đăng nhập

## 2.4.2 Chi tiết các lớp xử lý màn hình:

# Xử lý màn hình

Mainwindow: Màn hình chính chức năng

Stage: Xử lý liên quan đến việc chọn màn chơi.

Scene\_1: Xử lý màn chơi 1

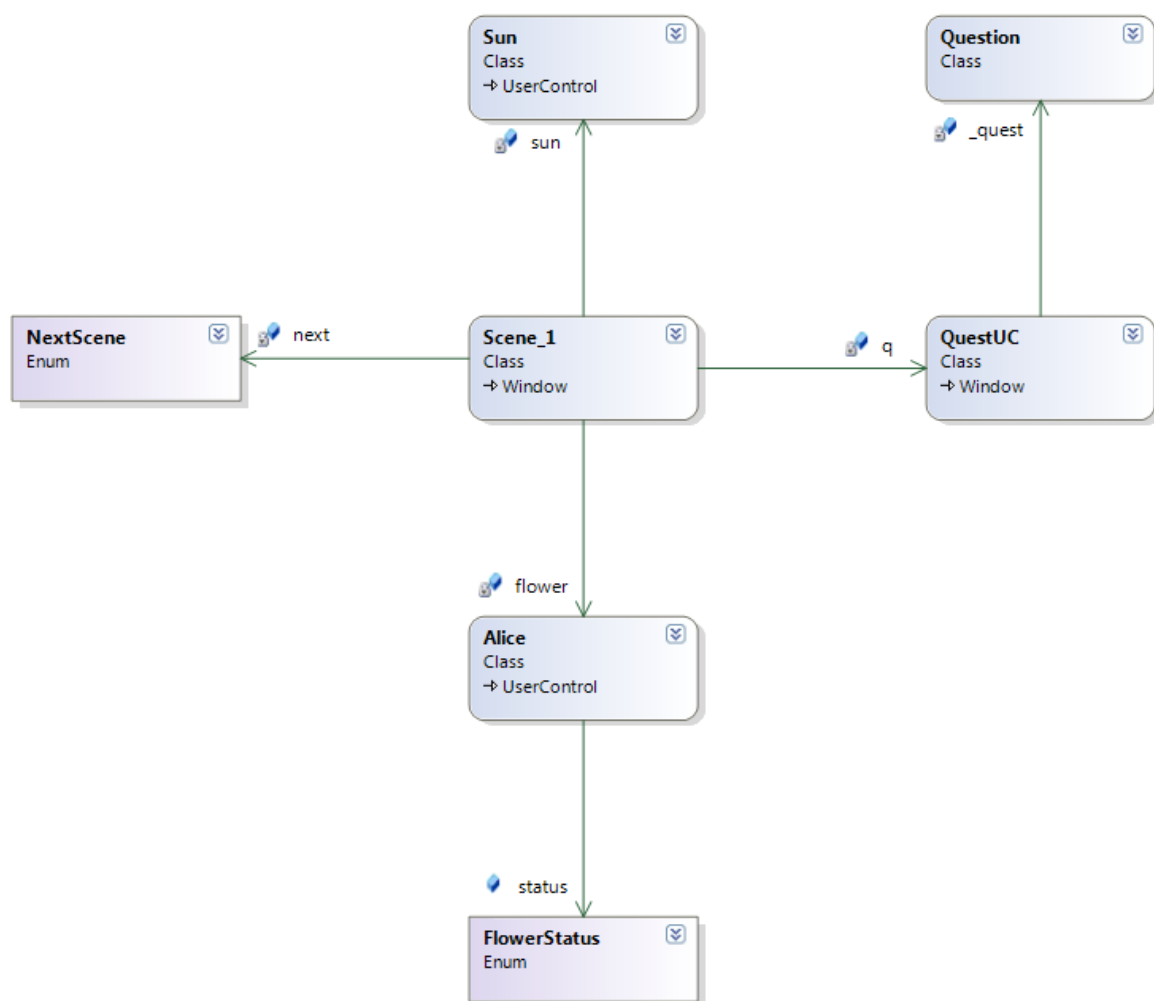
Sene\_2: Xử lý màn chơi 2

Scene\_3: Xử lý màn chơi 3

QuestUC: Xử lý liên quan đến câu hỏi,kiến thức.

Register: Xử lý liên quan đến việc đăng ký tài khoản

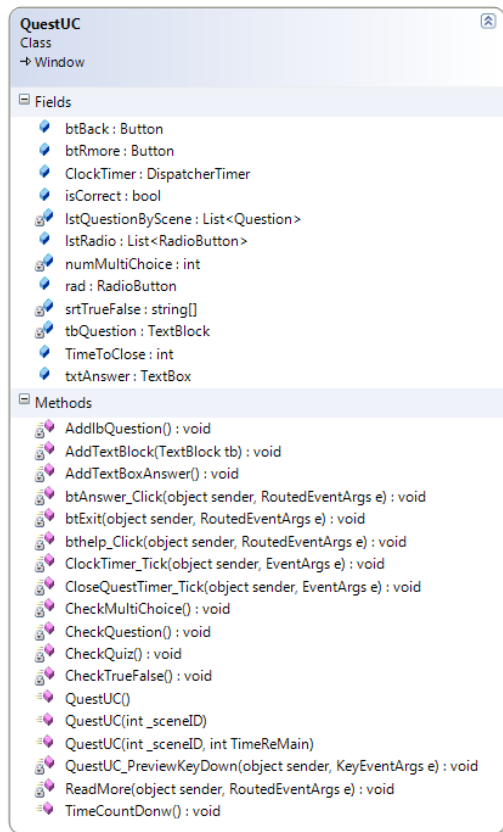
Login: Xử lý liên quan đến việc đăng nhập của người chơi



Hình 2.32: Các lớp xử lý màn hình



## ✚ Lớp QuestUC: Dùng để hiển thị câu hỏi, kiến thức

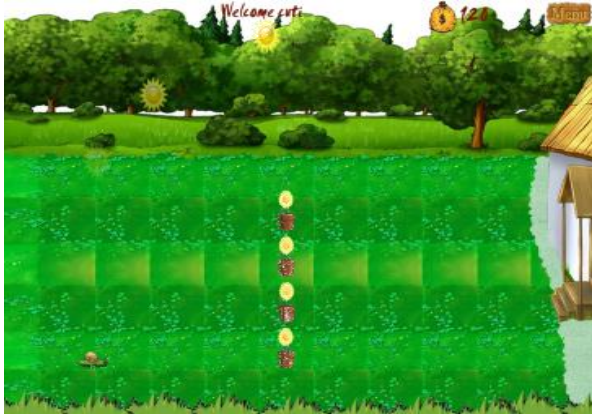


**Hình 2.33: Lớp QuestUC**

Thuộc tính			
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng
btBack	Button	public	Button xử lý việc quay lại màn hình câu hỏi
btReadmore	Button	public	Button xử lý việc đọc chi tiết chủ đề
_quest	Question	public	Câu hỏi tương ứng
IstQuestionByScene	List<Question>	public	Danh sách những câu hỏi tương ứng với Scene
numMultiChoice	int	public	Biến dùng để lưu số câu trả lời dạng nhiều đáp án
isCorrect	bool	public	Biến toàn cục để kiểm tra câu trả lời là đúng hay sai
TimeToClose	int	public	Khoảng thời gian tắt màn hình hiển thị câu hỏi
srtTrueFalse	string[]	public	Mảng lưu hiển thị đúng hay sai đối với câu hỏi dạng đúng sai
ClockTimer	DispatcherTimer	public	Timer thời gian đóng màn hình câu hỏi
IstRadio	List<RadioButton>	public	Mảng những Radio button hiển thị câu trả lời đối với câu hỏi dạng nhiều đáp án
rad	RadioButton	public	Radio button là một đáp án trong câu hỏi Multichoice
txtAnswer	TextBox	public	TextBlock hiển thị để người dùng gõ đáp án
tbQuestion	TextBlock	public	Textblock hiển thị nội dung câu hỏi

Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
QuestUC	public	NULL	QuestUC	Hàm hiển thị màn hình hiển thị câu hỏi
QuestUC	public	int_sceneID	QuestUC	Hàm hiển thị màn hình câu hỏi tương ứng với Scene
QuestUC	public	int_sceneID, int TimeReMain	QuestUC	Hàm khởi tạo Màn hình hiển thị câu hỏi tương ứng với Scene và khoảng dừng thời gian
ClockTimer_Tick	private	object sender, EventArgs	void	Đồng hồ đếm ngược thời gian trả lời câu hỏi
CloseQuestTimer_Tick	private	object sender, EventArgs	void	Tự động tắt màn hình hiển thị câu hỏi khi hết thời gian
QuestUC_PreviewKeyDown	private	object sender EventArgs	void	Hàm xử lý sự kiện khi người dùng sử dụng bàn phím
AddbQuestion	private	NULL	void	Chèn Label chứa câu hỏi
AddTextBoxAnswer	private	NULL	void	Chèn TextBox để gõ đáp án đối với câu điền khuyết
btAnswer_Click	private	object sender, EventArgs	void	Sử lý khi Click vào nút Trả lời câu hỏi
CheckMultiChoice	private	NULL	void	Kiểm tra đúng sai đối với câu hỏi nhiều lựa chọn
CheckQuiz	private	NULL	void	Hàm kiểm tra kết quả đúng sai đối với câu hỏi dạng Quiz
CheckTrueFalse	private	NULL	void	Hàm kiểm tra đúng sai đối với câu hỏi dạng đúng sai
CheckQuestion	private	NULL	void	Hàm kiểm tra câu hỏi
btExit	private	object sender, EventArgs	void	Tắt Màn hình hiển thị câu hỏi
TimeCountDown	private	object sender, EventArgs	void	Khởi tạo sự kiện đếm ngược thời gian
bthelp_Click	private	object sender, EventArgs	void	Hiện thị PI
ReadMore	private	NULL	void	Mở file PDF chứa nội dung chi tiết.
AddTextBlock	private	NULL	void	Chèn TextBlock chứa nội dung

- + **Lớp Scene\_1:**Giải thích:Dùng để hiển thị và xử lý cảnh chơi 1 như hiển thị nhân vật,xử lý qua màn chơi.



**Hình 2.34: Lớp Scene**

```

Scene_1
Class
  Window
Members
  addFlower(int numRows, int numCol) : void
  addFlowers() : void
  AddFlowerScene_1b() : void
  addSnail() : void
  addText() : void
  bt_Resume_Menu_Click(object sender, RoutedEventArgs e) : void
  btMainMenu_Click(object sender, RoutedEventArgs e) : void
  btMenu_Click(object sender, RoutedEventArgs e) : void
  btNext_Click(object sender, RoutedEventArgs e) : void
  btNextScene_Click(object sender, RoutedEventArgs e) : void
  btPause_Click(object sender, RoutedEventArgs e) : void
  btQuit_Click(object sender, RoutedEventArgs e) : void
  btQuitFail_Click(object sender, RoutedEventArgs e) : void
  btQuitWin_Click(object sender, RoutedEventArgs e) : void
  btRestart_Click(object sender, RoutedEventArgs e) : void
  btRestart_Menu_Click(object sender, RoutedEventArgs e) : void
  btResume_Click(object sender, RoutedEventArgs e) : void
  btStart_Click(object sender, RoutedEventArgs e) : void
  CheckCollision() : void
  ESC_Press(object sender, KeyEventArgs e) : void
  ESC_Unpress(object sender, KeyEventArgs e) : void
  ExtraTime(object sender, MouseButtonEventArgs e) : void
  Flower_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) : void
  FlowerManage() : void
  GameLoop(object sender, EventArgs e) : void
  isCollision(Snail_oc, Alice_flower) : bool
  isSceneComplete : bool
  isStart : bool
  isStopGame : bool
  lstFlower : List<Alice>
  lstSnail : List<Snail>
  lstSun : List<Sun>
  mwd : MainWindow
  numbCorrect : int
  numColumn : int
  numRows : int
  Point : int
  q_Closed(object sender, EventArgs e) : void
  random : Random
  RemoveAllFlower() : void
  RemoveAllGameObject() : void
  RemoveAllOccen() : void
  RemoveAllSun() : void
  RemoveFlower(Alice_flower) : void
  RemoveSnail(Snail_oc) : void
  RemoveSun(Sun_sun) : void
  ResetVariable() : void
  ResumeGame() : void
  ReturnGame(object sender, MouseButtonEventArgs e) : void
  Scene_1()
  ShowThongBao() : void
  snail_a(object sender, EventArgs e) : void
  snail_b(object sender, EventArgs e) : void
  snail_c(object sender, EventArgs e) : void
  SnailSpeed : double
  snailTimer : DispatcherTimer
  soundBoom : SoundPlayer
  soundButton : SoundPlayer
  soundFail : SoundPlayer
  soundScene : SoundPlayer
  StopAllTimer() : void
  StopGame() : void
  StudyPl(object sender, EventArgs e) : void
  sunClick(object sender, MouseButtonEventArgs e) : void
  SunManage() : void
  sunTimer : DispatcherTimer
  Target : int
  tbOcscn : UserControl
  TBOcscn() : void
  tempPoint : int
  TimerSunEvent(object sender, EventArgs e) : void
  TimerSunInit() : void
  TimeToAddSnail : int
  totalFlower : int
  totalSun : int
  updateMoney() : void
  UserID : int
  validNumRowCol() : void
  Window_Closed(object sender, EventArgs e) : void
  Window_Loaded(object sender, RoutedEventArgs e) : void
  
```

<b>Thuộc tính:</b>			
<b>Tên thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Phạm vi</b>	<b>Chức năng</b>
QuestUC	QuestUC	private	form thể hiện câu hỏi và PI
sunTimer	DispatcherTimer	private	Timer liên quan đến mặt trời
snailTimer	DispatcherTimer	private	Timer liên quan đến ốc sên
lstSun	List<Sun>	private	Danh sách các mặt trời xuất hiện
UserID	int	private	Mã người chơi đăng nhập
Point	int	private	Tổng điểm người chơi đạt được
tempPoint	int	private	Điểm theo Scene của người chơi
isStopGame	bool	private	Load Scene thì Stop Game chỉ khi nào Bấm vào StartGame thì mới cho bật chuyển động
tbOcsen	UserControl	private	Màn hình hiển thị thông báo ốc sên xuất hiện
totalFlower	int	private	Đếm số bông hoa được tạo ra để xuất hiện thông báo.
isStart	bool	private	Tắt hết tất cả Timer cũng như GameLoop khi chưa bắt đầu chơi
next	NextScene	private	Dùng để quản lý việc quan level mới của người chơi
numbCorrect	int	private	Đếm số câu trả lời đúng
Target	int	private	Số câu hỏi cần để qua 1 level
totalSun	int	private	Tổng số mặt trời trong 1 level
isSceneComplete	bool	private	Kiểm tra scene đã được chơi hay chưa
random	Random	private	Tạo số ngẫu nhiên
numRow	int	private	Số dòng
numColumn	int	private	Số cột
TimeToAddSnail	int	private	Thời gian xuất hiện của nhân vật đôi kháng.
SnailSpeed	double	private	Tốc độ di chuyển của ốc sên
flower	Alice	private	Nhân vật bông hoa
sun	Sun	private	Mặt trời
snail	Snail	private	Con sên



Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Scene_1	private	NULL	Scene	Khởi tạo liên quan đến màn chơi 1
ESC_Unpress	private	object sender, KeyEventArgs e	void	Xử lý khi nhả nút ESC
ESC_Press	private	object sender, KeyEventArgs e	void	Xử lý khi nhấn vào nút ESC
addFlowers	private	NULL	void	Chèn nhanh 5 bông hoa
addFlower	private	NULL	void	Chèn một bông hoa tại vị trí dòng và cột tương ứng
validNumRowCol	private	NULL	void	Số dòng và cột quy định là 5x6
AddFlowerScene_1b	private	NULL	void	Chèn nhanh 10 bông hoa đối với level 2 của màn chơi
flower_MouseLeftButtonDown	private	object sender, MouseButtonEventArgs e	void	Hiện thị câu hỏi và xử lý việc trả lời câu hỏi
ReturnGame	private	object sender, MouseButtonEventArgs e	void	Tiếp tục chơi game khi màn hình hiện thị câu hỏi, PI bị đóng
ExtraTime	private	object sender, MouseButtonEventArgs e	void	Dừng chuyển động của các nhân vật đối kháng để người chơi trả lời câu hỏi
QuestUC_Closed	private	object sender, EventArgs e	void	Tiếp tục chơi khi màn hình câu hỏi tắt.
StudyPI	private	object sender, EventArgs e	void	Xử lý khi màn hình PI xuất hiện
addSnail	private	NULL	void	Khởi tạo Timer Ốc sên
snail_a	private	object sender, EventArgs e	void	Sự kiện khi ốc sên xuất hiện ở level a
snail_b	private	object sender, EventArgs e	void	Sự kiện khi ốc sên xuất hiện ở level b
snail_c	private	object sender, EventArgs e	void	Sự kiện khi ốc sên xuất hiện ở level c
updatePoint	private	NULL	void	Cập nhật hiển thị điểm số lên màn hình
GameLoop	private	object sender, EventArgs e	void	Xử lý toàn bộ những sự kiện trong Game
FlowerManage	private	NULL	void	Xuất hiện thông báo
RemoveAllFlower	private	NULL	void	Xóa tất cả ốc sên trong game
RemoveAllSun	private	NULL	void	Xóa tất cả bông hoa
btNextScene_Click	private	object sender, RoutedEventArgs e	void	Xử lý khi người chơi hoàn thành màn chơi
Window_Loaded	private	object sender, RoutedEventArgs e	void	Khi chưa xuất hiện thông báo hướng dẫn thì không chơi game
StopAllTimer	private	NULL	void	Tắt tất cả các Timer trong màn chơi
btStart_Click	private	object sender, RoutedEventArgs e	void	Gọi hàm xử lý bắt đầu chơi game
btMainMenu_Click	private	object sender, RoutedEventArgs e	void	Quay về màn hình chính
btRestart_Menu_Click	private	object sender, RoutedEventArgs e	void void	Gọi hàm xử lý chơi lại từ đầu
Window_Closed	private	object sender, RoutedEventArgs e	void	Đóng màn hình game và lưu điểm của người chơi vào file

Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
ShowThongBao	private	NULL	void	Hiển thị thông báo ốc sên sắp xuất hiện.
SunManage	private	NULL	void	Sử lý liên quan đến mặt trời
isCollision	private	Snail_oc, Alice_flower	void	Kiểm tra ốc sên và bông hoa có tiếp xúc nhau hay không
CheckCollision	private	NULL	void	Sử lý va chạm giữa ốc sên và cây bông
RemoveSnail	private	Snail_oc	void	Xóa ốc sên khỏi vùng nhớ và màn hình trò chơi
RemoveFlower	private	Alice_flower	void	Xóa bông hoa
RemoveSun	private	Sun_sun	void	Xóa mặt trời khỏi vùng nhớ và màn hình trò chơi
TimerSunInit	private	NULL	void	Khởi tạo mặt trời xuất hiện
TBOcsen	private	NULL	void	Hiển thị thông báo ốc sên sắp xuất hiện
TimerSunEvent	private	object sender, EventArgs e	void	Tạo mặt trời ở một vị trí bất kỳ
sunClick	private	object sender, MouseButtonEventArgs e	void	Tạo nhanh 1 cây bông khi click vào mặt trời
btPause_Click	private	object sender, RoutedEventArgs e	void	Xử lý tạm dừng trò chơi khi người dùng click vào nút Pause
ResumeGame	private	NULL	void	Xử lý tiếp tục chơi game
btResume_Click	private	object sender, RoutedEventArgs e	void	Gọi hàm xử lý tiếp tục chơi game
StopGame	private	object sender, RoutedEventArgs e	void	Dừng trò chơi
btMenu_Click	private	object sender, RoutedEventArgs e	void	Gọi Dừng trò chơi
btquit_Click	private	object sender, RoutedEventArgs e	void	Thoát khỏi trò chơi
bt_Resume_Menu_Click	private	object sender, RoutedEventArgs e	void	Xử lý khi click vào button Resume.
btQuitWin_Click	private	object sender, RoutedEventArgs e	void	Thoát khỏi trò chơi
btQuitFail_Click	private	object sender, RoutedEventArgs e	void	Chơi lại Level bị thua.
btnext_Click	private	object sender, RoutedEventArgs e	void	Sử lý khi người chơi hoàn thành một level của màn chơi
ResetVariable	private	NULL	void	Khởi tạo giá trị cho các thông số của màn chơi về giá trị mặc định
btRestart_Click	private	object sender, RoutedEventArgs e	void	Gọi hàm xử lý chơi lại
RemoveAllGame Object	private	NULL	void	Chơi lại dung level bị thua
RemoveAllOccen	private	NULL	void	Xóa tất cả nhân vật trong game

### 2.4.3 Chi tiết các lớp xử lý đối tượng:

# Xử lý đối tượng

Turtle: Xử lý liên quan đến nhân vật con rùa .

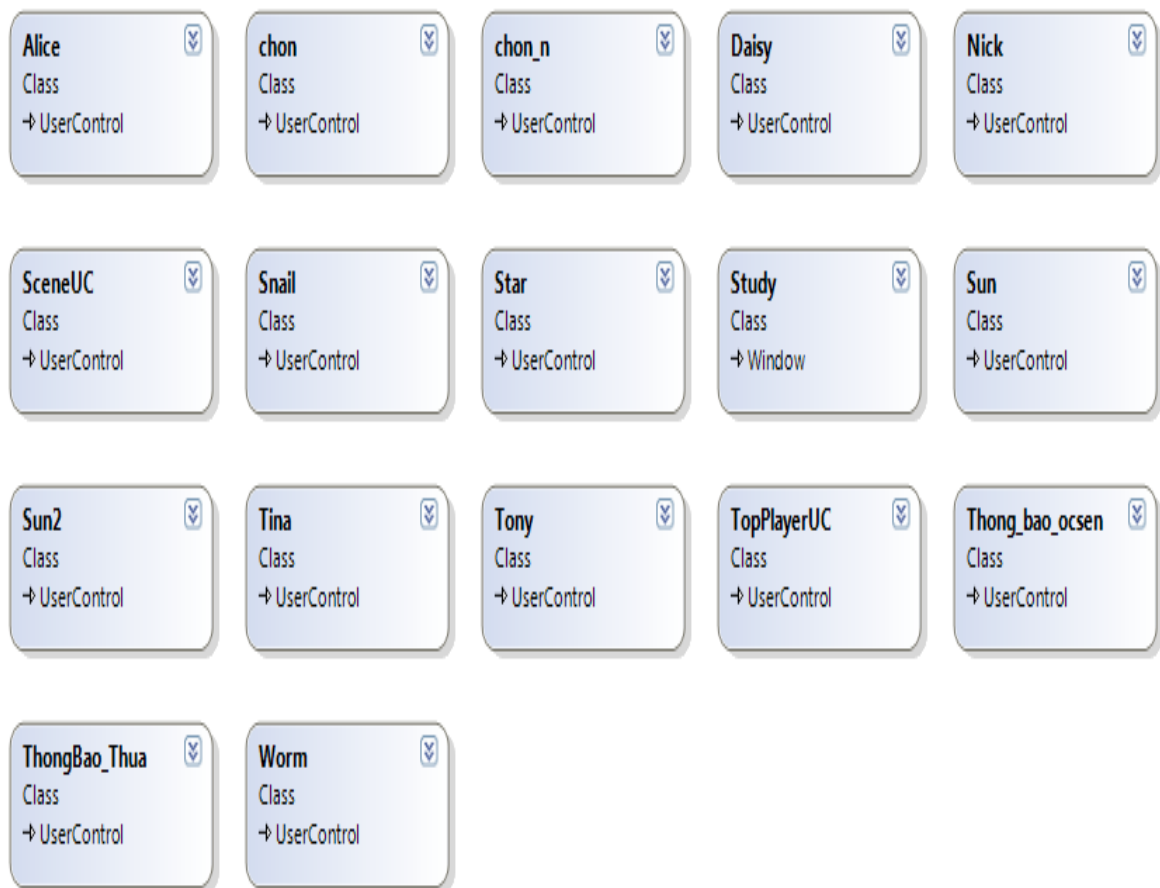
Snail: Xử lý liên quan đến nhân vật con ốc sên

Worm: Xử lý liên quan đến nhân vật con sâu

Flower: Xử lý liên quan đến những bông hoa


Sun: Xử lý liên quan đến mặt trời

Star: Xử lý liên quan đến ngôi sao.



**Hình 2.35: Các lớp xử lý đối tượng**

Vì các lớp này trong xử lý đối tượng tương tự nhau nên chỉ mô tả một lớp đại diện là lớp bông hoa:



**Alice**  
Class  
→ UserControl

**Fields**

- Blood : double
- isActive : bool
- sb : Storyboard
- status : FlowerStatus
- tImageChange : Dis...
- vitri : int

**Properties**

- BoundsRect { get; } : ..
- Location { get; } : Po..
- Size { get; } : Size
- X { get; set; } : double
- Y { get; set; } : double

**Methods**

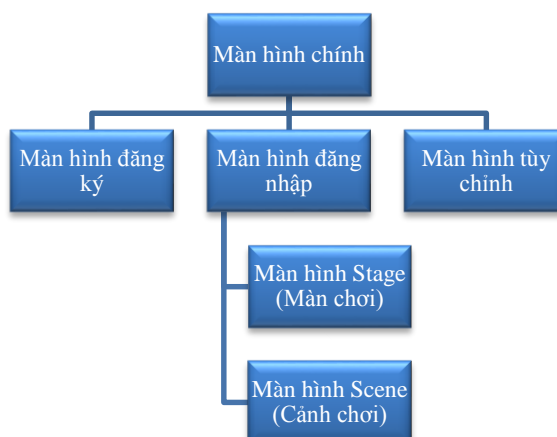
- Alice()
- Alice(Point \_initPoint)
- Correct(object send...
- False(object sender, ...
- Normal(object send...
- PlayAnimation(strin...
- StopAnimation(strin..
- UpdatePrBlood() : v...

Thuộc tính				
Tên thuộc tính	Kiểu dữ liệu	Phạm vi	Chức năng	
i	int	private	Biên dùng để biết hình thứ i của bông hoa	
Blood	Double	public	Máu của bông hoa	
status	FlowerStatus	public	Cho biết trạng thái của bông hoa	
tImageChange	DispatcherTimer	private	Timer dùng để sử dụng chuyển động	
Location	Point	public	Vị trí của bông hoa trên màn hình	
Size	Size	public	Kích thước của bông hoa	
BoundsRect	Rect	public	Khung hình chữ nhật bao quanh bông hoa	
X	double	public	Tọa độ X	
Y	double	public	Tọa độ Y	
Phương thức				
Tên phương thức	Phạm vi	Tham số	Kiểu dữ liệu trả về	Mô tả
Alice	public	NULL	void	Khởi tạo bông hoa
UpdatePrBlood	public	NULL	void	Cập nhật ProgressBar hiển thị máu
Normal	public	object sender, EventArgs e	void	Thay đổi hình dáng chuyển động của bông hoa ở trạng thái bình thường
False	public	object sender, EventArgs e	void	Thay đổi hình dáng chuyển động của bông hoa khi trả lời sai
Correct	public	object sender, EventArgs e	void	Chạy chuyển động hình dáng bông hoa khi trả lời đúng
PlayAnimation	public	string _sbName	void	Chạy chuyển động
StopAnimation	public	string sbName	void	Ngừng chuyển động

**Hình 2.36: Lớp Alice (bông hoa)**

## 2.5 Thiết kế giao diện:

Sơ đồ các màn hình chính trong trò chơi:



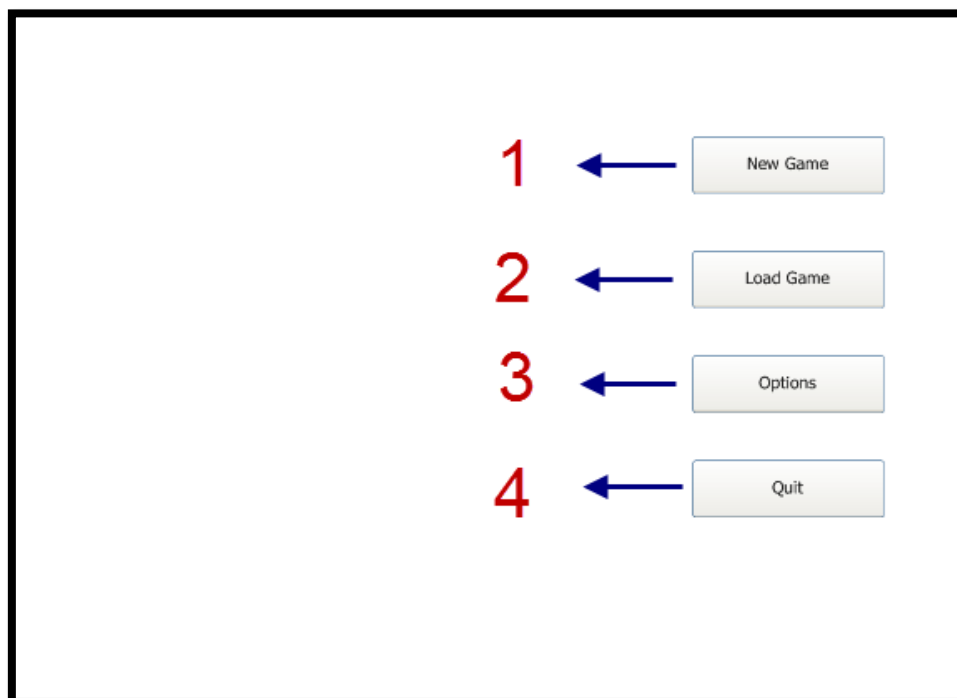
**Hình 2.37: Sơ đồ màn hình chính**



Mô tả chi tiết một số màn hình quan trọng:

### 2.5.1 Màn hình chính:

Được load lên khi người dùng chạy game:



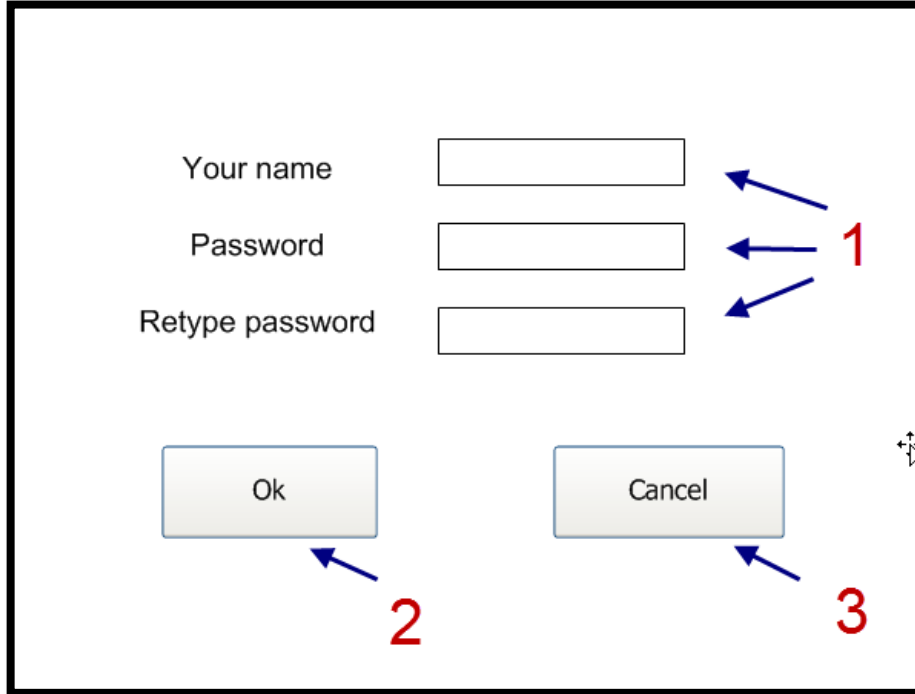
Hình 2.38:Màn hình chính

Mô tả các phần control trong màn hình:

STT	Tên control	Mô tả
1	New game	Bắt đầu một game mới, yêu cầu người chơi phải tạo một tài khoản ( <b>Màn hình đăng ký</b> )
2	Load game	Có chức năng là cho biết thông tin về các cảnh chơi (Scene) mà người chơi đã đi qua. Để thực hiện chức năng này người chơi phải đăng nhập ( <b>Màn hình đăng nhập</b> )
3	Options	Các tùy chỉnh cho trò chơi . ( <b>Màn hình tùy chỉnh</b> )
4	Quit	Thoát khỏi trò chơi.

