

Báo cáo môn: Nhập môn phát triển game

Tên đồ án: Game Binh Nhí

Giảng viên: TS Ngô Thanh Hùng

Nhóm thực hiện:

Trương Mai Thanh Trí 09520320

Nguyến Minh Vượng 09520364

Vũ Minh Tuấn 09520333

Lời nói đầu:

Lập trình game là một công việc khó và nặng nhọc, với khối lượng code rất lớn và phức tạp, đòi hỏi sự tập trung cao. Game cũng góp một phần nào đó làm thế giới thêm sinh động và những dòng code có thể gần gũi với nhiều người hơn, giúp mọi người giải trí hoặc tìm kiếm những cảm giác mới mẻ.

Nhóm đã chọn làm game Binh nhí(tựa game do nhóm sang chế) một game nhập vai hiện đại và gần gũi.

Với kiến thức về lập trình hướng đối tượng và có sự hỗ trợ của thầy, bè bạn nhóm đã hoàn thành game Binh nhí này. Tuy nhiên trong game vẫn còn nhiều thiếu sót do kinh nghiệm làm game còn chưa đủ, nhưng nhóm đã cố gắng hết mình đê hoàn thành những ý tưởng cơ bản mà một game cần phải có. Nhìn chung việc làm game ban đầu cần một ý tưởng và sau đó triển khai ý tưởng đó qua các dòng lệnh. Nhờ sự hỗ trợ của thư viện DirectX xử lý các Input và Graphics người lập trình chỉ cần chú ý vào cách sử dụng các giải thuật đã được học để triển khai và xử lý các tình

huống trong Game.

Nhóm xin chân thành cảm ơn sự quan tâm của thầy

Nhóm thực hiện:

Trương Mai Thanh Trí 09520320

Nguyến Minh Vượng 09520364

Vũ Minh Tuấn 09520333

Mục Lục

[II. Giới thiệu 5](#_Toc313939381)

[III. Cấu trúc chương trình 6](#_Toc313939382)

[A. Cấu trúc chương trình 6](#_Toc313939383)

[B. Cấu trúc một số class quan trọng 7](#_Toc313939384)

[1. WinMain 7](#_Toc313939385)

[2. DXDevice 7](#_Toc313939386)

[3. GameApp 7](#_Toc313939387)

[4. iPay 8](#_Toc313939388)

[5. State 8](#_Toc313939389)

[6. RSManager : 9](#_Toc313939390)

[7. SoundGame 9](#_Toc313939391)

[8. CRECT 9](#_Toc313939392)

[9. Terrain 10](#_Toc313939393)

[10. InfoSprite 10](#_Toc313939394)

[IV. Cấu trúc State 10](#_Toc313939395)

[V. Cấu trúc Object 12](#_Toc313939396)

[A. Sơ đồ các đối tượng trong game 12](#_Toc313939397)

[B. Cấu trúc đối tượng 13](#_Toc313939398)

[1. Đối tượng MyObject 13](#_Toc313939399)

[2. Đối tượng Character 13](#_Toc313939400)

[3. Đối tượng Monster: 14](#_Toc313939401)

[4. Đối tượng Items : Kế thừa lớp Object. Lớp cung cấp các đối tượng sau: 14](#_Toc313939402)

[5. Đối tượng BackGround 15](#_Toc313939403)

[6. Đối tượng Boss 18](#_Toc313939404)

[VI. Design Patter sử dụng trong game 19](#_Toc313939405)

[A. Singleton 19](#_Toc313939406)

[B. Ứng dụng singleton trong game : 19](#_Toc313939407)

[1. Quản lý resource ảnh : 19](#_Toc313939408)

[2. Quản lý Sound trong game : 19](#_Toc313939409)

[3. Quản lý Device trong game : 20](#_Toc313939410)

[VII. Particle trong game 20](#_Toc313939411)

[VIII. 6.Camera trong game 21](#_Toc313939412)

[IX. Trong hàm này có sử dụng một phương thức : 23](#_Toc313939413)

[X. Phân hoạch không gian với thuật QuadTree 23](#_Toc313939414)

[A. Vấn đề 23](#_Toc313939415)

[B. Ý tưởng 23](#_Toc313939416)

[C. Có hai cách phân chia phổ biến 24](#_Toc313939417)

[1. Cách một 24](#_Toc313939418)

[2. Cách hai 24](#_Toc313939419)

[D. Cài đặt 25](#_Toc313939420)

[1. Class QuadTreeNode 25](#_Toc313939421)

[E. Sử dụng 26](#_Toc313939422)

[XI. Map và mapeditor 26](#_Toc313939423)

[A. Cấu trúc map 26](#_Toc313939424)

[B. Load map 26](#_Toc313939425)

[C. Mapeditor 27](#_Toc313939426)

[1. Giao diện 27](#_Toc313939427)

[2. Tính năng 29](#_Toc313939428)

[XII. Giao diện 29](#_Toc313939429)

[XIII. Đánh giá hướng mở rộng : 36](#_Toc313939430)

[A. Đánh giá 36](#_Toc313939431)

[1. Ưu điểm 36](#_Toc313939432)

[2. Nhược điểm 36](#_Toc313939433)

[B. Hướng mở rộng 36](#_Toc313939434)

# Giới thiệu

Bắt đầu từ một ngôi trường đào tạo chiến binh, các chiến binh được đào tạo ở đây chủ yếu là những bạn nhỏ nhưng có lòng dũng cảm, nhân đạo rất đánh quý, các chiến binh nhỏ này được gọi là “Binh nhí”.

Trường đạo tạo binh nhí là nơi tràn đầy hạnh phúc và vui vẻ. Hằng ngày các “binh nhí” phải rèn luyện sức khỏe tập luyện các skill….

Nhưng các bí tịch của trường thì lại rơi vào tay của bọn đạo tặc phản loạn, điều kinh khủng hơn là đạo tặc phản loạn này là cấu kết với những thế lực xấu trong thế giới như bọn cướp biển, lũ phù thủy, lâm tặc, … những thế lực xấu này đã tu luyện thành công các bí tịch của trường, và đang thực hiện một âm mưu lớn.

Một ngày nọ, bỗng dưng bọn xấu đồng loạt tấn công các ngôi làng, chúng cướp bóc tiền bạc và nông sản, lấy hết trâu bò, nông cụ của người dân làm mọi người hoang mang lo sợ, cuộc sống trở nên khó khăn…

Trước cảnh lâm nguy của dân làng các bô lão và binh nhí trong trường không thế đứng nhìn được, và đã tuyển chọn ra một binh nhí xuất sắc nhất, dũng cảm nhất, và đầy lòng nhân ái…

Binh nhí được chọn xuất trường, với nhiệm vụ đánh tan bọn cướp bóc mang lại cuộc sống bình yên cho dân làng, nhưng binh nhí cũng không quên việc đánh bại bọn đạo tặc phản loạn để lấy lại các bí tịch của binh nhí và đó cũng là lúc binh nhí tu luyện được bí tịch.