### 2.5.2 Màn hình đăng ký:

Đây là màn hình sau khi người chơi nhấp vào nút lệnh **NEW GAME** ở Màn hình chính.



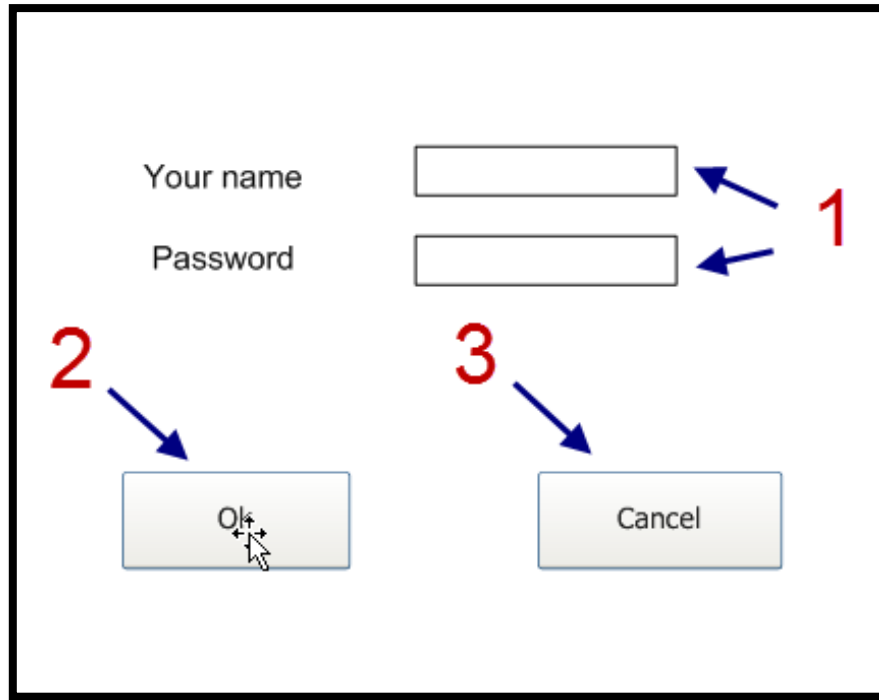
Hình 2.39:Màn hình đăng ký

#### Mô tả các phần control trong màn hình:

STT	Tên control	Mô tả
1	Text box Your name, Password, Retype password	Điền thông tin người chơi.
2	Ok	Thực hiện chức năng đăng ký. Nếu đăng ký thành công sẽ vào <b>Màn hình Stage (Màn chơi)</b> , còn khi đăng ký sai một màn hình thông báo xuất đăng ký chưa thành công
3	Cancel	Hủy chức năng đăng ký.

### 2.5.3 Màn hình đăng nhập:

Có nhiệm vụ kiểm tra thông tin người chơi có phải là thành viên hay không. Nếu đúng là thành viên sẽ cho phép người chơi truy cập vào màn hình Stage, còn nếu không sẽ thông báo lỗi đăng nhập.



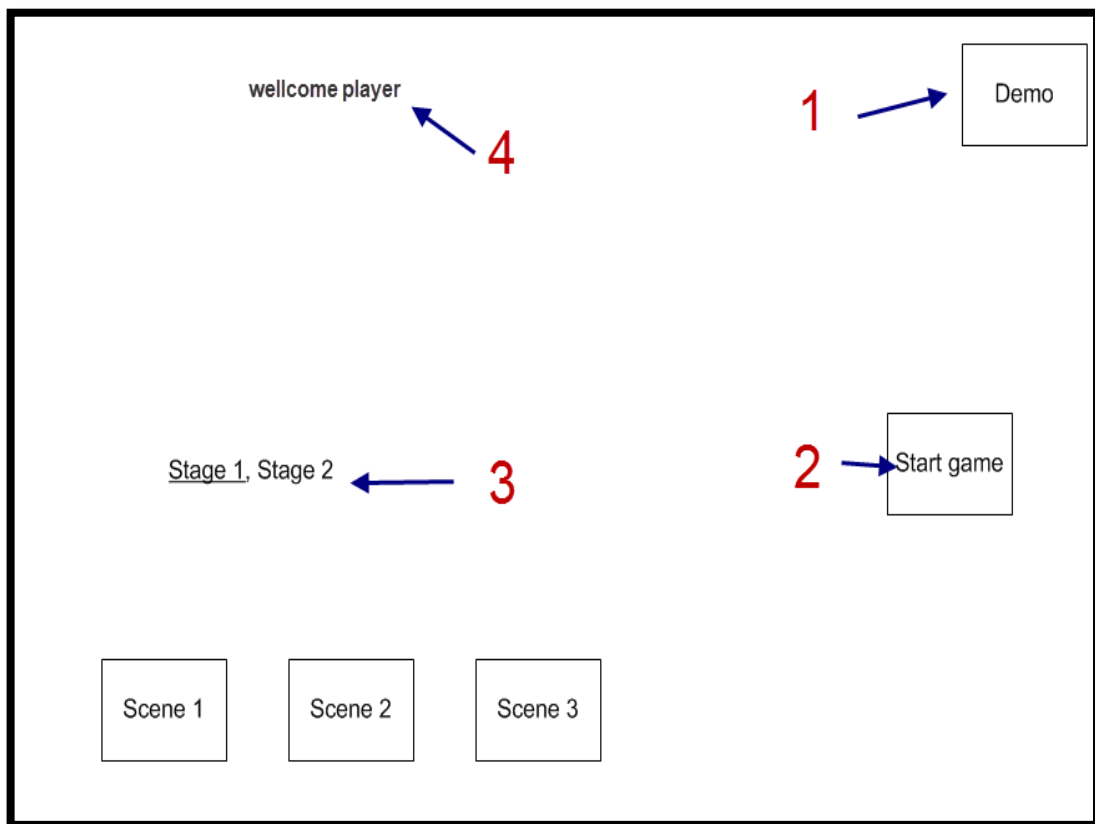
Hình 2.40:Màn hình đăng nhập

#### Mô tả các phần trong màn hình:

STT	Tên control	Mô tả
1	Textbox Your name, Password	Điền thông tin đăng nhập
2	Ok	Thực hiện chức năng đăng nhập. Nếu thành công sẽ vào <u>Màn hình stage</u> (màn chơi), nếu không sẽ thông báo lỗi.
3	Cancel	Hủy chức năng đăng nhập.

### 2.5.4 Màn hình Stage (Màn chơi):

Xuất hiện sau khi người chơi đăng nhập hoặc đăng ký thành công.



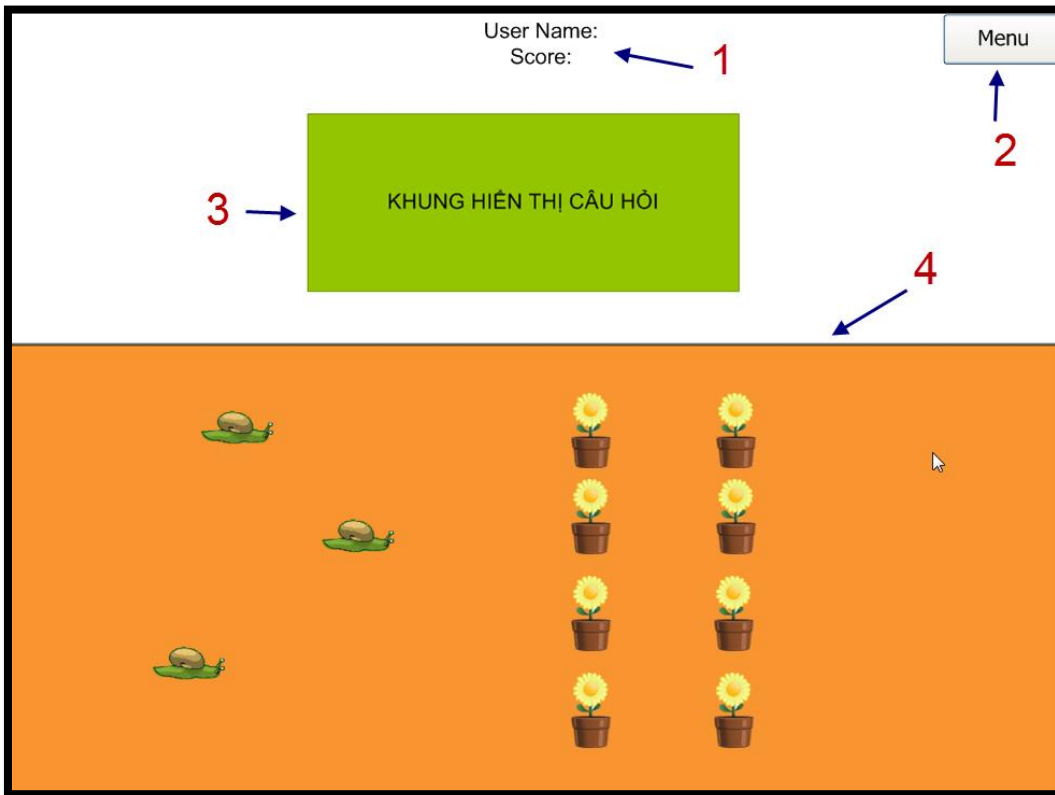
Hình 2.41:Màn hình Stage

#### Mô tả các phần trong màn hình:

STT	Tên control	Mô tả
1	Button Demo	Xem video về giới thiệu game.
2	Button Start game	Cho phép người chơi bắt đầu chơi. Hiển thị màn hình cảnh chơi (Scene)
3	Label Stage 1, Stage 2	Giúp người chơi biết được mình đã chơi qua những màn chơi, cảnh chơi nào.
4	Label Welcome	Hiển thị thông tin player

### 2.5.5 Màn hình 1 scene::

Đây là màn giao diện trong một cảnh chơi (scene) cụ thể.



Hình 2.42:Màn hình Scene

#### Mô tả các phần trong màn hình:

STT	Tên control	Mô tả
1	Label Thông tin người chơi, điểm số.	Hiển thị tên người chơi và điểm số người chơi đạt được
2	Button menu	Cung cấp một số lệnh xử lý trong khi chơi. Bao gồm: tạm dừng game, chơi lại game, thoát game...
3		Nơi hiển thị câu hỏi.
4		Không gian chơi.

CHƯƠNG 3:

**CÀI ĐẶT VÀ THỬ  
NGHIỆM TRÒ CHƠI**

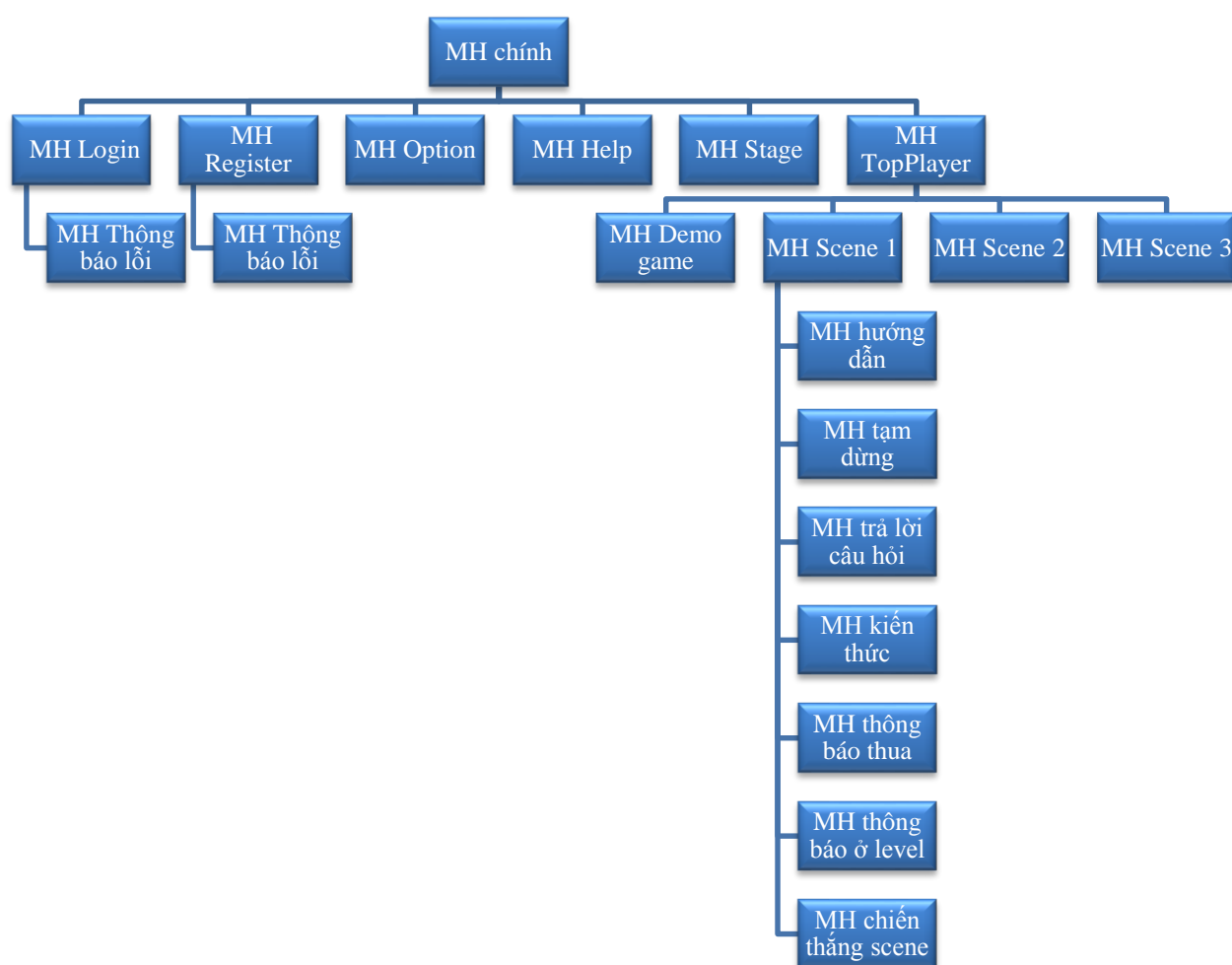
## CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM TRÒ CHƠI

### 3.1 Môi trường phát triển:

- Hệ điều hành: Windows.
- Cơ sở dữ liệu: XML.
- Ngôn ngữ C#, công nghệ WPF ( Windows Presentation Foundation).