# Cấu trúc chương trình

## Cấu trúc chương trình

Chương trình Game được xậy dựng theo cấu trúc FrameWork, trong đó chia thành 3 phần chính là:

* Environment
  + Quản lý các vần đề liên quan tới hệ điều hành, flatform, phần cứng… Là cầu nối trung gian giữa môi trường xử lý, các thiết bị nhập xuất và các xử lý trong game chính.
  + Environment không làm ảnh hưởng tới Game chính, tạo tính độc lập trong xử lý Game, thuận tiện cho việc port game.
  + Các vấn đề chính được sự lý ở Environment là:
    - Khởi tạo ứng dụng.
    - Chọn và khởi tạo thiết bị đồ hoạ.
    - Khởi tạo input.
    - Tạo vòng lặp cho game.
* FrameWork
  + Chứa các thành phần trong game bao gồm các Object, Cấu trúc, các lớp, các giải thuật.v.v.v
  + Thực hiện quản lý Resourcebằngsingleton.
  + Cung cấp các giải thuật trong game.
  + Xây dựng các lớp đối tượng.
* GamePlay
  + Thiết kế Game thực hiện tại đây, tất cả các thao tác từ đầu game đến kết thúc đều thực hiện ở GamePlay.
  + GamePlay được chia nhỏ thành các State (màn chơi) để dễ quản lý.

Cấu trúc FrameWork phải đảm bảo các yêu cầu sau:

* Tách biệt độc lập xử lý và đồ hoạ, các xử lý trong game không phụ thuộc vào môi trường.
* Quản lý tài nguyên chặt chẽ, sử dụng chung các Resource, trong mỗi state game ( màn chơi ), chỉ khởi tạo Resource, cấp phát bộ nhớ một lần, hạn chế cấp phát, khởi tạo nhiều lần.
* Có các lớp cơ chế xử lý collision, physics, logic …độc lập.
* Dễ dàng sử dụng các giải thuật quản lý game như A\*, list…
* Các Object trong game phải thừa kế từ một Object thuần ảo và thiết kế có tính hướng đối tượng cao.
* Cơ chế quản lý FPS và input tốt.
* Nâng cao tính độc lập của Environment và GamePlay.

## Cấu trúc một số class quan trọng

### WinMain

Chức năng khởi tạo chương trình Game, điều kiểu tất cả chương trình.

### DXDevice

Chức năng khởi tạo DirectX, xác định cấu hình thích hợp với máy. Định nghĩa các thông số cần thiết như độ phân giải. chất lượng màu..v.v.v..

Thành phần

LPDIRECT3D9 m\_pD3D;

D3DPRESENT\_PARAMETERS m\_pD3DPP,mpp;

LPDIRECT3DDEVICE9 m\_pD3DDev;

LPDIRECT3DSURFACE9 m\_pBackBuffer;

Phương thức

bool Create\_DirectX ();

Chức năng khởi tạo LPDIRECT3D9 m\_pD3D;

bool Create\_Para (bool \_iFullmode,HWND \_hwnd);

Chức năng khởi tạo D3DPRESENT\_PARAMETERS m\_pD3DPP; bao gồm các thông số thích hợp

bool Create\_Device (HWND \_hwnd);

Chức năng khởi tạo LPDIRECT3DDEVICE9 m\_pD3DDev; để thực hiện các thao tác draw

### GameApp

void ProcessMouse ():Xử lý nhận sự kiện chuột

void ProcessKeyBoard():Xử lý nhận sự kiện phím

void Update (float \_DetaTime) Xử lý update ( tinh toán logic vật lý)

Xử lý Draw (hiển thị game lên màn hình)

### iPay

Chức năng quản lý màn hình chơi, xác đinh state hiện tại để thực hiện draw và update. Khi chuyển sang state khác thực hiện hủy state hiện tại khởi tạo state mới….

Thành phần

iState\* m\_CurState;

iState\* m\_NextState;

LPDIRECT3DDEVICE9 m\_Device;

Phương thức

void SetCurState (iState\* \_curstate);

void SetNextState (iState\* \_nextstate);

iState\* GetCurState ();

iState\* GetNextState ();

void SetDevice (LPDIRECT3DDEVICE9 Device);

LPDIRECT3DDEVICE9 GetDevice ();

virtual void Init ()=0;

virtual void Update (float \_Time)=0;

virtual void Draw ()=0;

virtual void Release ()=0;

Hoạt động khi chuyển state

if(m\_CurState->GetIDState()!=m\_NextState->GetIDState())

{

m\_CurState->Release();

m\_CurState=m\_NextState;

m\_CurState->Init();

}

else

{

m\_CurState->Update(\_Time);

}

### State

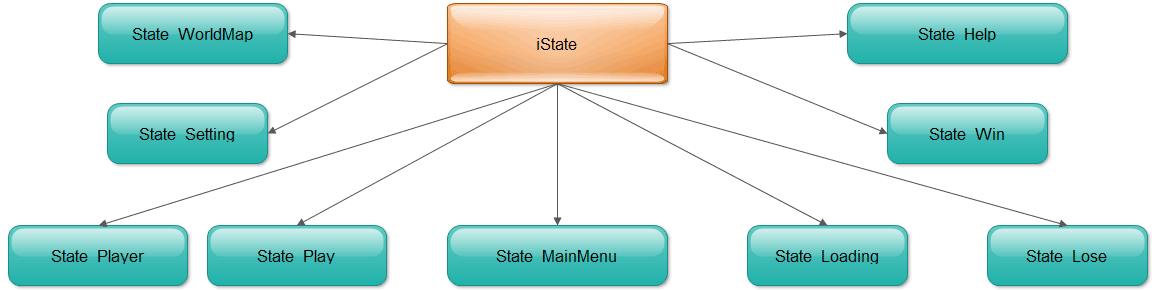
Tất cả xuất hiện trong game đều được thực hiện từ các state được thừa kế từ class iState.

Ở đây sử dụng mẫu design pattern để quản lý các state nhầm giúp tách rời các state ra giống như các thành phần riêng biệt.

Giúp cho nhóm dễ dàng phát triển Game, tránh bị lỗi dây chuyền.

Mỗi state đều có các phương thức chính

* Init () Khởi tạo state bao gồm khởi tạo các hình ảnh các biến, tránh trường hợp vừa sử dụng resource vừa khởi tạo.
* Update () Xử lý các logic vật lý AI.
* Draw () Hiển thị kết quả lên màn hình.
* Release () Hủy các biến, hình ảnh resource.
* IsKeyDown () Xử lý khi nhấn phím
* OnKeyDown () Xử lý khi nhấn phím
* OnKeyUp () Xử lý khi nhấn phím
* ProcessMouse() Xử lý chuột



### RSManager :

Quản lý Sprite trong StateMainGame.

Có nhiệm vụ quản lý chặt chẽ các Sprite sao cho mỗi hình ảnh chỉ tạo một Sprite duy nhất. Tiết kiệm bộ nhớ tối đa cho việc cấp phát Resource.

Được xây dựng theo cấu trúc designPattern Singleton, nhằm đảm bảo tính duy nhất, tiết kiệm bộ nhớ tối đa, tránh trường hợp khởi tạo nhiều lần.

Các thuộc tính:

* Các Sprite mà RSManager quản lý.
* private static RSManager instance : là một con trỏ, trỏ tới đối tượng thực thể, thể hiện cho cả lớp RSManger.

Các phương thức chính:

* protected RSManager(Game \_Game): constructor là phương thức protected không cho khởi tạo bên ngoài ( bên ngoài không được gọi hàm new để tạo đối tượng RSManager).
* public static RSManager Instance(Game \_Game) : trả lại con trỏ thể hiện cho lớp RSManager. (nếu instance chưa tồn tại thì tạo instance mới, nếu instance đã tồn tại thì không tạo mới nữa).

### SoundGame

Cũng tương tự như RSManager như dùng để quản lý Sound trong game.

### CRECT

Mô tả một hình chữ nhật dùng trong các object trong game.

Sử dụng để xác định size của nhân vật. tính toán va chạm.