### 3.2. Một số màn hình và chức năng minh họa:

Từ những phân tích – thiết kế ban đầu chúng em đã xây dựng được trò chơi với giao diện như sau:



**Hình 3.1:** Sơ đồ thể hiện các màn hình

### 3.2.1 Màn hình vào game:

#### a. Thể hiện và ý nghĩa: Màn hình xuất hiện khi vừa vào game:

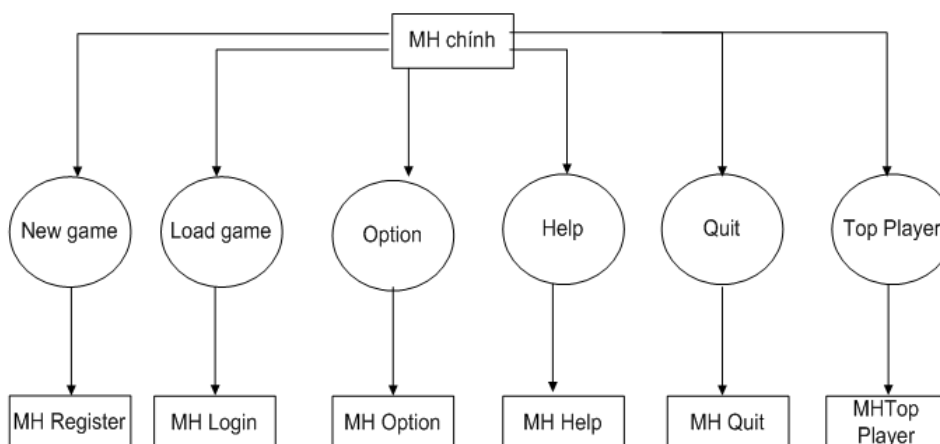


Hình 3.2: Màn hình giao diện chính

#### b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Button New game	Đăng ký một tài khoản mới để vào game.
2		Button Load game	Tiếp tục chơi game (Load game) đối với những người dùng đã có tài khoản.
3		Button Option	Tùy chỉnh âm thanh trong trò chơi.
4		Button Help	Xem giới thiệu về ý tưởng của trò chơi.
5		Button Quit	Thoát game.
6		Button Top Player	Xem những người chơi đạt cao điểm nhất

#### c. Sơ đồ mô tả tình huống sử dụng



Hình 3.3: Sơ đồ tình huống màn hình chính



### 3.2.2 Màn hình Register:

a. **Thể hiện và ý nghĩa:** Màn hình này sẽ được hiển thị khi người dùng có nhu cầu tạo một tài khoản mới để chơi game.

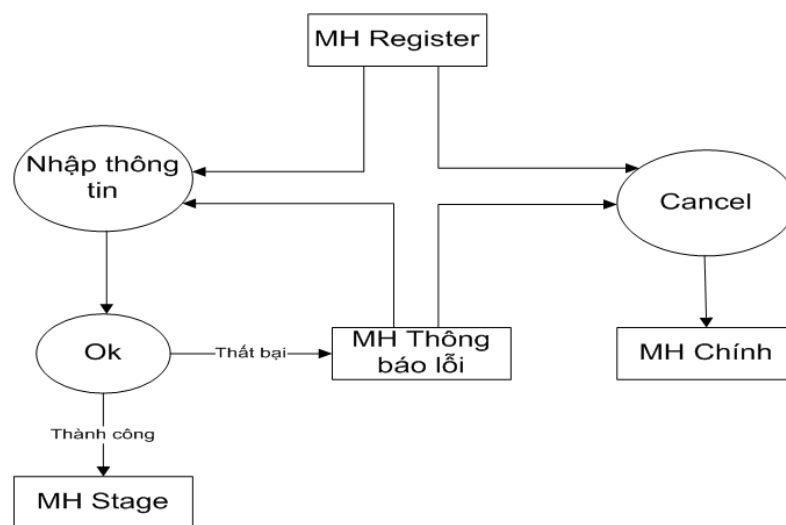


Hình 3.4: Màn hình Register

### b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Textbox thông tin cá nhân.	Dùng để biết thông tin người dùng
2		Button Ok	Kiểm tra thông tin đăng ký của người dùng
3		Button Cancel	Hủy thao tác đăng ký

### c. Sơ đồ mô tả tình huống sử dụng:



Hình 3.5: Sơ đồ tình huống màn hình Register



### 3.2.3 Màn hình Login :

a. Thể hiện và ý nghĩa: Cho phép người dùng đăng nhập vào game.

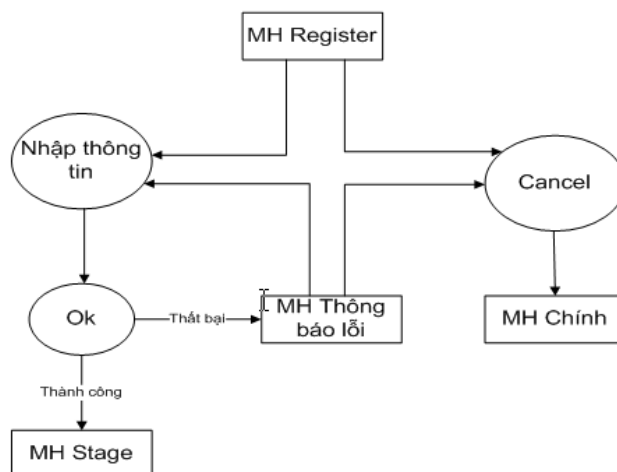


Hình 3.6: Màn hình Login

b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Textbox Your Name, Password	Nhập thông tin người dùng
2		Button Ok	Kiểm tra thông tin login của người dùng.
3		Button Cancel	

c. Sơ đồ mô tả tình huống sử dụng:



Hình 3.7: Sơ đồ tình huống màn hình Login


### 3.2.4 Màn hình hướng dẫn:

a. Thể hiện và ý nghĩa: Xuất hiện khi người dùng muốn xem thông tin về game.

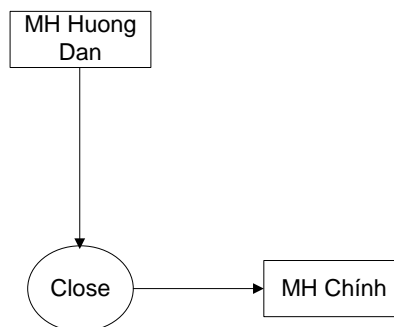


Hình 3.8: Màn hình hướng dẫn

b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Button Close	Tắt màn hình hiển thị thông tin hướng dẫn

c. Sơ đồ mô tả tình huống sử dụng:



Hình 3.9: Sơ đồ tình huống màn hình Hướng dẫn




### 3.2.5 Màn hình xếp hạng người chơi:

a. Thể hiện và ý nghĩa: Xem thông tin về 5 người chơi có điểm số cao nhất.

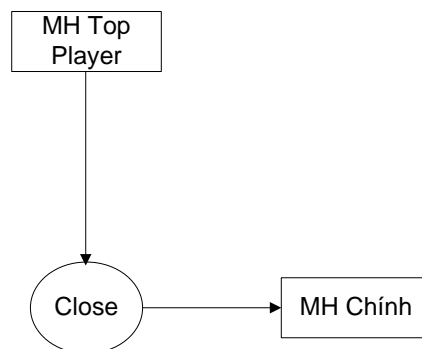


Hình 3.10: Màn hình Top player

c. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Button Close	Tắt màn hình hiển thị top player

d. Sơ đồ mô tả tình huống sử dụng:



Hình 3.11: Sơ đồ tình huống màn hình Top Player

### 3.2.6 Màn hình Stage:

a. **Thể hiện và ý nghĩa:** Hiện thị khi người dùng login tài khoản mới hoặc đăng nhập thành công vào hệ thống.

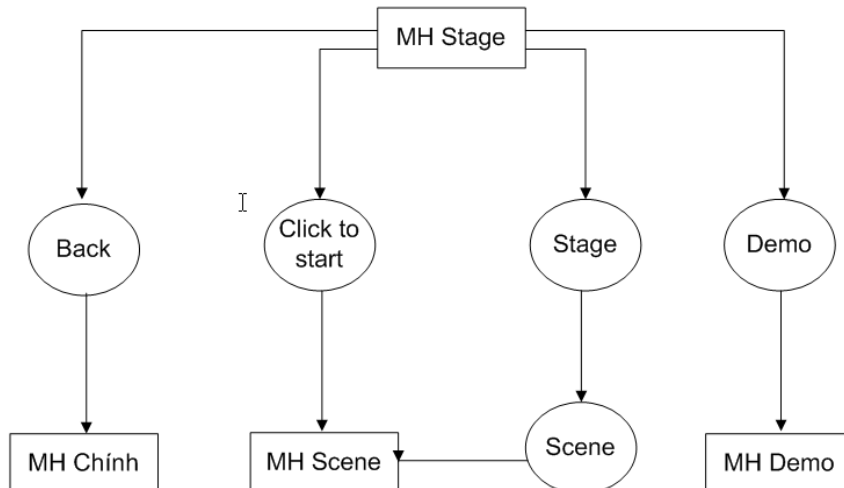


Hình 3.12: Màn hình Stage

b. **Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button Back	Quay trở về Màn hình chính.
2		Button Stage One	Xem thông tin về các màn chơi đã chơi.
3		Button Click to start!	Bắt đầu chơi game hoặc tiếp tục màn chơi ở lần trước.
4			Người dùng cũng có thể chơi lại các cảnh chơi cũ bằng cách nhấn trực tiếp vào màn chơi (scene). Chú ý màn chơi nào đã chơi qua mới được hiển thị, còn màn nào chưa chơi sẽ đánh dấu X.
5		Button Demo	Xem demo về game. Sẽ link đến trang youtube để học sinh xem demo về game.

c. **Sơ đồ mô tả tình huống sử dụng:**

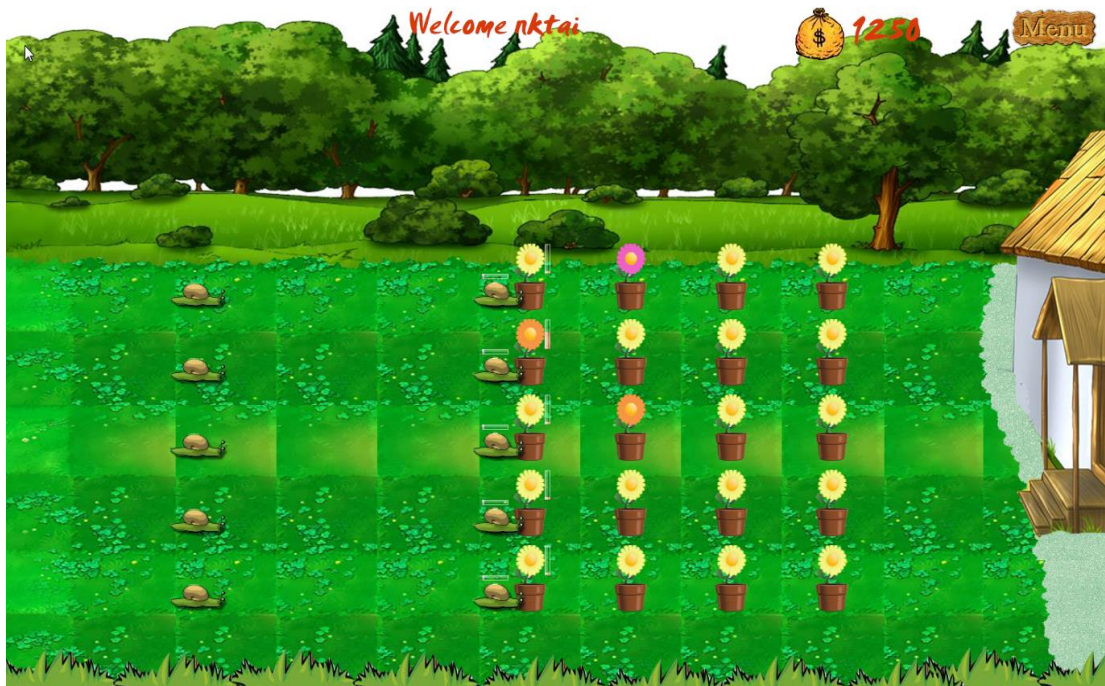


Hình 3.13: Sơ đồ tình huống màn hình Stage

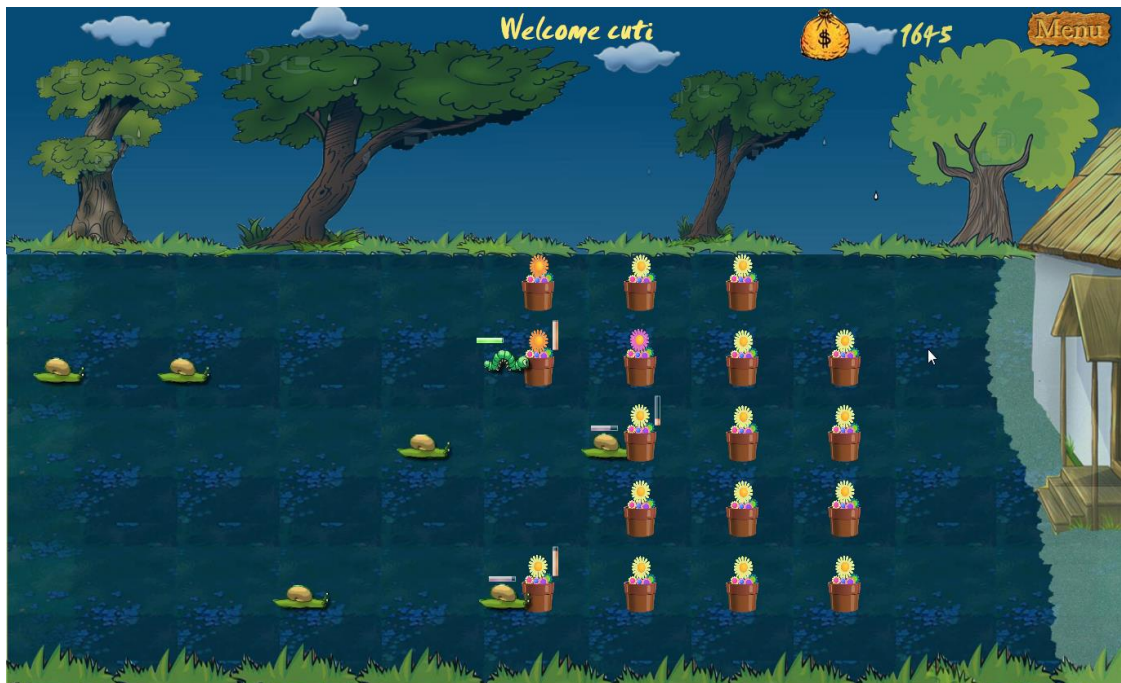


### 3.2.7 Màn hình các scene:

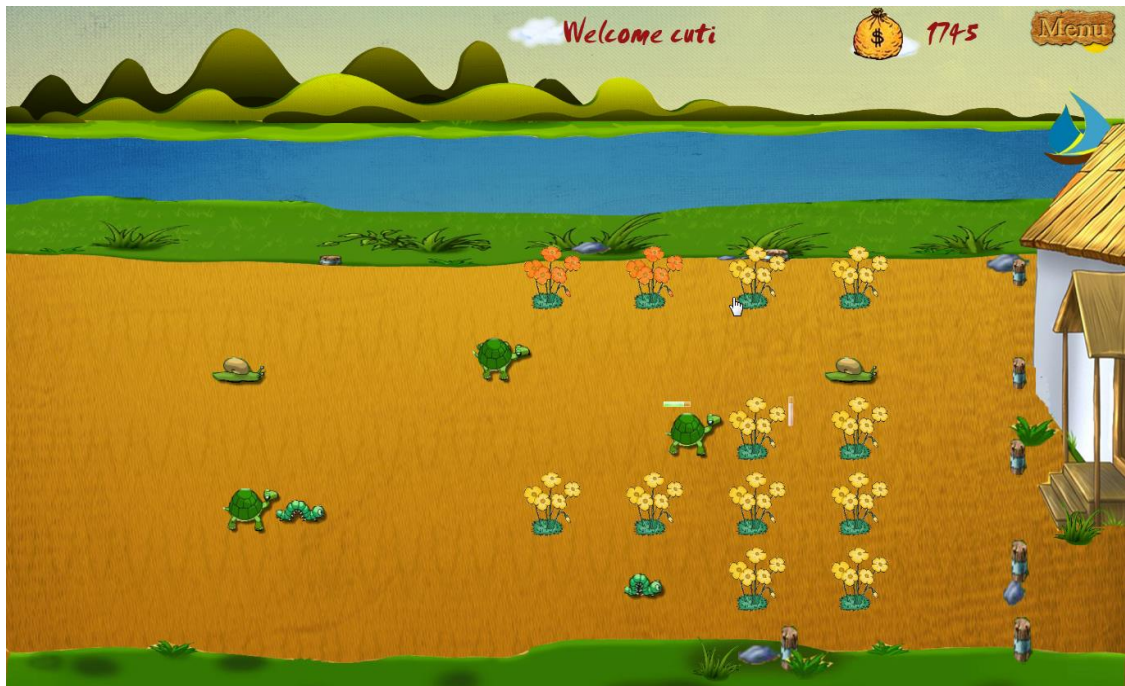
#### a. Thể hiện và ý nghĩa: Giao diện khi chơi game.



Hình 3.14: Màn hình Scene1




Hình 3.15: Màn hình scene 2

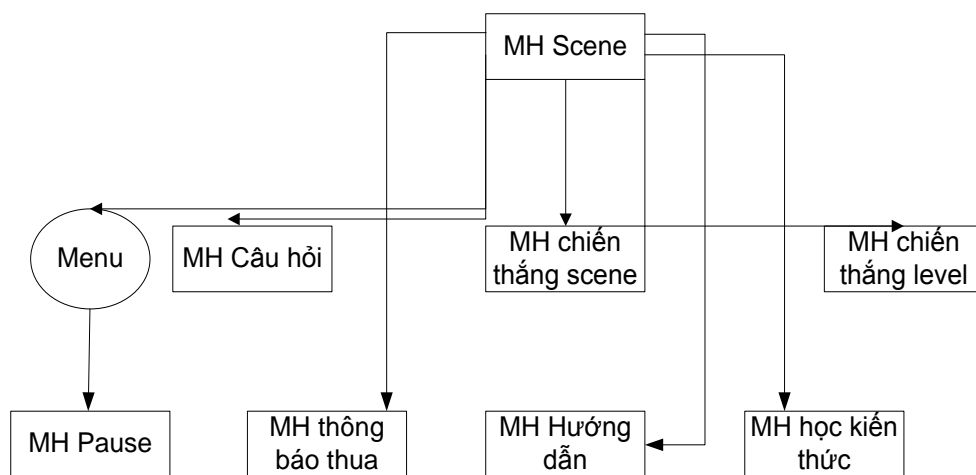


Hình 3.16:Màn hình scene 3

**b. Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button Menu	Cho phép người chơi chọn sử dụng chức năng như: Pause, Exit, Restart, Main Menu

**c. Sơ đồ mô tả tình huống sử dụng:**



Hình 3.17:Sơ đồ tình huống màn hình Scene




### 3.2.8 Màn hình hướng dẫn ở đầu mỗi scene:

a. **Thể hiện và ý nghĩa:** Đưa ra lời hướng dẫn cách chơi trong game.

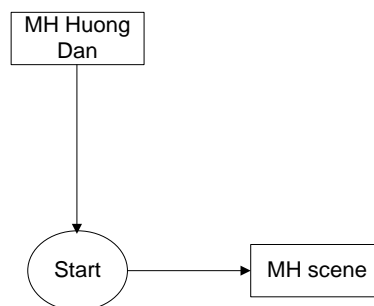


Hình 3.18: Màn hình hướng dẫn đầu mỗi scene

b. **Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button <b>Start</b>	Bắt đầu cuộc hành trình ở mỗi scene.

c. **Sơ đồ mô tả tình huống sử dụng:**



Hình 3.19: Sơ đồ tình huống màn hình Hướng dẫn đầu scene



### 3.2.9 Màn hình khi tạm dừng – Pause:

a. **Thể hiện và ý nghĩa:** Thể hiện trạng thái khi người chơi tạm dừng game.

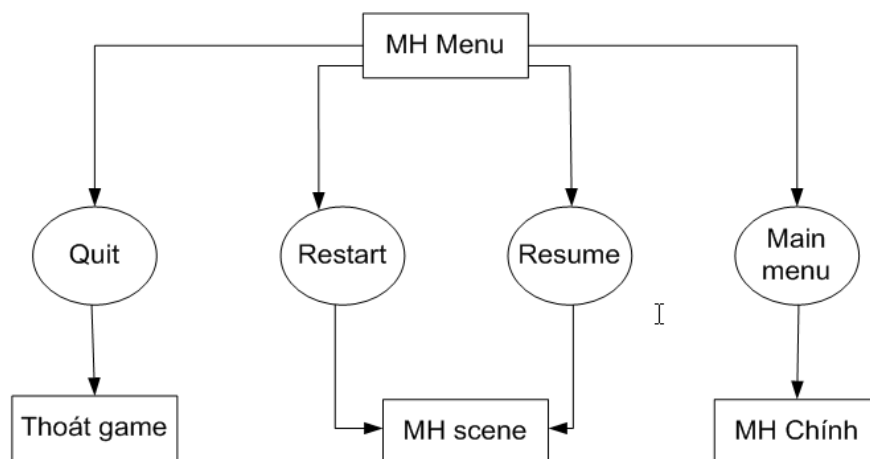


Hình 3.20: Màn hình Pause

b. **Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button <b>Main Menu</b>	Quay về màn hình chính.
2		Button <b>Resume</b>	Tiếp tục chơi game – Resume.
3		Button <b>Restart</b>	Chơi lại – restart lại scene.
4		Button <b>Quit</b>	Thoát khỏi game.

c. **Sơ đồ mô tả tình huống sử dụng:**



Hình 3.21: Sơ đồ tình huống màn hình Pause

### 3.2.10 Màn hình câu hỏi:

a. **Thể hiện và ý nghĩa:** Hiện thị khi người dùng chọn những cây hoa để trả lời câu hỏi.

- **Màn hình câu hỏi dạng nhiều lựa chọn:**



Hình 3.22: Màn hình câu hỏi Multichoice

- **Màn hình câu hỏi dạng đúng sai:**



Hình 3.23: Màn hình câu hỏi True/False



**- Màn hình câu hỏi dạng điền khuyết:**



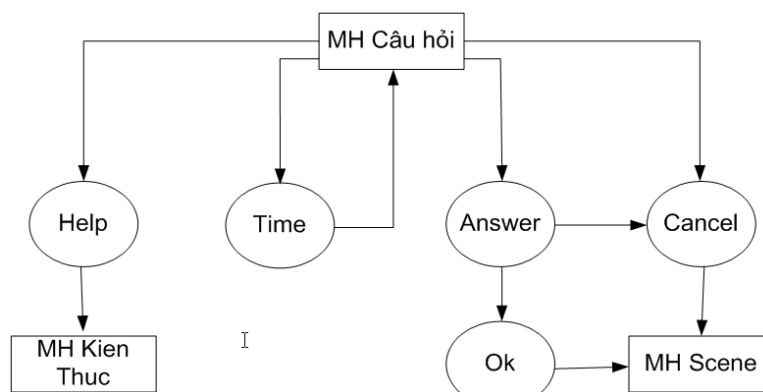
**Hình 3.24:Màn hình câu hỏi Quiz**

**c. Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1			Xem lại kiến thức cũ.
2			Mua thời gian để trả lời câu hỏi.
3		Button Cancel	Không muốn trả lời câu hỏi này.
4		Button Ok	Đồng ý với câu trả lời sau khi chọn đáp án.

Chú ý đối với 1 câu hỏi người dùng chỉ được sử dụng 1 trong 2 trợ giúp: Xem lại kiến thức và mua thời gian trả lời câu hỏi

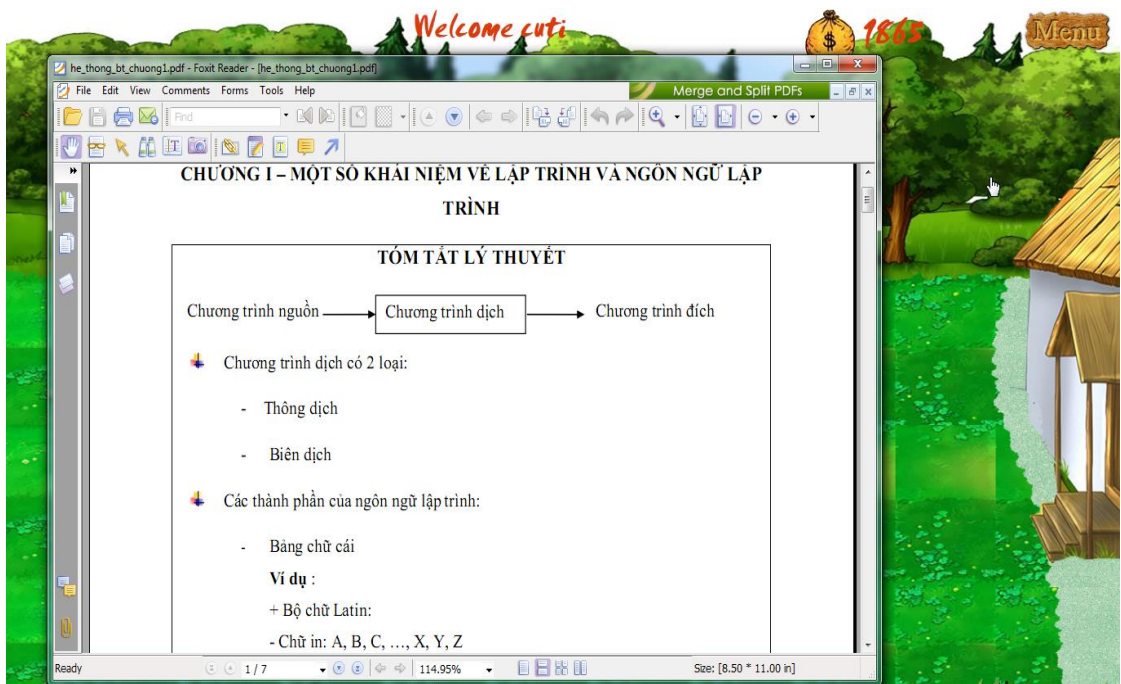
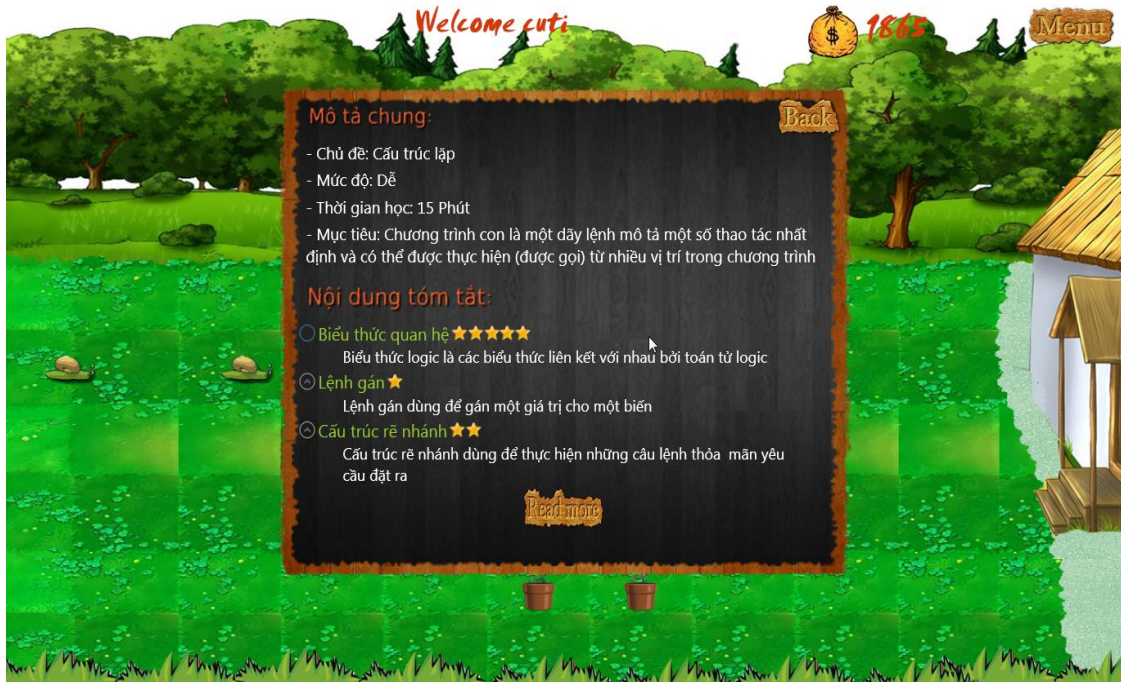
**d. Sơ đồ mô tả tình huống sử dụng:**



**Hình 3.25:Sơ đồ tình huống màn hình câu hỏi**


### 3.2.11 Màn hình xem lại kiến thức:

a. **Thể hiện và ý nghĩa:** Cho phép người chơi học kiến thức liên quan đến các câu hỏi.