Có hai phương thức chính

* iCollision (CRECT) Xác định xem có va chạm với một hình chữ nhật khác không.
* iContains (CRECT) Xác định xem có chứa toàn bộ một hình chữ nhật không.

### Terrain

Địa hình trong game được lưu trữ trong hai mảng hai chiều.

Một mảng m\_Terrain lưu trữ các địa hình xem có khả dụng hay không

Một mảng m\_TerrainExtends lưu trữ các background của game không sử dụng để tính toán va chạm.

### InfoSprite

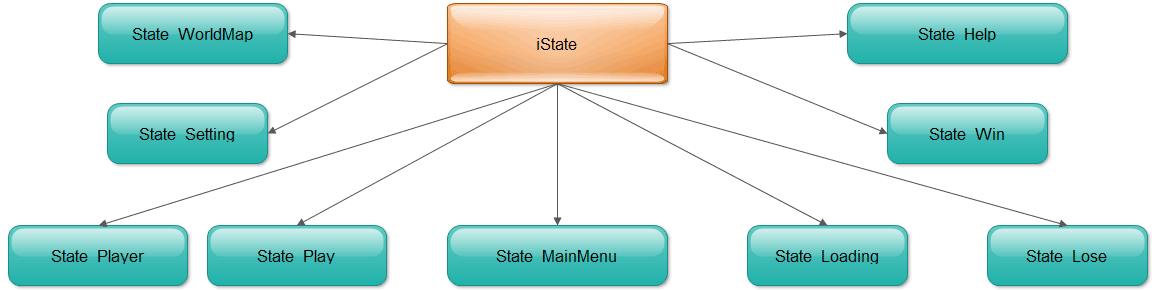
Chức năng lưu trữ thông tin của sprite bao gồm chiều rộng, chiều dài, vị trí, frame hiện tại, độ sâu, độ xoay, .v.v.v nhầm hỗ trợ draw.

Phương thức chính

NextFrame (): animation frame

getMatrixTransformSprite (): Từ các thông số của info\_sprite sử dụng phép toán D3DXMatrixTransformation2D() để tạo một Matrix, sử dụng Matrix này nhân với Matrix World sẽ ra Matrix cuối cùng để sử dụng cho kỹ thuật MatrixTranfroms.

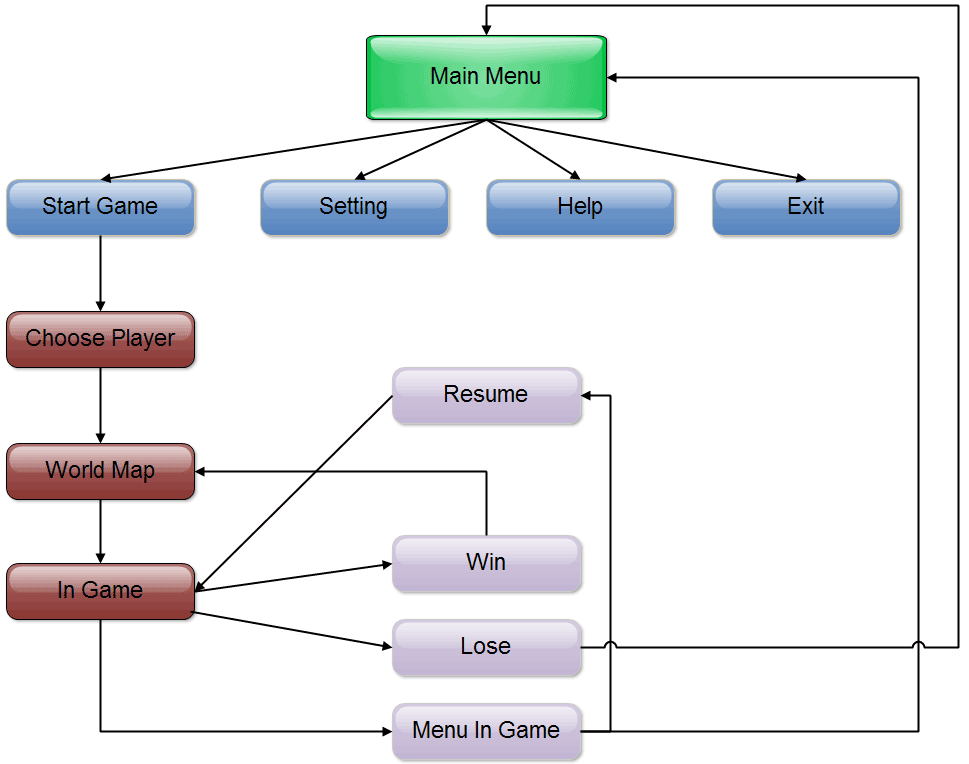
# Cấu trúc State



*Hình 9: Cấu trúc các State.*

* Các State được kế thừa từ Class *iState .*
* Mỗi State thực hiện một chức năng riêng:
* State MainMenu: Chọn thao tác chính
* State Setting: Cài đặt trong game như Âm thanh
* State Help: Cung cấp thông t in về nhóm
* State Player: Chọn Player cho trò chơi
* State Play: Vào game chính
* State WorldMap: Chọn địa điểm chơi
* State Loading: Load các tài nguyên cho game
* State Win: Load khi chiến thắng một màn chơi
* State Lose: Load khi bị thua cuộc

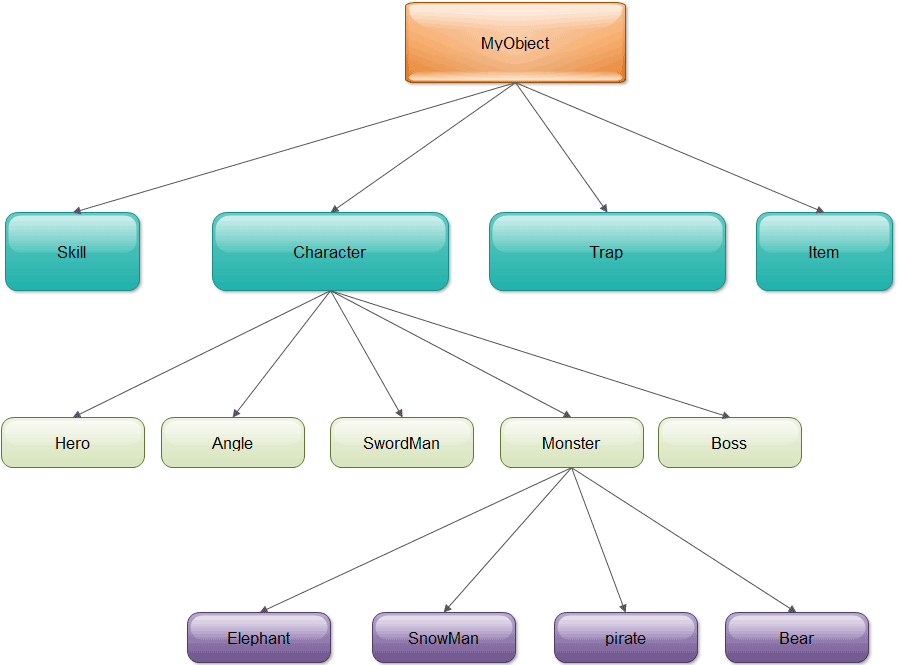
## Game Flow



*Hình 10: Sơ đồ Game Flow*

# Cấu trúc Object

## Sơ đồ các đối tượng trong game



*Hình 11: Sơ đồ các đối tượng trong Game*

* Các đối tượng trong game đều kế thừa ( trực tiếp hay gián tiếp) từ class Object. Class object cung cấp các thao tác cơ bản cho các đối tượng khác kế thừa: Move, khởi tạo, hủy…
* Các đối tượng chính như: Skill, character… kế thừa trực tiếp từ object.
* Các đối tượng tiếp theo có các đặc tính chung được kế thừa từ các đối tượng chính:
* Hero, Angle, Boss… kế thừa từ character.
* Bear, pirate … cùng bản chất Monster kế thừa từ Monster.