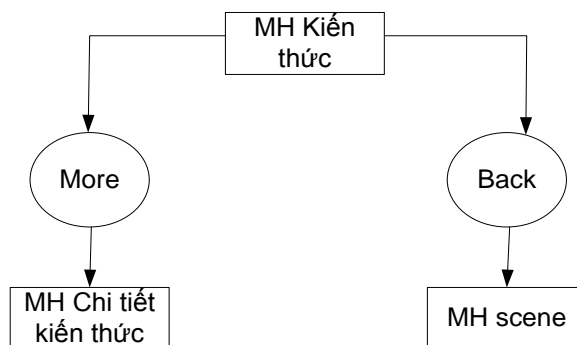


Hình 3.26: Màn hình kiến thức

b. **Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button <b>Back</b>	Quay trở lại câu hỏi và tiếp tục trò chơi.
2		Button Readmore	Để xem chi tiết kiến thức của một chủ đề

c. Sơ đồ mô tả tình huống sử dụng:



**Hình 3.27:**Sơ đồ tình huống màn hình kiến thức



### 3.2.12 Màn hình thông báo thua:

a. Thể hiện và ý nghĩa: Thông báo khi người chơi thua cuộc.

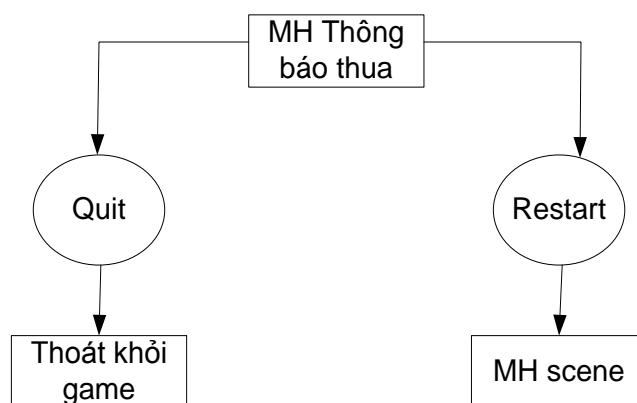


Hình 3.28: Màn hình thông báo thua

b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Button <b>Restart</b>	Chơi lại level đang chơi trong scene
2		Button <b>Quit</b>	Thoát khỏi game.

c. Sơ đồ mô tả tình huống sử dụng:



Hình 3.29: Sơ đồ tình huống màn hình thông báo thua

### 3.2.13 Màn hình chiến thắng ở level:

a. **Thể hiện và ý nghĩa:** Thông báo cho người chơi biết mình đã vượt qua một level trong scene.

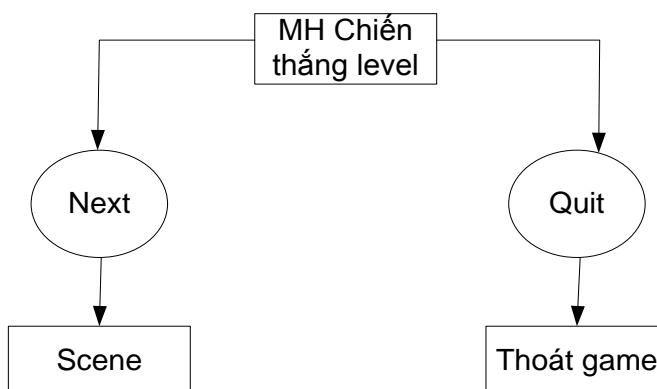


Hình 3.40: Sơ đồ tình huống màn hình thông báo thua

c. **Ý nghĩa các control:**

STT	Hình	Control	Mô tả
1		Button Next	Đến với level cao hơn trong scene.
2		Button Quit	Thoát khỏi game.

d. **Sơ đồ mô tả tình huống sử dụng:**



Hình 3.41: Sơ đồ tình huống màn hình chiến thắng ở level


### 3.2.14 Màn hình chiến thắng ở scene:

a. **Thể hiện và ý nghĩa:** Sau khi người chơi hoàn thành một scene sẽ có có màn hình thông báo và cho phép người chơi qua scene tiếp theo.

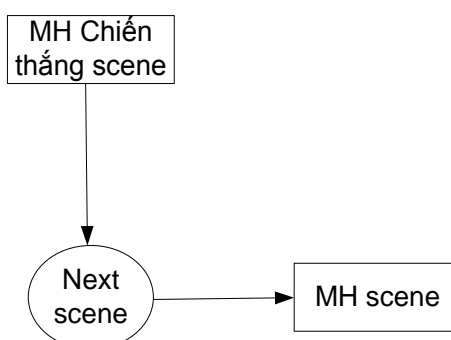


Hình 3.42: Màn hình chiến thắng ở scene

b. Ý nghĩa các control:

STT	Hình	Control	Mô tả
1		Button Next Scene	Cho phép người chơi đến với scene tiếp theo

c. Sơ đồ mô tả tình huống sử dụng:



Hình 3.43: Sơ đồ tình huống màn hình chiến thắng ở scene



# KẾT LUẬN

## KẾT LUẬN

### **✚ Hạn chế và khó khăn của đề tài.**

Trong suốt thời gian thực hiện đề tài, mặc dù đã cố gắng nhưng vẫn còn tồn tại những hạn chế nhất định. Một trong những lý do để tạo nên sự hạn chế đó chính là trình độ, khả năng nghiên cứu của nhóm chúng em.

Do thời gian ngắn, nên mặc dù đã nỗ lực hết sức nhưng sản phẩm của chúng em vẫn còn một số hạn chế sau:

- Kho dữ liệu kiến thức về ngôn ngữ lập trình Pascal chưa lớn.
- Giao diện đồ họa trong game chưa được hấp dẫn.

### **✚ Hướng phát triển:**

Từ những hạn chế trên nhóm chúng em xin đưa ra hướng phát triển cho đề tài như sau:

- Viết hoàn thiện Stage 2 của trò chơi.
- Cung cấp thêm nhiều dạng trắc nghiệm như sắp xếp đoạn code, trắc nghiệm nhiều lựa chọn, dạng câu hỏi dạng lắp ghép .
- Xây dựng thêm những tính năng mới nhằm tăng sự hấp dẫn trong game.

## **TÀI LIỆU THAM KHẢO**

### **Tiếng Anh:**

- [1] Allyn & Bacon Roger B. Myerson (1991), Game Theory: Analysis of Conflict, Harvard University Press, p.1
- [2] Andrew Troelsen (2010), Pro Expression Blend 4, Apress.
- [3] Jack Xu (2009), Practical WPF Charts and Graphics, Apress.
- [4] John Sharp (2010), Microsoft Visual C# 2010, Microsoft Press.
- [5] Kosinska and Chris Leeds (2010) , Step by step Microsoft Expression Blend, Microsoft.
- [6] Le, D.-L, Tran, V.-H, Hunger, A., Nguyen, D.-T (2008), e-Course and its Applications in Blended-Learning Environment. In Proceedings of the 2nd International Conference on Communications and Electronics (ICCE 2008), published by IEEE-ISBN: 978-1-4244-2425-2/08, Hoi An, Vietnam (4-6/6/2008). (in English), pp 482-487
- [7] Le, D.-L, Nguyen, D.-T, Nguyen, A.-T, Tran, V.-H, Hunger, A. (2010), Applying Pedagogical Analyses to Create an On-line Course for e Learning. In Lecture Notes in AI (LNAI 6277) from the 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2010) published by Springer-Verlag Berlin Heidelberg, 8-10th Sep 2010, Cardiff, Wales, UK (in English), Part II, pp 114-123 .
- [8] Le, D.-L, Nguyen, D.-T, Nguyen, A.-T, Tran, V.-H, Hunger, A. (2011), Pedagogical domain knowledge for Adaptive e-Learning. In the Science and Technology Development Journal of VNU-HCM - Natural Sciences: Mathematics & Information Technology - Vol. 14 (T1-2011) - ISSN 1859-0128, Hochiminh city Vietnam (in English), pp 14-34.
- [9] Leonard, Robert. Von Neumann, Morgenstern (2010), The Creation of Game Theory, Cambridge University Press.
- [10] Margaret D. Roblyer & Aron H. Doering (2010), Integrating Educational Technology into Teaching – Fifth edition.
- [11] Matthew MacDonald (2010), Pro WPF in C# 2010, Apress
- [12] Marc Prensky (2001), Digital Game –Based learning, Magazine Computers in Entertainment (CIE) - Theoretical and Practical Computer Applications in Entertainment.
- [13] Patrick Stack (2005), "History of video game consoles" *Time Magazine website*.
- [14] Pavel Zemliansky & Diane Wilcox (2010), Design and Implementation of Educational Games, Information science reference, pp. 108 – 125.
- [15] Rod Stephen (2010), WPF programmer's Reference, Wiley Publishing – Inc.
- [16] Satoshi Amagai et al (2005), Summit on educational games, Federation of American Scientists.

### **Website tham khảo:**

- [17] **Inventors.about** - [http://inventors.about.com/library/inventors/blcomputer\\_videogames.htm](http://inventors.about.com/library/inventors/blcomputer_videogames.htm)
- [18] Game online - <http://www.trochoivui.com/game/5936/choi-co-vua-8.html>

[19]Trang chủ phần mềm giáo dục - <http://www.educational-freeware.com/freeware/timez-attack.aspx>

[20]Trang hỗ trợ giáo viên - <http://teacherswithapps.com/ansel-and-clairs-adventures-in-africa/>

[21]Forum game online - <http://forums.gamevn.com/showthread.php?572999-Huong-dan-Sim-City-4-Deluxe>.

[22]Game online - <http://www.pappumathgames.com/zombie-master-bt/>

[23]Game online- <http://boardgamegeek.com/boardgame/22681/atommate>

# PHỤ LỤC

# PHU LUC

## • Hướng dẫn sử dụng:

### 1. Cài đặt phần mềm:

- Xem chi tiết hướng dẫn cài đặt tại địa chỉ sau:

<http://www.youtube.com/watch?v=RJT1ML-c-a0&feature=youtu.be>

### 2. Hướng dẫn thêm câu hỏi mới:

- Cấu trúc của 1 câu hỏi (Question) được lưu dưới dạng file xml như sau:

```
<Question idQuestion="1" idType="2" idTopic="1" idLevel="1" Title="Chọn đáp án đúng" Content="Câu trúc rẽ nhánh có mấy dạng" />
```

Trong đó :

- **idQuestion** : là khóa chính của mã câu hỏi.
- **idType**: là khóa ngoại dùng để biết câu hỏi thuộc loại nào (điền khuyết, đúng sai, câu đố).

Chi tiết về bảng **QuestionType** (loại câu hỏi) như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<Data>
  <QuestionType idType="1" Description="Quiz" Note="Câu hỏi lý thuyết" />
  <QuestionType idType="2" Description="Multichoice" Note="Câu hỏi multichoice lý thuyết" />
  <QuestionType idType="3" Description="Excercise Quiz" Note="Câu hỏi quiz bài tập" />
  <QuestionType idType="4" Description="Excercise Multichoice" Note="Câu hỏi multichoice bài tập" />
  <QuestionType idType="5" Description="TrueFail" Note="Câu hỏi đúng sai" />
  <QuestionType idType="6" Description="True fail bài tập" Note="True false bài tập" />
</Data>
```

- **idLevel**: Khóa ngoại dùng để biết mức độ khó của câu hỏi.

Chi tiết về bảng **Level** (mức độ khó) như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<Data>
  <Level idLevel="1" Description="Dễ" Note="Mức độ dễ" />
  <Level idLevel="2" Description="Trung bình" Note="Mức độ trung bình" />
  <Level idLevel="3" Description="Khó" Note="Mức độ khó" />
</Data>
```

- **Title**: Tiêu đề của câu hỏi.
- **Content**: Nội dung câu hỏi.

Ví dụ: Để chèn một câu hỏi loại MultiChoice lý thuyết với 4 đáp án, mức độ dễ thì cần làm như sau:

- B1: Chèn thêm một câu hỏi với cấu trúc như hướng dẫn trên vào

file **Question.xml**:

```
<Question idQuestion="1" idType="2" idTopic="1" idLevel="1" Title="Chọn đáp án đúng" Content="Cấu trúc rẽ nhánh có mấy dạng" />
```

- B2: Sau đó thêm những đáp án tương ứng với mã câu hỏi vừa tạo vào file

**MultiChoice.xml**:

Với cấu trúc đáp án (MultiChoice) như sau:

```
<MultiChoice idAnswer="1" Answer="1" IsCorrect="0" idQuestion="1" />
```

Trong đó:

- **idAnswer**: Mã đáp án.
- **Answer**: Nội dung của đáp án.
- **IsCorrect**: Nếu đáp án đúng thì nhập 1, ngược lại nhập 0
- **idQuestion**: Khóa ngoại để biết đáp án này của câu hỏi nào.

Trong ví dụ này mã câu hỏi ( idQuestion ) là 1. Cấu trúc đầy đủ của câu hỏi trên với 4 đáp án là:

```
<MultiChoice idAnswer="1" Answer="1" IsCorrect="0" idQuestion="1" />  
<MultiChoice idAnswer="2" Answer="2" IsCorrect="1" idQuestion="1" />  
<MultiChoice idAnswer="3" Answer="3" IsCorrect="0" idQuestion="1" />  
<MultiChoice idAnswer="4" Answer="4" IsCorrect="0" idQuestion="1" />
```

Kết quả xuất ra màn hình :



Tương tự cho câu hỏi dạng Quiz, TrueFalse.

### 3. Cấu hình cho từng Scene trong game:

Chúng ta sẽ tùy chỉnh trong file Scene\_Config, sau đây là ví dụ về tùy chỉnh cấu hình trong scene 3 (nhiều thông số nhất):

Mô tả về cấu trúc:

```
<Scene_Config>
  <idScene>3</idScene>
  <TotalSupporter>10</TotalSupporter>
  <TimeRemain>10</TimeRemain>
  <TimeToAddSnail>5</TimeToAddSnail>
  <TimeToAddWorm>10</TimeToAddWorm>
  <TimeToAddTurtle>20</TimeToAddTurtle>
  <Target>3</Target>
  <SnailSpeed>0.5</SnailSpeed>
  <WormSpeed>0.75</WormSpeed>
  <TurtleSpeed>0.25</TurtleSpeed>
</Scene_Config>
```

- Giải thích:
  - **idScene:** Khóa ngoại dùng để xác định cấu hình này dùng cho cảnh chơi nào.
  - **TotalSupporter:** Dùng để giới hạn số lượng nhân vật hỗ trợ giúp tạo nhanh cây bông khi nhấp chuột vào. Ví dụ ở màn 1 là mặt trời.
  - **TimeRemain:** Dùng để định thời gian tạm dừng di chuyển của các nhân vật khi người chơi nhấp chuột vào biểu tượng đồng hồ.
  - **TimeToAddSnail:** Dùng để định thời gian xuất hiện của con ốc sên cách nhau bao lâu.
  - **TimeToAddWorm:** Dùng để định thời gian xuất hiện của con sâu cách nhau bao lâu.
  - **TimeToAddTurtle:** Dùng để định thời gian xuất hiện của con rùa cách nhau bao lâu.
  - **Target:** Dùng để định số câu hỏi đúng cần đạt để có thể vượt qua một level của màn chơi.
  - **SnailSpeed:** Dùng để định tốc độ di chuyển của con ốc sên.
  - **WormSpeed:** Dùng để định tốc độ di chuyển của con sâu.
  - **TurtleSpeed:** Dùng để định tốc độ di chuyển của con rùa.

Như vậy để cấu hình trong game chúng ta chỉ việc thay đổi các thông số ở các dòng lệnh tương ứng với chức năng muốn tùy chỉnh ở trên.



#### 4 .Cấu hình loại câu hỏi,mức độ khó của Scene:

Chúng ta phải tùy chỉnh thông số trong file `Scene_Question.xml`, cấu trúc file như sau:

```
<Scene Question idScene="1" idLevel="1" idType="5" />
```

Trong đó:

- **idScene**: là khóa ngoại dùng để biết những cấu hình này dùng cho cảnh chơi(Scene) nào.
- **idLevel** : là khóa ngoại dùng để tham chiếu đến bảng Level để biết mức độ khó của câu hỏi.
- **idType**: là khóa ngoại dùng để tham chiếu đến bảng QuestionType để biết câu hỏi thuộc loại nào.

### • Hướng dẫn chơi game:

➤ Cách chơi chung là ở mỗi màn chơi hệ thống sẽ cung cấp cho người chơi một số nhân vật hỗ trợ với số lượng giới hạn, người chơi có nhiệm vụ là hãy nhấp chọn thật nhanh chúng để trồng được nhiều bông hoa. Và mỗi bông hoa sẽ chứa nội dung câu đố, do đó người chơi phải nhấp chọn bông hoa để giải đáp đủ số lượng câu đố quy định ở mỗi màn chơi – scene nếu muốn qua scene tiếp theo.

Link: <http://www.youtube.com/watch?v=FaR9r5qnUGo&feature=youtu.be>

#### ➤ **Hướng dẫn scene 1:**

- Nhiệm vụ của bạn trong màn chơi 1 là phải trả lời đúng 3 câu đố trong mỗi level của màn chơi.
- Để hiển thị các câu đố bạn phải chọn vào các cây bông mà mình trồng được trong khu vườn và các câu đố chỉ xuất hiện khi bạn trồng được nhiều hơn 5 cây hoa. Riêng với level 2 sẽ có sẵn 10 cây hoa cho bạn và bạn phải nhanh chóng chọn hoa để trả lời câu đố.
- Câu đố ở đây có 2 lựa chọn và bạn chỉ việc chọn một đáp án đúng nhất.
- Trong lúc trả lời sẽ có 1 loài vật chống phá khu vườn của bạn. Khi con vật tiếp

xúc cây hoa sẽ ăn mất cây hoa của bạn. Và nếu chúng tấn công vào tới căn nhà của mình coi như bạn thua cuộc.

- Link: <http://www.youtube.com/watch?v=mtNXsD4TZbU>

➤ **Hướng dẫn scene 2:**

- Ở màn chơi này cũng sẽ tương tự màn chơi 1 nhưng có một số thay đổi như sau:

+ Câu đố ở đây có 4 lựa chọn và bạn chỉ việc chọn một đáp án đúng nhất.

+ Sẽ có 2 con vật tấn công khu vườn.

- Link: [http://www.youtube.com/watch?v=GBO\\_D0IxyLw](http://www.youtube.com/watch?v=GBO_D0IxyLw)

➤ **Hướng dẫn scene 3:**

- Ở màn chơi này sẽ khó hơn một chút so với màn chơi 2 như sau:

+ Câu đố ở đây bạn phải tự suy nghĩ đáp án và điền vào chỗ trống bên dưới câu hỏi

+ Có thêm 1 loài vật nữa, như vậy có tất cả 3 con vật tấn công khu vườn.

- Link: <http://www.youtube.com/watch?v=MnkXudpyxWQ>

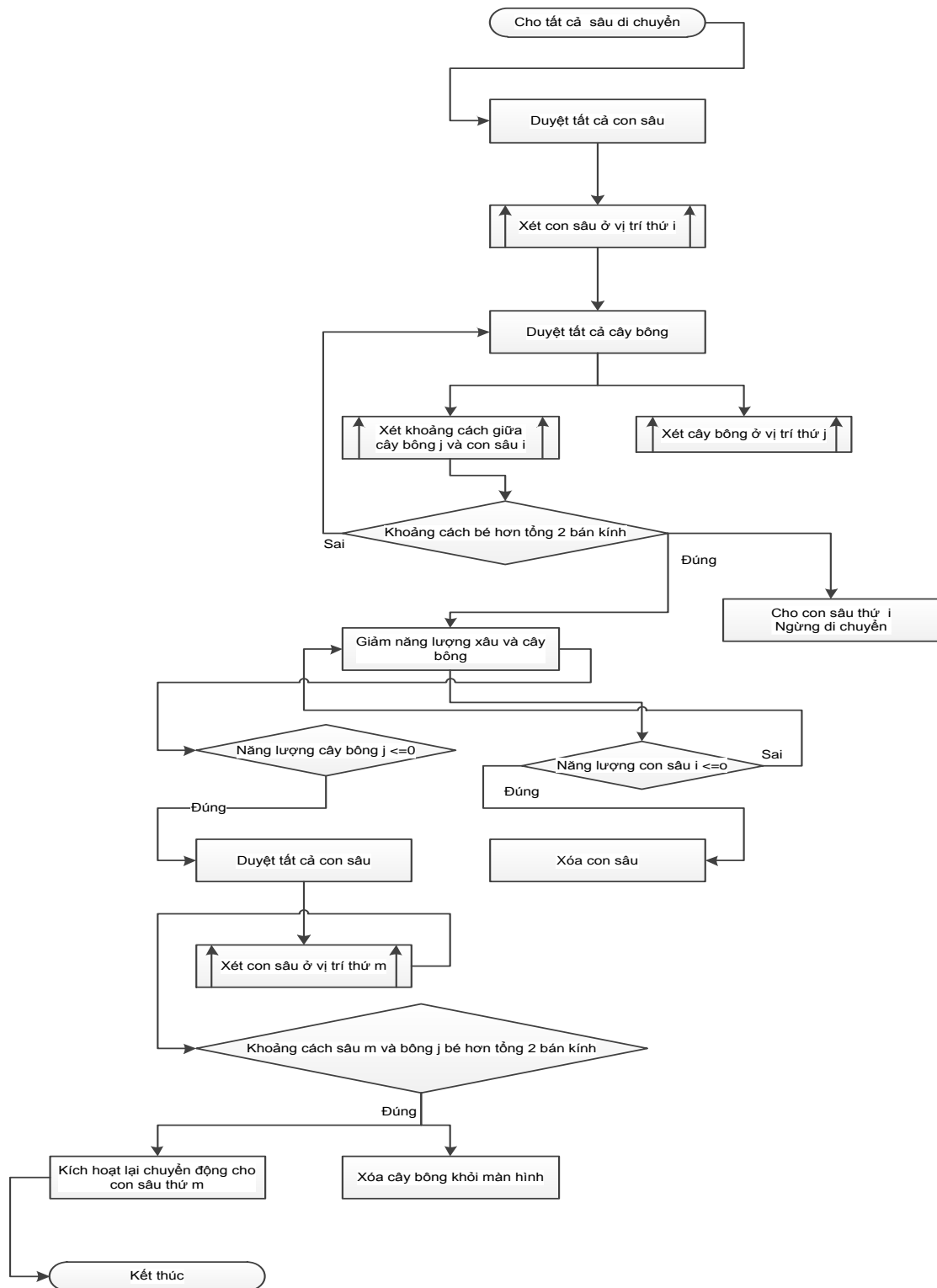
## • Công nghệ xử lý:

### 1. Giải thuật xử lý khi cây bông và con xâu tiếp xúc nhau.



- **Ý tưởng** : Ta đưa các nhân vật trong game về dạng hình học cơ bản như hình tròn hoặc hình chữ nhật, để đơn giản trong đề tài này chúng em đưa về dạng hình chữ nhật, từ đó ta có thể xét xem khi nào hai hình chữ nhật này tiếp xúc nhau bằng cách sử dụng định lý Pytago để tính khoảng cách và so sánh với tổng 2 bán kính của các nhân vật cần xử lý khi tiếp xúc nhau. Nếu khoảng cách này bé hơn hoặc bằng tổng hai bán kính thì các nhân vật này tiếp xúc nhau.

**- Lưu đồ xử lý tiếp xúc giữa con sâu và cây bông:**



Đoạn code xử lý va chạm

```

...
bool isCollision(David _oc, Alice _flower)
{
    Rect rect1 = _oc.BoundsRect;
    Rect rect2 = _flower.BoundsRect;
    double xd = rect1.X - rect2.X;
    double yd = rect1.Y - rect2.Y;
    double sumRadius = (rect1.Width / 2) + (rect2.Width / 2); //R1+R2=R12
    double sqrRadius = sumRadius * sumRadius; //R12*R12
    double distSqr = (xd * xd) + (yd * yd); //|AB|=sqrt((xA-xB)^2+(yA-yB)^2)
    if (distSqr <= sqrRadius) return true; //Khoang cach be hon tong 2 ban kinh
    else return false;
}

void CheckCollision()
{
    #region Kiểm tra va chạm
    for (int i = 0; i < lstSnail.Count; i++)
    {
        //Xét con sên ở vị trí i
        var snailI = lstSnail[i];
        //Nếu con sên ở trạng thái active thì cho chuyển động
        if (snailI.isActive == true)
        {
            //Ấn ống máu
            snailI.prBlood.Visibility = Visibility.Collapsed;
            //Di chuyển con sên với vận tốc là SnailSpeed
            snailI.X += SnailSpeed;
        }
        //Ngược lại cho ngừng chuyển động
        else if (snailI.isActive == false)
        {
            snailI.X += 0;
        }
    }
    #region Ốc sên dung bình hay không
    //Nếu ốc sên đang ở vị trí mà lớn hơn Độ rộng màn hình trừ đi
    //-2 lần chiều rộng thực tế khi ốc sên được vẽ ra trên màn hình
    //thì hiển thị thông báo thua
    if (snailI.X > System.Windows.SystemParameters.PrimaryScreenWidth -
snailI.ActualWidth * 2)
    {
        //Dừng trò chơi và cho xuất hiện thông báo thua
    }
}

```

```

#endregion
if (lstSnail.Count > 0)
{
    for (int j = 0; j < lstFlower.Count; j++)
    {
        var flowerJ = lstFlower[j];
        if (snail.Y == flowerJ.Y)
        {
            //Kiểm tra xem ốc sên và bông hoa có tiếp xúc nhau hay không
            if (isCollision(snail, flowerJ) == true)
            {
                //Hiển thị ống máu
                flowerJ.prBlood.Visibility = Visibility.Visible;
                snail.prBlood.Visibility = Visibility.Visible;
                //Giảm máu
                flowerJ.Blood -= 0.25;
                //Cập nhật hiển thị máu lên progressbar
                flowerJ.UpdatePrBlood();
                //Cập nhật lại trạng thái active cho ốc sên
                //để ốc sên ngừng chuyển động
                snail.isActive = false;
                snail.Blood -= 0.25;
                snail.UpdatePrBlood();
            }
            if (flowerJ.Blood <= 0)
            {
                snail.isActive = true;
                for (int l = 0; l < lstSnail.Count; l++)
                {
                    if (lstSnail[l].Y == flowerJ.Y)
                    {
                        if (isCollision(lstSnail[l], flowerJ) == true)
                        {
                            lstSnail[l].isActive = true;
                        }
                    }
                }
                RemoveFlower(flowerJ);
            }
            if (snail.Blood <= 0)
            {
                {
                    RemoveSnail(snail);
                }
            }
        } //End of Snail and flower Same Row
    } //End of for Flower
} //End of for Snail
#endregion
} //End of Collision

```

## 2. Cách làm cho nhân vật chuyển động trong WPF



## - Ý tưởng:

+ Sự chuyển động của nhân vật thực chất là một chuỗi các hình ảnh diễn tả các trạng thái của nhân vật được xuất hiện trong khoảng thời gian ngắn.

+ Bất chước ý tưởng này nhóm chúng tôi đã nghiên cứu và tìm ra cách tạo chuyển động cho nhân vật trong WPF.

## - Cách thực hiện.

Có 3 cách để thực hiện việc này. Dùng Storyboard, Dùng DispatcherTimer, dùng Compositarget.

### + Dùng Storyboard

```
<Storyboard x:Key="Animation" RepeatBehavior="Forever" >
  <ObjectAnimationUsingKeyFrames Storyboard.TargetName="img" Storyboard.TargetProperty="(Image.Source)">
    <DiscreteObjectKeyFrame KeyTime="0:0:0.00" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/1.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="0:0:0.10" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/2.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="0:0:0.20" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/3.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="0:0:0.30" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/4.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="0:0:0.40" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/5.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
    <DiscreteObjectKeyFrame KeyTime="0:0:0.50" >
      <DiscreteObjectKeyFrame.Value>
        <BitmapImage UriSource="/Images/GameObject/Tony/6.png" />
      </DiscreteObjectKeyFrame.Value>
    </DiscreteObjectKeyFrame>
  </ObjectAnimationUsingKeyFrames>
</Storyboard>
```

+ Dùng DispatcherTimer: Trước khi bắt đầu bằng cách này thì chúng em xin giới thiệu sơ qua DispatcherTimer đây là một class kế thừa từ lớp Dispatcher, để sử dụng thì cần khai báo thư viện: **using System.Windows.Threading;**

Nếu như Window form có Timer thì WPF có DispatcherTimer.

+ **Cách sử dụng:**

- Với **Interval** là khoảng thời gian tiếp theo sẽ xảy ra sự kiện **Tick(animation)**, **Interval** có thể là mili giây, giây, phút, giờ.
- **Start** để chạy khởi động **DispatcherTimer** này.

```

public partial class chon : UserControl
{
    int i; //Hình thứ i
    DispatcherTimer timer = new DispatcherTimer();
    public chon()
    {
        InitializeComponent();
        timer.Interval = TimeSpan.FromSeconds(0.1); //khoảng cách xuất hiện các hình
        timer.Tick += new EventHandler(timer_Tick); // sau thời gian trên thực hiện việc gì
        timer.Start();
    }

    void timer_Tick(object sender, EventArgs e)
    {
        i++;
        switch (i)
        {
            case 1: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/1.png", UriKind.RelativeOrAbsolute)); break;
            case 2: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/2.png", UriKind.RelativeOrAbsolute)); break;
            case 3: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/3.png", UriKind.RelativeOrAbsolute)); break;
            case 4: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/4.png", UriKind.RelativeOrAbsolute)); break;
            case 5: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/5.png", UriKind.RelativeOrAbsolute)); break;
            case 6: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/6.png", UriKind.RelativeOrAbsolute)); break;
            default: img.Source = new BitmapImage(new Uri(@"pack://application:,,,/Images/GameObject/Chon/1.png", UriKind.RelativeOrAbsolute)); i = 0; break;
        }
    }
}

```

Tương tự như vậy với **CompositionTarget** chỉ cần gọi event thì nhân vật sẽ chuyển động.

**CompositionTarget.Rendering += timer\_Tick;**

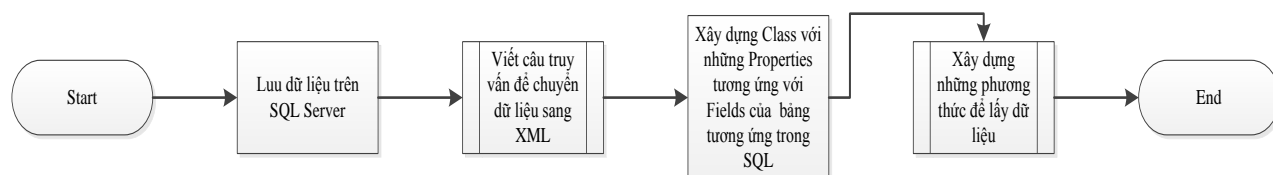
### 3. Lấy dữ liệu với XML

#### - Ý tưởng:

Trong đề tài này chúng em sử dụng LINQ To XML để lấy dữ liệu với XML vì dữ liệu được xây dựng là dạng dữ liệu quan hệ và việc viết câu truy vấn lấy dữ liệu bằng LINQ cũng gần như tương tự viết bằng SQL.