## Cấu trúc đối tượng

### Đối tượng MyObject

Là một class thuần ảo dùng để dành cho các Object khác kế thừa

### Đối tượng Character

Nhân vật chính trong game có thể chọn 1 trong 3 nhân vật : Angle, SwordMan, Hero.

|  |  |  |
| --- | --- | --- |
| Nhân vật | Loại Nhân vật | Mô tả |
| C:\Users\Tuan\Desktop\Choose_Magician.png | Angle | HP(máu) : 50000.  Khả năng di chuyển : đi, nhảy  Khả năng chiến đấu :  Skill 1: Dùng gậy ánh sáng để tấn công kẻ thù .  Skill 2 : Triệu hồi Thánh thú từ địa ngục để tấn công kẻ thù, Có Damage là 1500 .  Skill 3 : Gọi Kim tự tháp huyền bí phóng ra năng lượng ánh sáng để tiêu diệt kẻ thù , có Damage là 2000 .  Skill 4 : Gọi Sâm sét từ trên trời đánh xuông để tiêu diệt kẻ thù, có Damage là 3000 . |
| C:\Users\Tuan\Desktop\Choose_Punch.png | Hero | HP(máu) : 50000.  Khả năng di chuyển : đi, nhảy.  Khả năng chiến đấu :  Skill 1: Dùng năng lượng bản thân để tấn công kẻ thù trong diện rộng, có Damage là 5000.  Skill 2: Tạo ra 1 đàn cá mập để tấn công kẻ thù trong diện rộng, có Damage là 1500 .  Skill 3: Dùng năng lượng bản thân, nhảy lên rồi đấm mạnh xuống đất để tạo ra chấn động gây sát thương trong diện rông, có Damage là 2000.  Skill 4 : Tung cú đấm thôi sơn tấn công đối thủ, Có Damage là 400 . |
| C:\Users\Tuan\Desktop\Choose_Sword.png | SwordMan | HP : 50000.  Khả năng di chuyển : đi, nhảy .  Khả năng chiến đấu :  Skill 1: Sử dụng kiêm tấn công kẻ thù, có Damage là 700 .  Skill 2: Tạo ra Băng Tuyết để tấn công kẻ thủ, có Damage là 1500 .  Skill 3 : Gọi Rồng thần đi theo để hỗ trợ tấn công , có Damage là 300 .  Skill 4 : Gọi Năng lượng Sấm Sét để tấn công kẻ thù, có Damage là 3000 . |

### Đối tượng Monster:

Đối tương Monster có AI(Artificial Intelligence) : Khi mà khoảng cách giữa Nhân vật chính của mình và Monster nhỏ hơn 500 nó sẽ truy sát nhân vật (nhân vật đi hướng nào nó đi hướng đó). Nếu khoảng cách giữa Nhân vật và Monster nhỏ hơn 1 khoảng vừa đúng rectangle của skill , nó sẽ tung skill để tấn công nhân vật .

|  |  |  |
| --- | --- | --- |
| Loại Monster | Kind | Mô tả |
|  | Monster | Con gấu có Lực tấn công : 300  Máu(HP) : 2100 .  Số lượng skill : 1 .  Range skill : 250 . |
| C:\Users\Tuan\Desktop\Monster\Elephant\attack1_0.png | Monster | Voi có lực tấn công : 300  Máu(HP) : 2100  Số lượng skill : 1 .  Range Skill : 200 . |
| C:\Users\Tuan\Desktop\Monster\SnowMan\attack1_0.png | Monster | Người tuyết có lực tấn công : 300  Máu(HP) : 2100  Số lượng skill : 1 .  Range Skill : 250 . |
| C:\Users\Tuan\Desktop\Private\attack1_0.png | Monster | Cướp biển có lực tấn công : 300  Máu(HP) : 2100  Số lượng skill : 1 .  Range Skill : 250 . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\Trap.png | Trap | Loại bẫy, nếu nhân vật chạm vào sẽ mất 100 máu . |

### Đối tượng Items : Kế thừa lớp Object. Lớp cung cấp các đối tượng sau:

|  |  |  |
| --- | --- | --- |
| Loại Item | Kind | Mô tả |
| C:\Users\Tuan\Desktop\Item\0.hit_0.png | Item | Có thể Animation .  Cộng máu cho nhân vật khi ăn Item này . |
| C:\Users\Tuan\Desktop\IncreaseDamage\bullet_0.png | Item | Có thể Animation .  Công 500 Lực sát thương cho Skill nhân vật trong vòng 20 giây . |
| C:\Users\Tuan\Desktop\Bone\bullet_0.png | Item | Có thể Animation .  Trừ 3000 máu của nhân vật khi ăn phải đầu lâu . |

### Đối tượng BackGround

|  |  |
| --- | --- |
| Loại cảnh vật | Mô tả |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\100.png | Mây trên trời . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\101.png | Mây trên trời. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\102.png | Mây trên trời . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\104.png | Mây trên trời. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\105.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\106.png | Cây trang trí hình nấm . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\107.png | Cây trang trí hình ốc sên. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\108.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\109.png | Cây trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\110.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\111.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\112.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\113.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\114.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\117.png | Cây trang trí. |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\118.png | Tượng trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\119.png | Nhà trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\120.png | Hoa trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\121.png | Tượng trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\122.png | Cổng trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\123.png | Hoa trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\124.png | Tượng Nhân Sư |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\125.png | Hoa trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\126.png | Lều trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\128.png | Nhà trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\129.png | Cổng trang trí . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\130.png | Tượng Nhân Sư . |
| C:\Users\Tuan\Desktop\BinhNhi_ver4\[DirectX]_Game2D_BIGSE\[DirectX]_Game2D_BIGSE\data\image\131.png | Cầu thang trang trí . |

### Đối tượng Boss

Đối tương Boss có AI(Artificial Intelligence) : Khi mà khoảng cách giữa Nhân vật chính của mình và Monster nhỏ hơn một khoảng cho phép nó sẽ truy sát nhân vật (nhân vật đi hướng nào nó đi hướng đó). Nếu khoảng cách giữa Nhân vật và Monster nhỏ hơn 1 khoảng vừa đúng rectangle của skill , nó sẽ tung skill để tấn công nhân vật .Mỗi Boss có 4 skill, tung hết skill này sẽ random để tung skill khác

|  |  |  |
| --- | --- | --- |
| Loại Boss | Tên Boss | Mô tả |
| C:\Users\Tuan\Desktop\BannedStory_SpriteSheet\move_0.png | SnakeMan : Phù thủy bóng tối điều khiển một con ác xà. | HP : 60000  Khả năng di chuyển : đi.  Khả năng chiến đấu :  Skill 1 : Điều khiển rắn tấn công người chơi .  Skill 2: Gọi đôi cánh bóng tối tấn công người chơi .  Skill 3: Gọi lửa địa ngục tấn công người chơi  Skill 4: Gọi năng lượng bóng tối tấn công người chơi . |
| C:\Users\Tuan\Documents\attack1_11.png | Thánh thú Von Leon | HP : 40000 .  Khả năng di chuyển : đi  Khả năng chiến đấu :  Skill 1: Tung 1 quả đấm sấm sét gây chấn động địa cầu . Damage : 1400 .  Skill 2: Sử dụng móng vuốt để tấn công người chơi. Damage : 2000 .  Skill 3: Sư tử hống. Một tiếng hét vang dậy đất trời gây choáng váng cho người chơi. Damage : 1800 .  Skill 4 :Bắn Laze từ mắt, lực sát thương mạnh. Damage : 2500 . |

# Design Patter sử dụng trong game

## Singleton

là một class điều hành lớp qua phương thức static.Nó chỉ cho phép ta tạo một đối tượng duy nhất.

## Ứng dụng singleton trong game :

### Quản lý resource ảnh :

Class RSMainGame : Quản lý tất cả sprite trong MainGame (màn hình chơi chính của mình), có các phương thức trả về Sprite, ví dụ:

Sprite\* getCharacter (){

*return m\_Character;*

### Quản lý Sound trong game :

Class SoundGame : Quản lý tất cả sound trong Game.

Có các phương thức hỗ trợ Play Sound Effect và Play Music, mỗi khi cần play sound thì chỉ cần gọi phương thức này ra ở bất kì đâu .

Có các phương thức để pause hoặc stop hoàn toàn music và sound effect trong game .

### Quản lý Device trong game :

Class DXManager :Có các phương thức để get hoặc set Device và HWND(HandleWindows) .

Ưu điểm : Lợi ích của việc sử dụng singleton là rất linh hoạt trong vấn đề thêm hoặc xóa 1 đối tượng va dễ quản lý object ,sound, device, sprite, ảnh .

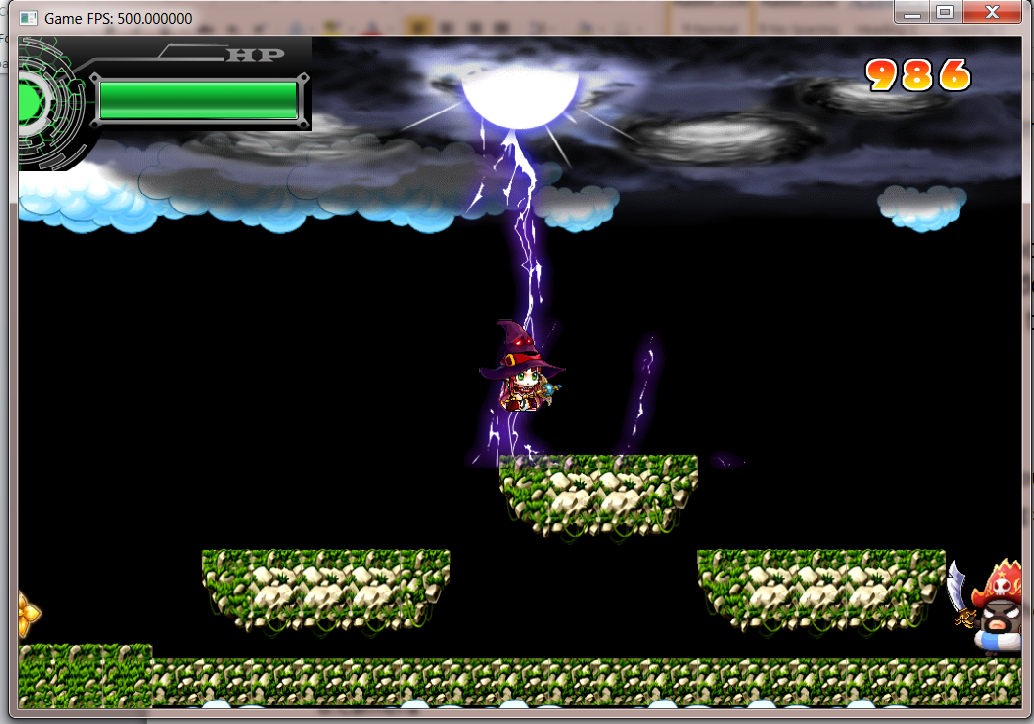
# Particle trong game

Trong game có sử dụng particle tuyết rơi ở màn hình mainmeunu.

Class: Effect\_MainMenu .

Particle tuyết rơi :Rơi từ trên xuống dưới và có bay dao động theo trục x . Mỗi bông tuyết có thời gian sống, nếu hết thời gian sống sẽ tự động biến mất, nếu xuất hiện ở ngoài màn hình thì sẽ được Refresh để xuất hiện lại trong màn hình .Số lượng Object trong list Particle là 50 .

# 6.Camera trong game



Trong game có sử dụng Camera để quan sát game, đi theo nhân vật chính của game .

class Camera

{

private:

float m\_Width ;

float m\_Height;

float m\_X, m\_Y;

float m\_Offset;

float m\_Dy,m\_VDy;

float m\_SC;

public:

Camera(void);

Camera (float \_X, float \_Y,float \_Width,float \_Height);

~Camera(void);

float getX ()

{

return m\_X;

};

float getY ()

{

return m\_Y;

};

float getWidth ()

return m\_Width;

};

float getHeight ()

{

return m\_Height;

};

CRECT getRect()

{

CRECT r;

r.Left = m\_X - m\_Offset ;

r.Top = m\_Y - m\_Offset;

r.Right = m\_X + m\_Width + m\_Offset;

r.Bottom = m\_Y + m\_Height + m\_Offset ;

return r;

};

void Update(Character\* \_Character,float \_MaxWidth,float \_MaxHeight);

void UpdateEffect (float \_Time);

void NoEffect ();

D3DXMATRIX getMatrixTransform ();

};

Trong Class camera có các phương thức:

getRect(): Trả về Rect của Camera .

Update(): Cho Camera chạy theo nhân vật, luôn đặt nhân vật vào trung tâm màn hình .

UpdateEffect() : Hiệu ứng làm rung màn hình của Camera .

NoEffect() : Dừng hiệu ứng rung cho camera .

D3DMATRIX getMatrixTransform() :

D3DXMATRIX Camera::getMatrixTransform ()

{

D3DXMATRIX mtFinal,mtCamera,mtEffect;

D3DXVECTOR2 CenterEffect (m\_X + g\_SCREEN\_WIDTH/2,m\_Y +g\_SCREEN\_HEIGHT/2);

D3DXVECTOR2 TranEffect (0,0);

D3DXVECTOR2 ShakeEffect(m\_Dy,0);

D3DXVECTOR2 ScaleEffect (m\_SC,m\_SC);

D3DXMatrixTransformation2D(&mtEffect,&CenterEffect,1,&ScaleEffect,NULL,NULL,&TranEffect);

D3DXVECTOR2 TransCamera (-m\_X,-m\_Y); //

D3DXMatrixTransformation2D(&mtCamera,NULL,0,NULL,NULL,NULL,&TransCamera);

mtFinal = mtEffect\*mtCamera;

return mtFinal;

}

# Trong hàm này có sử dụng một phương thức :

D3DXMatrixTransformation2D(&mtEffect,&CenterEffect,1,&ScaleEffect,NULL,NULL,&TranEffect);

D3DXMatrixTransformation2D(&mtCamera,NULL,0,NULL,NULL,NULL,&TransCamera);

* &mtEffect :trỏ tới matrix chứa kết quả của việc transform .
* &CenterEffect : Đảm bảo cho nhân vật luôn ở giữa màn hình.
* 1 : Tỷ lệ giữa zoom và rotation .
* % ScaleEffect : Để zoom ra, zoom vào Camera .
* Rotation Center : trỏ tới một structure D3DXVECTOR2. Xác định trung tâm xoay màn hình.
* Rotation : xác định góc xoay băng radian .
* &TranEffect: trỏ tới một structure D3DXVECTOR2, xác định giá trị transform .

# Phân hoạch không gian với thuật QuadTree

## Vấn đề

Khi cần xét xác định va chạm để xử lý tính toán trong game ta cần xét đối tượng đó với tất cả các đối tượng khác trong game nên cần xử lý gần bằng n\*(n-1), điều này là vô cùng bất cập khi số lượng đội tượng trong game vô cùng lớn mà lại có nhiều tính toán xử lý dư thừa.

Đối với một game cuộn map 2D người chơi thường sẽ được chạy từ đầu đến cuối map. Các đối tượng trong một map hầu như không cần chuyển động xét va chạm khi không nhín thấy.

Đều này nhầm đảm bảo sẽ luôn giữ được địa hình cho đến khi người chơi di chuyển tới mỗi khu vực.

## Ý tưởng

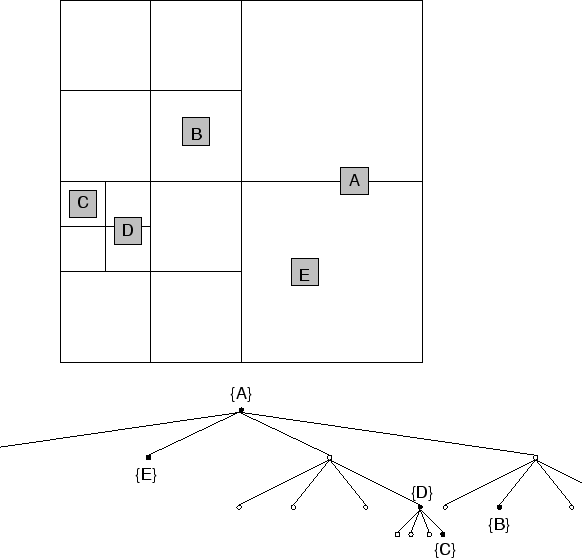
Không gian được chia ra làm bốn phần bằng nhau, sử dụng đệ quy để tính toán chi xuất đến tất cả các đối tượng trong các node.

Việc phân chia node và đối tượng trong quadtree phục thuộc vào cách quản lý riêng.

## Có hai cách phân chia phổ biến

### Cách một

Cách một phân chia phụ thuộc và số lượng đối tượng tại một node

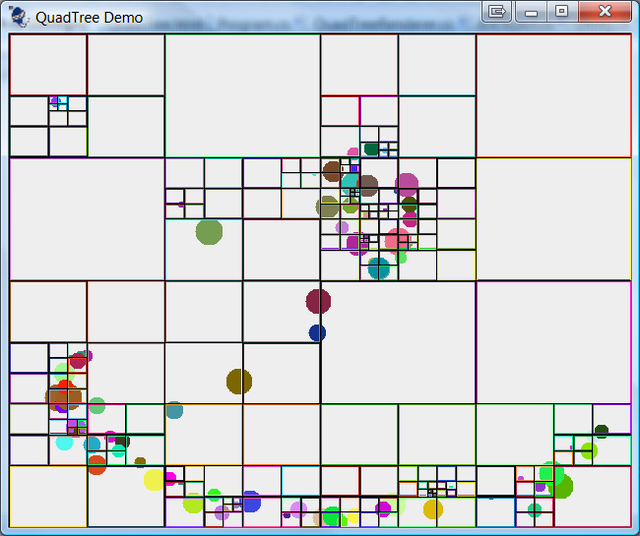


Ưu điểm không cần phải chia node quá nhiều lần, tránh trường hợp phải tạo quả nhiều node khi chi có một Object.

Nhược điểm với số lượng đối tượng lớn cần phải tạo rất nhiều node để lưu trữ tương ứng.

### Cách hai

Cách hai phân chia phục thuộc vào vị trí của đối tượng với nguyên tắt nếu node con không bao toàn bộ đối tượng thì nude mẹ sẽ lưu trữ đối tượng



Ưu điểm truy xuất đối tượng nhanh, thích hợp với các mô hinh có nhiều đối tượng

Nhược điểm một số trường hợp phải tọa node mới lưu trữ được đối tượng.

## Cài đặt

Cấu trúc đối tượng trong game tương đối nhiều nên nhóm sử dụng cách hai để xay dựng quadtree.

### Class QuadTreeNode

Class quadtreenode dùng để xác định từng node của quadtree, trong class quadtreenode

Sẽ có

* m\_Contents (cấu trúc list hoặc vector): lưu trữ tất cả đối tượng trong node đó.
* m\_Child mảng 4 quadtreenode con.
* m\_Bounds Kích thướt của một node.

Phương thức

* CreateSubNodes (): tạo 4 node con khi cần thiết (lúc cần đưa một đối tượng vào node đó).
* Insert (): Đưa một đối tượng vào quadtree sử dung đệ qui để xác định chính xác vị trí cần đưa vào.
* Query (CRECT): Lấy toàn bộ đối tượng trong quadtree va chạm với vùng camera.

## Sử dụng

QuadTree có tác dụng lưu trữ các đối tượng ra khởi vùng camera nhìn thấy, các đối tượng này sẽ không được xét va chạm hay update.

Cần kết hợp hài hòa liên tục giữa camera và quadtree, sau khi lấy các object va chạm với camera ra khỏi quadtree. Ta cần xem xét object còn sống hay không nếu không thì hủy object đó đi. Trong trường hợp object đó ra khỏi camera thì ta lập tức đưa object đó vào quadtree lại.

# Map và mapeditor

## Cấu trúc map



Game cuộn map nên có cấu trúc map là một mảng hai chiều để lưu trữ id của từng object tương ứng tại vị trí đó.

Cấu trúc map là một file ảnh png. Với kênh alpha là time game. Kênh red là id tương ứng của các object.

## Load map

Việc loadmap được thực hiện thông qua class LoadMap.

Có 2 phương thức chính là

* LoadMapFormFile: đọc map từ một file png lưu lại với dạng Texture lưu ý cần cấp phát quyền truy cập lại file Texture để thực hiện dịch map.
* TranslateMap: dịch map, tạo các object tương ứng trên map rồi đưa vào game.

## Mapeditor

### Giao diện

#### C:\Users\ThanhTri\Desktop\new.pngMain

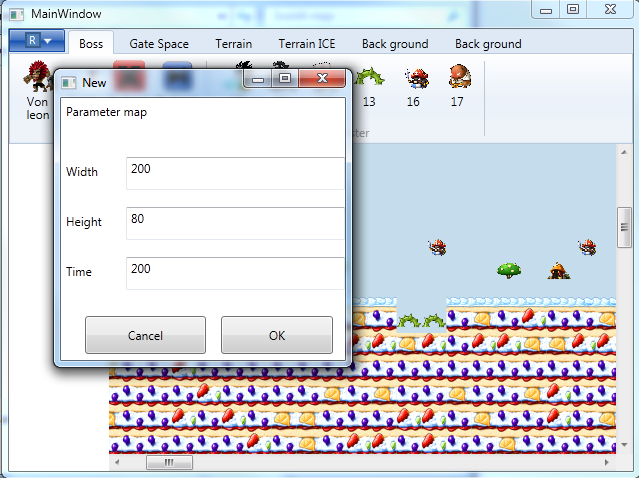
New map:

Thông số Width, Height, Time

Nhấn OK để xác nhận tạo một map mới có kích thước như khai báo

Nhấn Cancel để hủy lệnh.

#### edit



Edit map:

Thông số Width, Height, Time

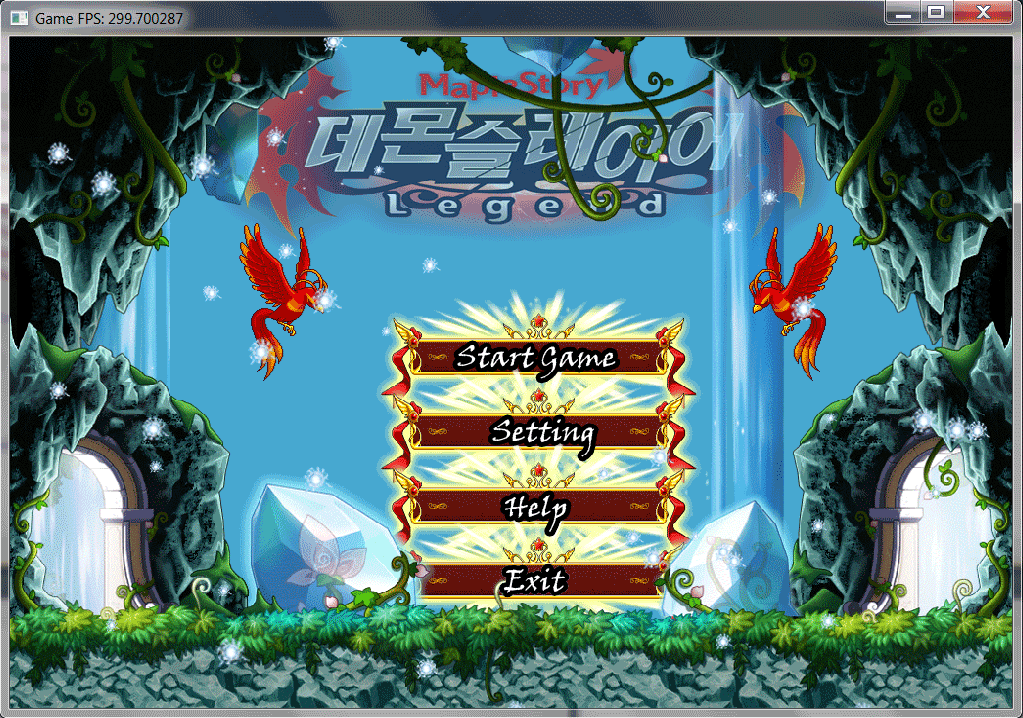
Nhấn Cancel để hủy lệnh.

Nhấn OK để xác nhận tạo một map mới có kích thước như khai báo, sau đó sẽ copy dữ liệu từ map cũ sang map có size mới.

### C:\Users\ThanhTri\Desktop\ui.pngTính năng

# Giao diện

***Main Menu***



*Hình 1: Màn hình MainMenu*

* Trong màn hình MainMenu, chọn một trong các chức năng:
* Start Game: Bắt đầu game
* Setting: Cài đặt âm thanh
* Help: Thông tin nhóm
* Exit: Thoát game

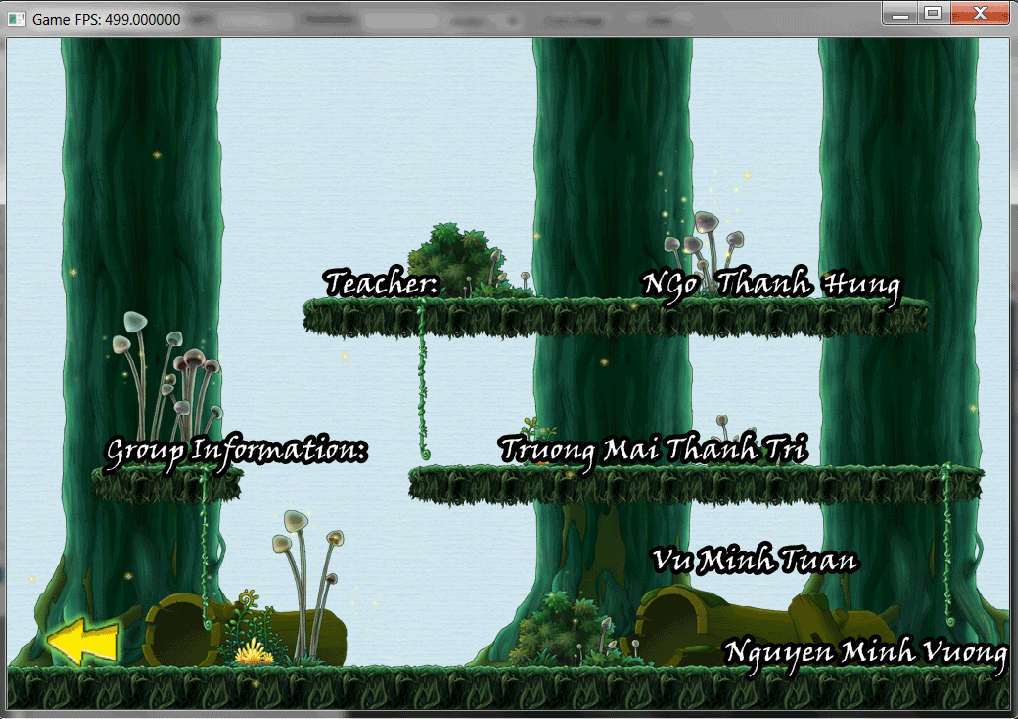
1. ***Setting***



*Hình 2: Màn hình Setting*

* Trong màn hình Setting: Chọn bật tắt tất cả âm thanh
* Có dấu “√”: Bật âm thanh.
* Không có: Tắt âm thanh.
* Sau khi cài đặt, nhấn mũi tên để trở lại MainMenu.

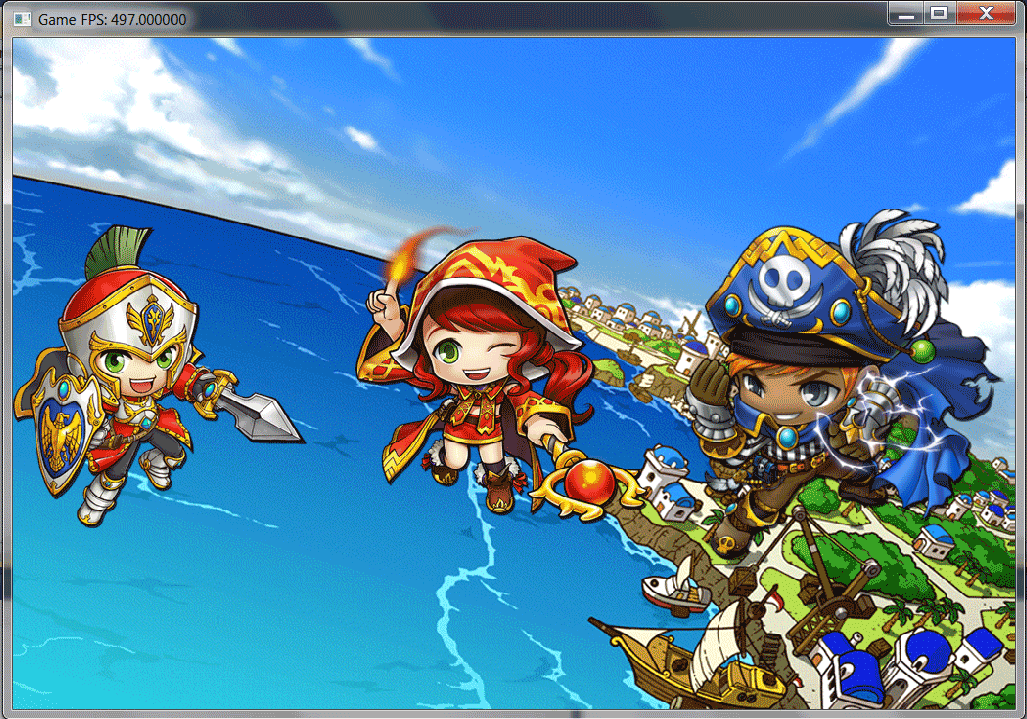
1. ***Help***



*Hình 3: Màn hình Help*

* Màn hình chứa thông tin nhóm.
* Nháy vào mũi tên để về MainMenu

1. ***Chọn nhân vật***



*Hình 4: Màn hình chọn nhân vật*

* Trong màn hình chọn nhân vật, chọn một trong 3 nhân vật mà bạn thích chơi:
* Hero
* Angle
* SwordMan
* Sau khi chọn, game chuyển sang màn chơi chính với nhân vật được chọn

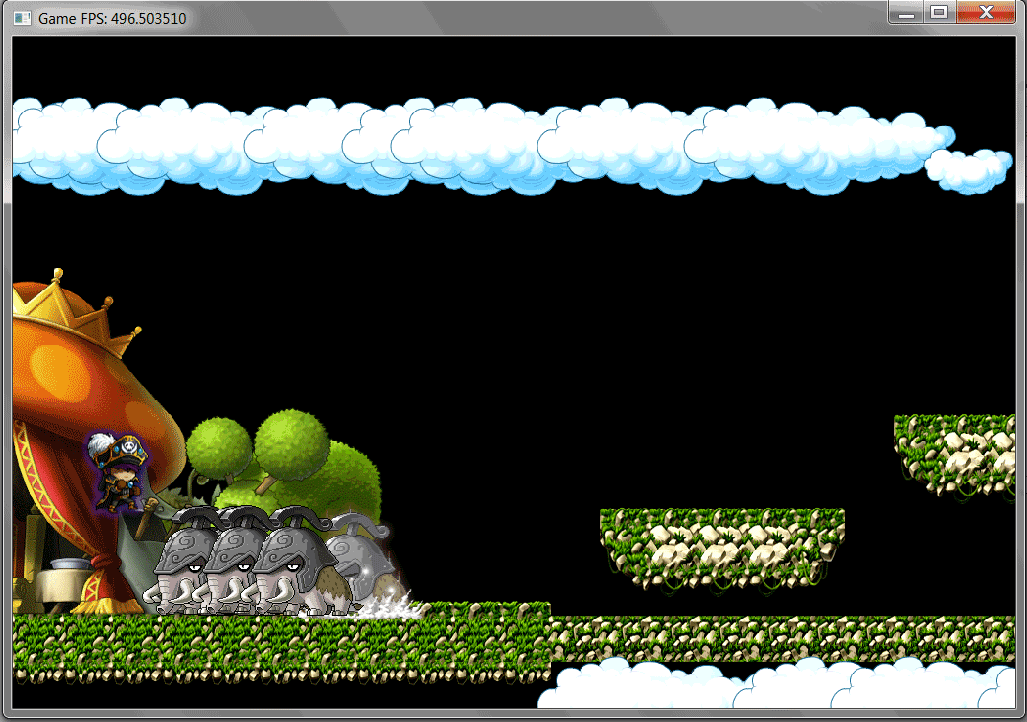
1. ***Chọn màn chơi***



Hình 5: Màn hình chọn màn chơi

* Màn hình chọn màn chơi: Chọn màn chơi trong số 9 màn.
* Có thể dùng chuột nháy vào tên mỗi địa danh hoặc dùng phím di chuyển để chọn,

1. ***Giao diện chơi chính***



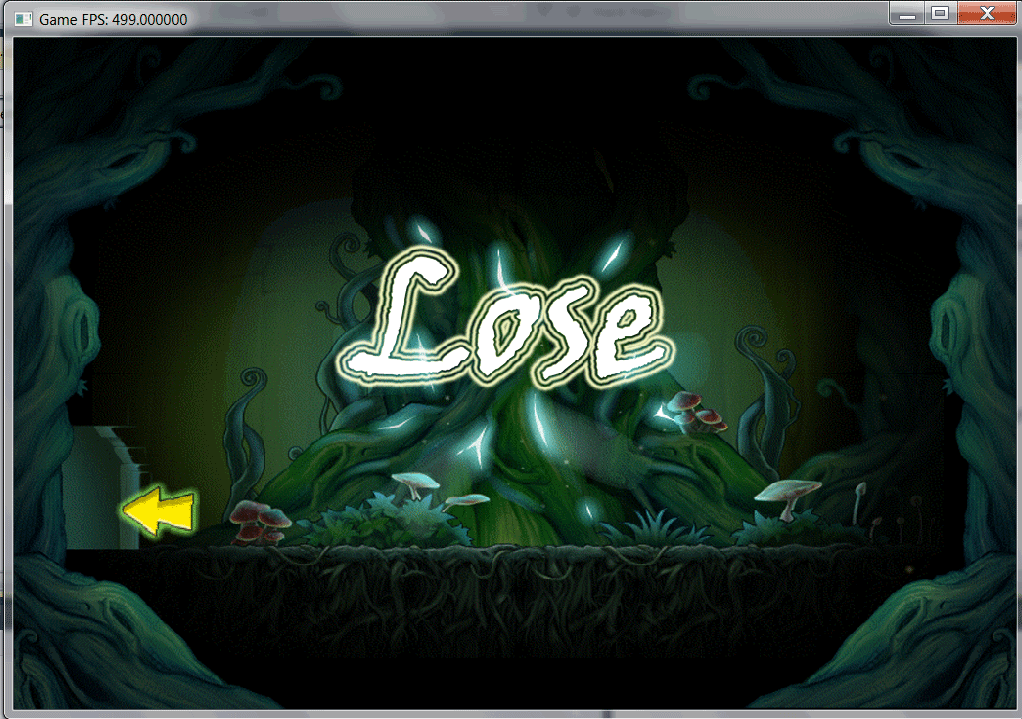
*Hình 6: Màn hình chơi chính*

* Màn hình chơi chính: Nơi người chơi tấn công quái vật để vượt qua màn chơi.
* Màn chơi kết thúc khi giết được tất cả các Boss.

1. ***Win và Lose***



*Hình 7: Màn hình Win*



*Hình 8: Màn hình Lose*

# Đánh giá hướng mở rộng :

## Đánh giá

### Ưu điểm

Cấu trúc quản lý game tốt sử dụng hướng đối tượng cao.

Hình ảnh hiệu ứng trong game rất đẹp.

Đồ họa được tối ưu tốt nhờ quản lý bằng singleton.

GamePlay đặc sắc nhờ hệ thống skill đa dạng.

Âm thanh hoàng tráng phù hợp với các skill

Mapeditor giao diện dễ dùng, xuất ra map chính xác không có lỗi.

### Nhược điểm

Đồ họa cao quá, ảnh hưởng trực tiếp đến các máy có cấu hình không cao.

Loading tương đối lâu do xử lý nhiều hình ảnh.

## Hướng mở rộng

Do cấu trúc game theo hướng đối tượng , quản lý resouce sprite, sound , device băng singleton, quản lý state rõ ràng nên dễ dàng phát triển game sau này theo các hướng sau :

* Hỗ trợ Multiplayer : 2 người chơi trong game cùng lúc.
* Thêm các tính năng : Shop bán đồ cho nhân vật, lên level cho nhân vật, tương tác giữa nhân vật và đời sống .