Để thực hiện việc này cho đơn giản và tiết kiệm thời gian, đầu tiên dữ liệu cần được lưu trên SQL Server sau đó dùng SQL Server để chuyển dữ liệu này thành dữ liệu dạng XML.

Sơ đồ



Ví dụ câu query:

```
SELECT*  
FROMLevel  
FORXMLAUTO
```

Sẽ tạo ra file có cấu trúc như sau

```
<LevelidLevel="1"Level="Dễ"Note="Mức độ dễ" />  
<LevelidLevel="2"Level="Trung bình"Note="Mức độ trung bình" />  
<LevelidLevel="3"Level="Khó"Note="Mức độ khó" />
```

Câu Query

```
SELECT*  
FROMLevel  
FORXMLAUTO,ELEMENTS
```

Sẽ tạo ra file có cấu trúc như sau

```
<Level>  
<idLevel>1</idLevel>  
<Description>Dễ</Description>  
<Note>Mức độ dễ</Note>  
</Level>  
<Level>  
<idLevel>2</idLevel>  
<Description>Trung bình</Description>
```



```

<Note>Mức độ trung bình</Note>
</Level>
<Level>
<idLevel>3</idLevel>
<Description>Khó</Description>
<Note>Mức độ khó</Note>
</Level>

```

Sau khi có file XML chúng ta xây dựng lớp có tên là Level để thao tác dữ liệu trên file này:

```

class Level
{
    public int idLevel { get; set; }
    public string Description { get; set; }
    public string Note { get; set; }
    public const string xmlPath = "XML/Level.xml";
}

```

Để lấy dữ liệu chúng ta xây dựng phương thức như sau:

```

public static List<Level> GetLevel()
{
    List<Level> lst = new List<Level>();
    XElement root = XElement.Load(xmlPath);
    IEnumerable<XElement> xe =
        from ele in root.Elements("Level")
        select ele;
    foreach (XElement e in xe)
    {
        Level q = new Level
        {
            idLevel = int.Parse(e.Element("idLevel").Value),
            Description = e.Element("Description") == null ? null : e.Element("Description").Value,
            Note = e.Element("Note") == null ? null : e.Element("Note").Value,
        };
        lst.Add(q);
    }
    return lst;
}

```

Giải thích: Đầu tiên dùng **Xelement.Load(xmlPath)** để đọc file xml chứa dữ liệu LevelSau đó dùng **Ienumerable** để lấy ra những Element là Level.Tiếp theo duyệt từng Element trong Element của Level và gán dữ liệu vào những thuộc tính của lớp Level.

Ví dụ về Element Level

```

<Level>
  <idLevel>1</idLevel>
  <Description>Dễ </Description >
  <Note>Mức độ dễ</Note>|
</Level>

```

Trong *Element* Level có các *Element* như *idLevel*, *Description*, *Note*

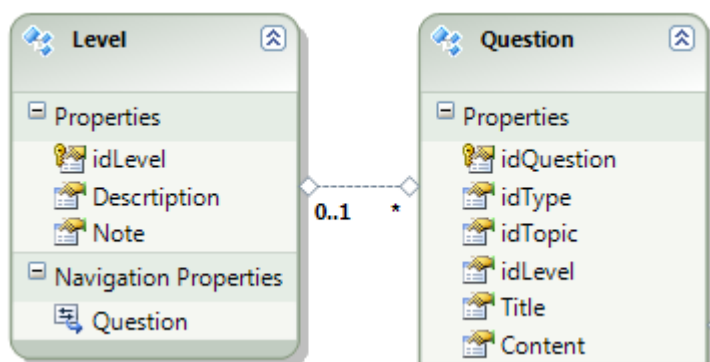
Tương tự nếu *Element* Level có cấu trúc như sau

```
<Level idLevel="1" Level="Dễ" Note="Mức độ dễ" />
```

Thì để gán giá trị của *Attribute* *idLevel* cho thuộc tính *idLevel* của lớp *Level* ta phải sử dụng *Attribute* *idLevel* = `int.Parse(e.Attribute("idLevel").Value)`

## 4. Cách Join bảng với LINQ To XML

Ví dụ bảng *Level* và *Question*



Để thực hiện bạn thêm một *Properties* cho lớp *Level* có tên là

```

public List<Question> Questions
{
  get
  {
    return Question.GetByLevel(idLevel);
  }
  set { }
}

```

Ở lớp Level bạn xây dựng phương thức

```
public static List<Question> GetByLevel(int level)
{
    List<Question> lst = new List<Question>();
    XElement root = XElement.Load(xmlPath);
    IEnumerable<XElement> xe =
        from ele in root.Elements("Question")
        where int.Parse(ele.Element("idLevel").Value) == level
        select ele;
    foreach (XElement e in xe)
    {
        Question m = new Question
        {
            idQuestion = int.Parse(e.Element("idQuestion").Value),
            Content = e.Element("Content").Value,
            idLevel = int.Parse(e.Element("idLevel").Value),
            idType = int.Parse(e.Element("idType").Value),
            Title = e.Element("Title") == null ? null : e.Element("Title").Value,
        };
        lst.Add(m);
    }
    return lst;
}
```

Cách 2:

```
public static Scene GetScenePlaying(int userID, int stageID)
{
    Scene _data = (from s in Scene.GetByStageID(stageID)
        join us in UserScene.GetByUserID(userID)
        on s.idScene equals us.SceneID
        where us.isPlaying == "1"
        select s).FirstOrDefault();
    return _data;
}
```

Tạo một file xml chứa thông tin người chơi bằng LINQ To XML

```
public void NewUser(string UserName, string Password)
{
    XElement doc = XElement.Load("XML/User.xml");
    XElement newUser =
        new XElement("User",
            new XElement("UserID", User.Users().Count() + 1),
            new XElement("UserName", UserName),
            new XElement("Password", Password),
            new XElement("Point", 0),
            new XElement("Rank", 0));
    doc.Add(newUser);
    doc.Save("XML/User.xml");
}
```

Sau khi gọi chúng ta sẽ lưu được thông tin người chơi với cấu trúc như sau

```
<User>
  <UserID>2</UserID>
  <UserName>nktai</UserName>
  <PassWord>nktai</PassWord>
  <Point>75</Point>
  <Rank>2</Rank>
</User>
```

## 5. Cập nhật dữ liệu với LINQ To XML

Ví dụ để cập nhật điểm của người chơi

```
public static void UpdateUserPoint(int UserID)
{
    int Point = UserScene.GetByUserID(UserID).Sum(s => s.Point);
    XElement doc = XElement.Load("XML/User.xml");
    IEnumerable<XElement> user = from b in doc.Elements("User")
                                where (b.Element(
                                    "UserID").Value == UserID.ToString())
                                select b;
    foreach (XElement xe in user)
    {
        xe.SetElementValue("Point", Point.ToString());
    }
    doc.Save("XML/User.xml");
}
```

Sử dụng “NOT IN” trong LINQ To XML

```
public static List<Scene> GetSceneNotPlayed(int _userID, int _stageID)
{
    List<Scene> scene = Scene.GetByStageID(_stageID);
    List<Scene> played = UserScene.GetScenePlayed(_userID, _stageID);
    List<Scene> _notplay = new List<Scene>();
    if (played.Count() == 0) { _notplay = scene; }
    else
    {
        _notplay = scene.Except(played, new SceneComparer()).ToList();
    }
    return _notplay;
}
```

Sử dụng Except để thực hiện lệnh NOT IN tương tự SQL ,

Và khởi tạo lớp `SceneComparer` để so sánh hai đối tượng cùng loại với nhau

```

public class SceneComparer : IEqualityComparer<Scene>
{
    /// <summary>
    /// Kiểm tra xem 2 đối tượng(Scene)có giống nhau hay không
    /// </summary>
    public bool Equals(Scene a, Scene b)
    {
        if (object.ReferenceEquals(a, b)) return true;
        if (object.ReferenceEquals(a, null) || object.ReferenceEquals(b, null))
            return false;
        return a.idScene == b.idScene && a.idStage == b.idStage;
    }

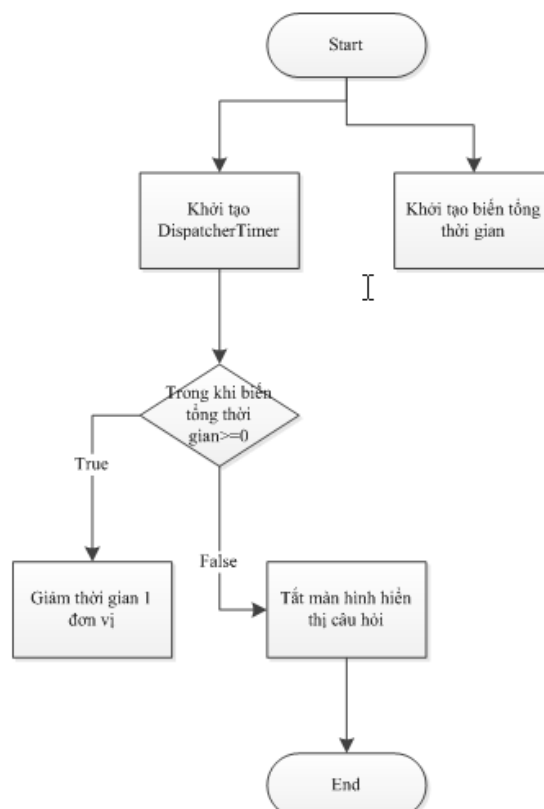
    /// <summary>
    /// Lấy mã GetHashCode
    /// </summary>
    public int GetHashCode(Scene scene)
    {
        if (object.ReferenceEquals(scene, null)) return 0;
        int hashSceneID = scene.idScene.GetHashCode();
        return hashSceneID;
    }
}

```

## 6. Cách tạo đồng hồ đếm ngược với DispatcherTimer

-**Ý tưởng:** thông qua thuộc tính Interval của DispatcherTimer để xét khoảng thời gian xuất hiện giữa các sự kiện. Để làm đồng hồ đếm ngược chúng ta thiết lập interval là 1 giây, và thông qua một biến dùng để thiết lập tổng thời gian đếm ngược. Mỗi khi DispatcherTimer được Tick lên thì giảm biến tổng thời gian đi 1 đơn vị.

- Sơ đồ:



### Đoạn code tương ứng:

```
/// <summary>
/// Khoảng thời gian tắt màn hình hiển thị câu hỏi
/// </summary>
public int TimeToClose;
/// <summary>
/// Timer thời gian Dem ngược thời gian,moi lan 1 giay
/// </summary>
public DispatcherTimer ClockTimer;

/// <summary>
/// hàm xử lý đếm ngược thời gian
/// </summary>
public void TimeCountDown()
{
    lbTimeRemain.Visibility = Visibility.Visible;
    ClockTimer = new DispatcherTimer();
    ClockTimer.Interval = TimeSpan.FromSeconds(1);
    ClockTimer.Tick += new EventHandler(ClockTimer_Tick);
    ClockTimer.Start();
}

/// <summary>
/// Đồng hồ đếm ngược thời gian trả lời câu hỏi
/// </summary>
void ClockTimer_Tick(object sender, EventArgs e)
{
    if (TimeToClose >= 0)
    {
        lbTimeRemain.Content = TimeToClose--;
    }
    else
    {
        ClockTimer.Stop();
        this.Close();
    }
}
}
```

# • Sơ lược về nội dung chương trình Pascal

Chương trình, chuẩn kiến thức và nội dung môn Tin học lớp 11 được xây dựng dựa trên yêu cầu, mục tiêu môn Tin học ở cấp Trung học phổ thông

## **1. Mục tiêu chung của Tin học 11:**

- Kiến thức: trang bị cho học sinh một số khái niệm cơ bản về lập trình và ngôn ngữ lập trình bậc cao.
- Kỹ năng: giải được một số bài toán đơn giản trên máy tính bằng cách vận dụng được các kiến thức về thuật toán, cấu trúc dữ liệu, ngôn ngữ lập trình cụ thể, sử dụng các chương trình con có sẵn.
- Thái độ: ham thích môn học, có tính kỉ luật cao và tinh thần làm việc theo nhóm.

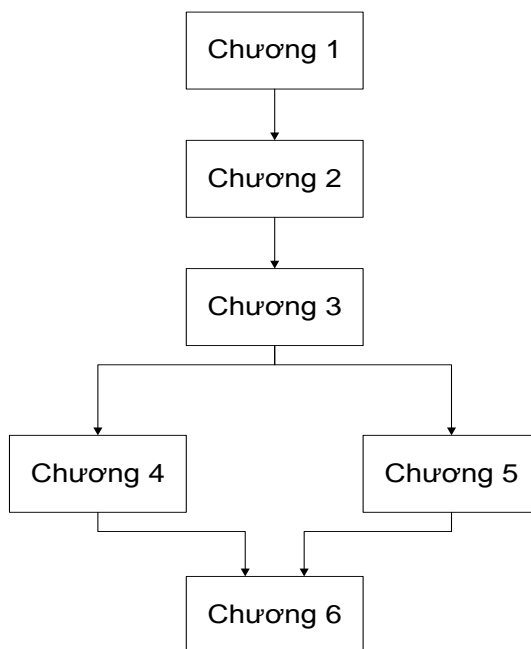
## **2. Nội dung chính của môn học Tin học 11**

- Các khái niệm cơ sở của lập trình, những đặc điểm của ngôn ngữ lập trình bậc cao.
- Các thành phần trong ngôn ngữ lập trình: Tên, tên chuẩn, từ khóa, hằng, biến,...
- Các quy định trong ngôn ngữ lập trình và các quy định riêng trong ngôn ngữ lập trình Pascal.
- Các cấu trúc quan trọng trong ngôn ngữ lập trình bậc cao và minh họa trong ngôn ngữ lập trình Pascal.
- Kỹ năng sử dụng ngôn ngữ lập trình cụ thể (Pascal) để lập trình giải một số bài toán để xây dựng thuật toán hoặc đã có thuật toán trong sách giáo khoa Tin học 10.

### 3.Cấu trúc chương mục

Sách giáo khoa Tin học 11 gồm 6 chương:

- Chương I: **Một số khái niệm về lập trình và ngôn ngữ lập trình**, gồm 2 bài (2 tiết lý thuyết, 0 tiết thực hành, 1 tiết bài tập).
- Chương II: **Chương trình đơn giản**, gồm 6 bài lý thuyết và 1 bài tập thực hành (4 tiết lý thuyết, 2 tiết thực hành, 1 tiết bài tập).
- Chương III: **Cấu trúc rẽ nhánh và lặp**, gồm 2 bài lý thuyết và 1 bài tập thực hành (4 tiết lý thuyết, 2 tiết thực hành, 1 tiết bài tập).
- Chương IV: **Kiểu dữ liệu có cấu trúc**, gồm 3 bài lý thuyết và 3 bài tập thực hành (7 tiết lý thuyết, 6 tiết thực hành, 2 tiết bài tập).
- Chương V: **Tập và thao tác với tập**, gồm 3 bài lý thuyết (2 tiết lý thuyết, 0 tiết thực hành, 1 tiết bài tập).
- Chương VI: **Chương trình con và lập trình có cấu trúc**, gồm 3 bài lý thuyết và 3 bài tập thực hành (6 tiết lý thuyết, 6 tiết thực hành, 1 tiết bài tập).





## Hệ thống hóa kiến thức theo bài:

