

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM

BÁO CÁO ĐỒ ÁN MÔN HỌC
XÂY DỰNG ỨNG DỤNG GAME MINH HỌA
XNA GAME STUDIO

GIẢNG VIÊN HƯỚNG DẪN
THS. HUỲNH HỒ THỊ MỘNG TRINH
THS. PHAN NGUYỆT MINH

SINH VIÊN THỰC HIỆN
HỒ THÁI NGỌC – 16520825

TP. HỒ CHÍ MINH, 2018

LỜI CẢM ƠN

Trước hết nhóm xin gửi lời cảm ơn chân thành đến thầy, ThS. Huỳnh Hồ Thị Mộng Trinh, ThS. Phan Nguyệt Minh, ThS. Phạm Thi Vương, khoa Công Nghệ Phần Mềm, Trường Đại học Công Nghệ Thông Tin. Trong suốt thời gian thực hiện đồ án, thầy đã dành rất nhiều thời gian và tâm huyết trong việc hướng dẫn nhóm. Thầy đã luôn có những định hướng, góp ý, sửa chữa những chỗ sai sót giúp nhóm có thể đi được đúng hướng. Đồ án có được kết quả ngày hôm nay là nhờ sự nhắc nhở và giúp đỡ nhiệt tình của quý thầy cô.

Nhóm cũng xin trân trọng cảm ơn tất cả thầy cô trong khoa Công Nghệ Phần Mềm, cũng như các thầy cô đã giảng dạy, giúp đỡ chúng em trong suốt thời gian học tập ở trường. Những kiến thức nền tảng và chuyên môn mà chúng em học được từ các thầy cô đã giúp chúng em có thể hoàn thành đồ án này.

Sinh viên thực hiện đồ án

LỜI MỞ ĐẦU

Thiết bị vũ trụ ([tiếng Anh](#): *spacecraft*) là tên gọi chung của các thiết bị với chức năng là thực hiện nhiều bài toán khác nhau về không gian vũ trụ, tiến hành nghiên cứu các công việc khác nhau trên bề mặt của những thiên thể khác nhau. Các thiết bị vũ trụ được đưa lên quỹ đạo nhờ các [tên lửa đẩy](#).

Thiết bị vũ trụ được dùng để vận chuyển người hay các trang bị, hàng hóa lên khoảng không ở bên ngoài tầng [khí quyển Trái Đất](#) được gọi là tàu vũ trụ. **Tàu vũ trụ**, còn có tên gọi là **phi thuyền không gian** ([tiếng Anh](#): *spaceship*), có hai loại cơ bản là tàu vũ trụ có người lái như tàu Phương Đông (Liên Xô), [tàu Soyuz](#) (Nga), hệ thống [tàu con thoi](#) (Mỹ), tàu Thần Châu (Trung Quốc); tàu vận tải (tàu vũ trụ không người lái) như [tàu vận tải Tiến bộ](#) (Nga), [tàu vận tải HTV](#) (Nhật), v.v.

Spaceship war là đề tài làm nên tên tuổi cho nhiều tựa game đình đám.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI	1
1.1 Tổng quan về phát triển game	1
1.2 Tổng quan về ứng dụng game thực hiện	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 Ngôn ngữ C#	4
2.1.1 Định nghĩa	4
2.1.2 Đặc điểm ngôn ngữ	4
2.2 XNA framework và XNA Game Studio	5
2.2.1 Giới thiệu	5
2.2.2 Kiến trúc XNA và các trạng thái xử lý trong game	7
2.2.3 XNA Audio system	10
2.2.4 XNA Content Pipeline	10
2.2.5 XNA Input Control	11
2.2.6 XNA Sprite image 2D	11
CHƯƠNG 3: ỨNG DỤNG XNA GAME STUDIO VÀO PHÁT TRIỂN GAME	13
3.1 Tổng quan	13
3.1.1 Khảo sát các sản phẩm XNA GameStudio	13
3.1.2 Bối cảnh nghiên cứu	13
3.1.3 Mục tiêu nghiên cứu	14
3.2 Thiết kế và lập trình game	14
3.2.1 Tổng quan	14
3.2.2 Game play và Mechanics	15
3.2.3 Thiết kế giao diện	24
3.2.4 Môi trường và công cụ phát triển	25
CHƯƠNG 4 TỔNG KẾT	26
4.1 Kết quả đạt được	26
4.2 Khó khăn	26
4.3 Hướng phát triển	27
TÀI LIỆU THAM KHẢO	28

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

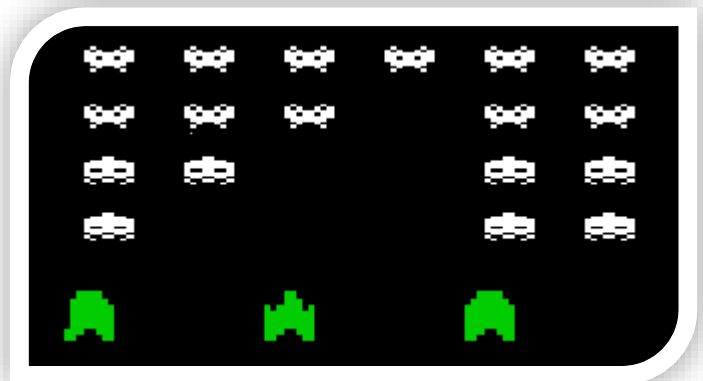
1.1 Tổng quan về phát triển game

Nửa thế kỉ qua, lịch sử làng game thế giới được tạo nên bởi những câu chuyện đáng kinh ngạc và những con người kiệt xuất đã làm thay đổi cuộc sống của chúng ta ngày nay. Bạn có biết rằng những cải tiến lớn trong thế giới ảo cũng đều tiêu tốn tài năng, sức lao động, mồ hôi và nước mắt của không biết bao nhiêu người đổ xuống. Đôi khi, đó còn là một cuộc chiến, một cuộc chiến vươn đến cái mới và sự tiến bộ của nhân loại

Sự bùng nổ của Space Invaders

Vào những năm 70 của thế kỉ trước, xu thế chính của thời đại là game arcade hơi hướng science-fiction với những cuộc xâm chiếm thế giới của người ngoài hành tinh. *Space Invader* ra đời như một điều tất yếu của xu thế này.

Space Invader là một game vô cùng quen thuộc đối với bất kì ai đã từng có thời gian chơi điện tử 4 nút bởi về mặt gameplay, bạn có thể hình dung nó giống hệt với game “bắn ruồi”. *Space Invader* là game bắn súng 2D mà người chơi được điều khiển một khẩu pháo lazer di chuyển ngang để tiêu diệt lũ alien xếp hàng trước mặt. Lũ Alien sẽ tấn công người chơi bằng việc lao thẳng vào khẩu pháo bạn đang điều khiển và nhiệm vụ của người chơi là tránh né và ghi thật nhiều điểm để qua bàn.



Game ban đầu được phát triển bởi Tomohiro Nishikado dựa trên các sản phẩm điện ảnh như Breakout, The War of the Worlds hay Star Wars và được phát hành tại Nhật Bản nhưng sau đó bị mua lại bởi một công ty của Mỹ. Ngay sau đó, dưới bàn tay kinh doanh của những “ông chủ thực sự”, game đã đạt được những thành công thương mại ngoài sức tưởng tượng và được ghi vào kỷ lục Guinness là top game Arcade với 500 triệu đô la tính đến năm 2007. *Space Invader* là sản phẩm game đi tiên phong cho xu thế game chinh phục toàn cầu.

1.2 Tổng quan về ứng dụng game thực hiện

-Tên Game: Vankyrie War

-Môi trường cài đặt và phát triển:

- + Sử dụng Microsoft Visual Studio

- + Lập trình ngôn ngữ C# (XNA)

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Ngôn ngữ C#

2.1.1 Định nghĩa

C# là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường.

Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

2.1.2 Đặc điểm ngôn ngữ

C#, theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến .NET Framework mà tất cả các chương trình .NET chạy, và nó phụ thuộc mạnh mẽ vào framework này.

Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception... phản ánh rõ ràng những đặc trưng của .NET runtime.

2.2 XNA framework và XNA Game Studio

2.2.1 Giới thiệu

Microsoft XNA (viết tắt của XNA's Not Acronymed) là bộ công cụ miễn phí chạy trong môi trường runtime được cung cấp bởi Microsoft giúp phát triển và quản lý video game, bao gồm **XNA Framework**, **XNA Build** và **XNA Game Studio**. XNA dựa trên [.NET Framework](#) với các phiên bản chạy trên Windows NT, Windows Phone và Xbox 360. Các nội dung XNA được tạo dựng bằng XNA Game Studio, chạy bằng XNA Framework (cho các game trên Windows) và phát hành dưới dạng file thực thi nguyên gốc (native executable) cho Xbox 360, Windows Phone và Zune.

Về nhiều mặt thì XNA có thể được xem như bản tương đồng với hệ thống phát triển game nổi tiếng của Microsoft là DirectX nhưng hướng tới những nhà phát triển chủ yếu lập trình game nhẹ, chạy trên nền tảng Windows. XNA cũng là nền tảng cơ bản của các game Xbox Live Indie Games.



XNA Game Studio là một loạt các thư viện được Microsoft xây dựng nhằm hỗ trợ lập trình game, XNA khá mạnh mẽ - được tích hợp với Visual Studio và viết bằng ngôn ngữ C#.

XNA Game Studio là [môi trường lập trình](#) để phát triển trò chơi. Năm bản sửa đổi đã được phát hành cho đến nay, nhưng đến năm 2015, sẽ không có phiên bản mới nào được phát triển.

XNA Game Studio Express là phiên bản đầu tiên được phát hành vào ngày 30 tháng 8 năm 2006, và dành cho sinh viên, người có sở thích và các nhà phát triển trò chơi độc lập (và [homebrew](#)). Nó có sẵn dưới dạng bản tải xuống miễn phí. Express cung cấp "bộ khởi động cơ bản" cơ bản để phát triển nhanh các thể loại trò chơi cụ thể, chẳng hạn như [trò chơi nền tảng](#), [chiến lược thời gian thực](#) và [game bắn súng góc nhìn người đầu tiên](#). Nhà phát triển có thể tạo [Windows](#) Trò chơi miễn phí với XNA Framework, nhưng để chạy các trò chơi của họ trên Xbox 360, họ sẽ phải trả một khoản phí hàng năm là 99 đô la Mỹ (hoặc một khoản phí bốn tháng là 49 đô la Mỹ) để được nhận vào Câu lạc bộ của Microsoft XNA Creator. Với bản cập nhật, các nhà phát triển XNA có thể biên dịch các tệp nhị phân của Xbox 360 và chia sẻ chúng với các thành viên Câu lạc bộ của Microsoft XNA Creator.

XNA Game Studio 2.0 được phát hành vào ngày 13 tháng 12 năm 2007. XNA Game Studio 2.0 có khả năng linh được sử dụng với tất cả các phiên bản của [Visual Studio 2005](#) (bao gồm cả miễn phí [Visual C # 2005 Express Edition](#)), một API mạng sử dụng [Xbox Live](#) trên cả Windows và Xbox 360 và xử lý thiết bị tốt hơn.

XNA Game Studio 3.0 (cho Visual Studio 2008 hoặc Visual C # 2008 Express Edition miễn phí) cho phép sản xuất trò chơi nhắm mục tiêu nền tảng [Zune](#) và thêm hỗ trợ cộng đồng [Xbox Live](#). Nó được phát hành vào ngày 30 tháng 10 năm 2008, và hỗ trợ C # 3.0, LINQ và hầu hết các phiên bản của Visual Studio 2008.

XNA Game Studio 4.0 được phát hành vào ngày 16 tháng 9 năm 2010. Nó hỗ trợ nền tảng [Windows Phone](#) (bao gồm tăng tốc phần cứng 3D), cấu hình phần cứng khung, hiệu ứng cấu hình, đối tượng trạng thái tích hợp, scalars thiết bị đồ họa

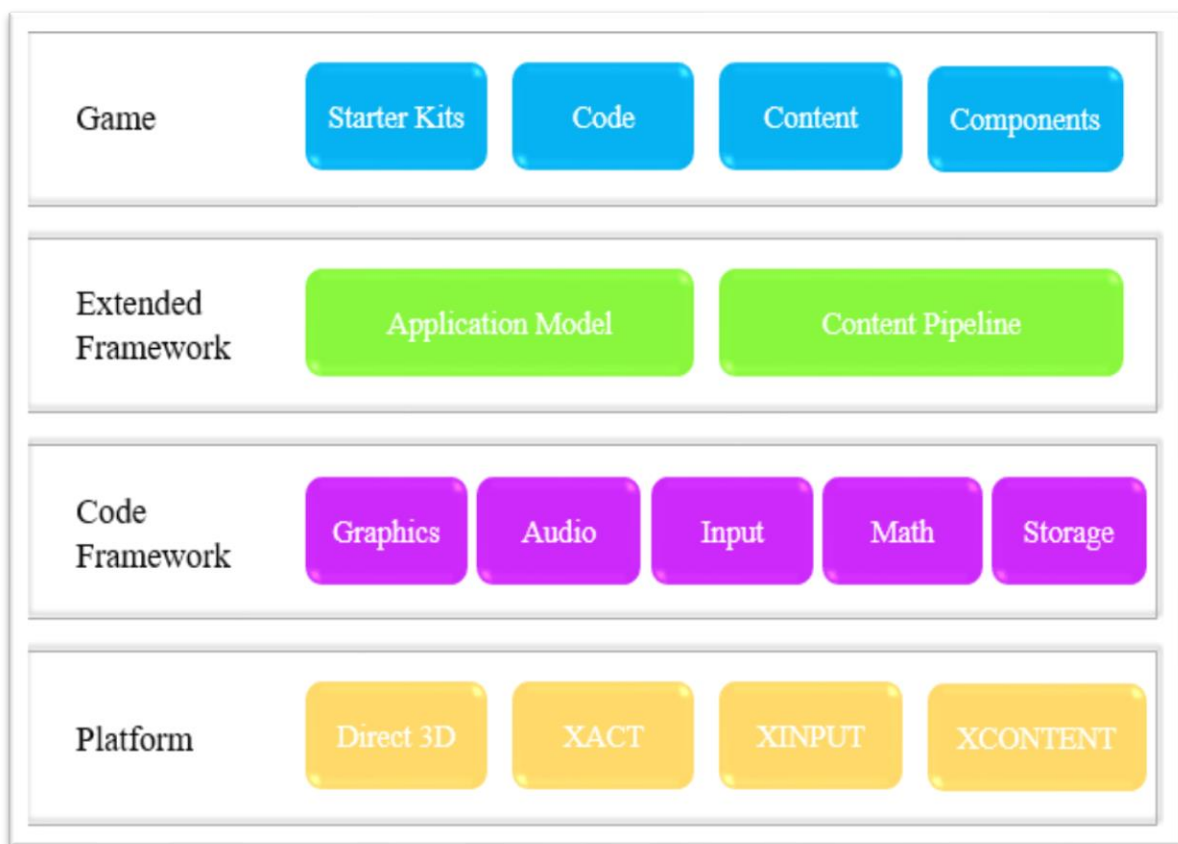
và hướng, chéo đầu vào - nền tảng và đầu vào đa chạm , đầu vào micrô và phát lại âm thanh đệm và tích hợp Visual Studio 2010 .

XNA "Game Studio 4.0 Refresh" được phát hành vào ngày 6 tháng 10 năm 2011 và hỗ trợ thêm cho Windows Phone 7.5 (Mango) và Visual Basic .

2.2.2 Kiến trúc XNA và các trạng thái xử lý trong game

2.2.2.1 Kiến trúc XNA

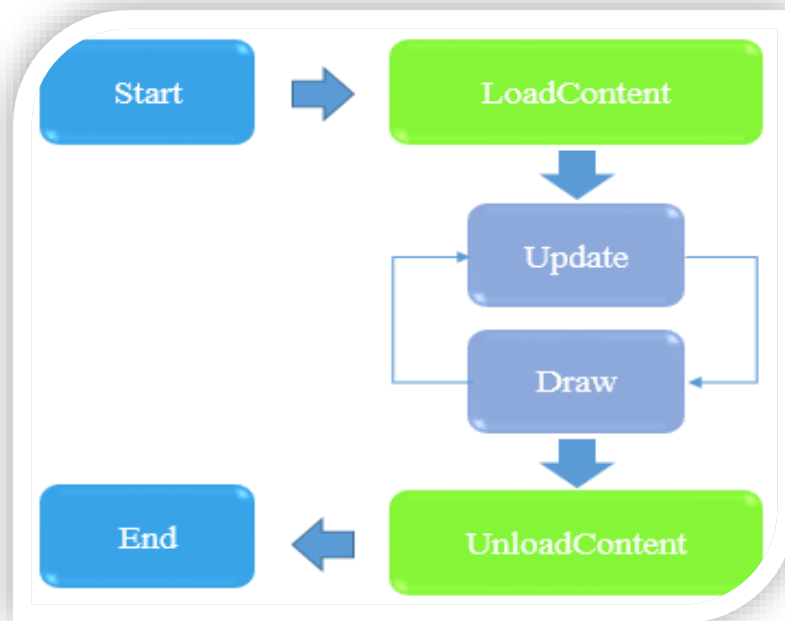
Kiến trúc XNA bao gồm các thành phần xử lý đồ họa, âm thanh, toán học, quá trình nhập xuất, mạng.



2.2.2.2 Các trạng thái xử lý trong Game

- Để xây dựng một game XNA, bước đầu tiên là phải tạo ra một lớp kế thừa từ `Microsoft.Xna.Framework.Game`, lớp này cần phải khai báo override các phương thức **Update**, **Draw**, **LoadContent** và **Initialize**.
- Phương thức **Update** có trách nhiệm xử lý logic trong game và phương thức **Draw** có trách nhiệm vẽ các thành phần của game ra màn hình.
- Phương thức **Initialize** cùng với hàm constructor có trách nhiệm khởi tạo các giá trị ban đầu cho các thành phần đồ họa, âm thanh và input trong game.
- Phương thức **LoadContent** sẽ load tài nguyên đồ họa bao gồm các hình ảnh sẽ vẽ trong game và **UnloadContent** có trách nhiệm giải phóng tài nguyên đồ họa.

```
public class GameEx : Microsoft.Xna.Framework.Game
{
    public GameEX()
    protected override void Initialize()
    protected override void LoadContent()
    protected override void UnloadContent()
    protected override void Update(GameTime gameTime)
    protected override void Draw(GameTime gameTime)
}
```



Quá trình thực thi có thể ánh xạ qua Solution của XNA game

- **GameEX()** : Khởi tạo các thành phần chung
- **Initialize()**: khởi tạo game
- **LoadContent()**: Load các tài nguyên đồ họa hình ảnh vào game
- **Run()**: Chạy vòng lặp game
 - + **Update()**: Đọc Input từ người dùng, tính toán logic game, kiểm tra điều kiện dừng của game
 - + **Draw()**: Vẽ lên màn hình game dựa trên cơ sở các tài nguyên của LoadContent đã thực thi xử lý.
- **UnloadContent()**: Giải phóng tài nguyên đồ họa.

2.2.3 XNA Audio system

- XNA Framework 2.0, các file âm thanh được add vào Content dưới dạng đặc biệt, nó được tạo ra từ tool Microsoft Cross-PlatformAudio Creation (XACT).

- **XACT** tạo ra các Sound bank, Wave bank từ file .wav; .AIFF; .XMA...

Biên dịch chúng thành file XAP.

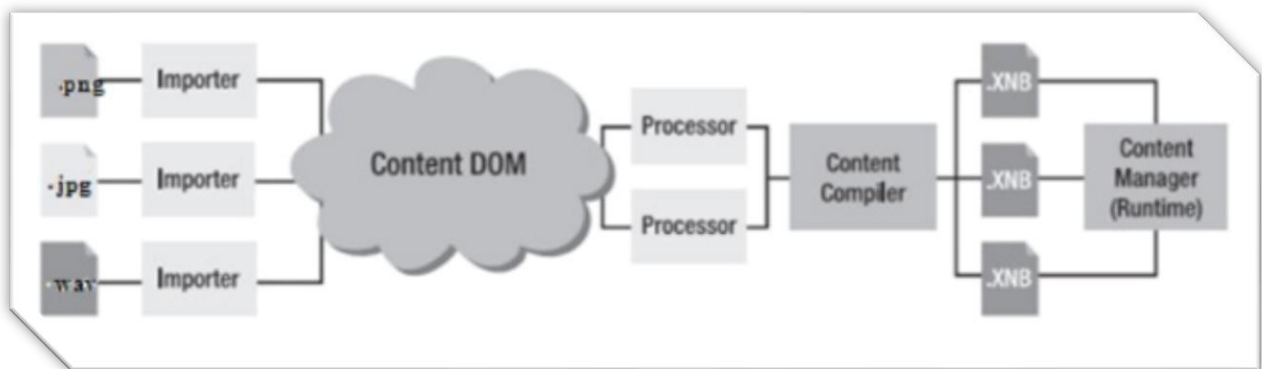
- **File XAP** được load từ Content và thực thi thông qua các đối tượng Sound bank, Wave bank, AudioEngine.

- XNA Framework 3.0 trở lên XACT được thay thế bởi SoundEffect, SoundEffectInstance đơn giản và dễ sử dụng cho các nhà phát triển game.

2.2.4 XNA Content Pipeline

- XNA Game Studio Content Pipeline giúp các nhà phát triển game đơn giản hóa trong quá trình load các loại tài nguyên khác nhau: Audio, Photos,...mà không cần quan tâm cách thức load của nó.

- Các file tài nguyên sẽ có định dạng **.xnb**, nó sẽ được load và thực thi trong game nhờ vào **Content Manager**



2.2.5 XNA Input Control

- Trong Microsoft XNA Framework, namespace `Microsoft.Xna.Framework.Input` hỗ trợ nhiều thiết bị input khác nhau.
- Các thiết bị input bao gồm:
 - + Keyboard
 - + Mouse
 - + Gamestick
 - + Xbox 360

2.2.6 XNA Sprite image 2D

Khái niệm:

Sprite image 2D là 1 image tập hợp tất cả các tư thế cơ bản của nhân vật theo một qui luật nào đó.



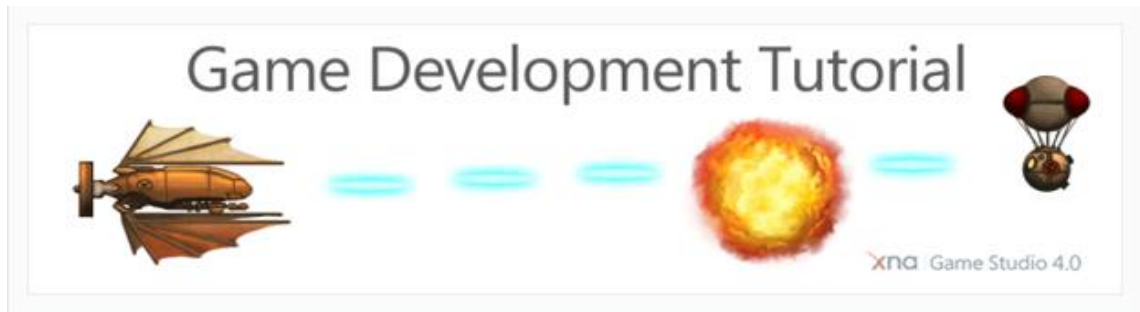
Tạo chuyển động từ Sprite image 2D

- Nhân vật hiển thị trong Game được cắt từ 1 trong các tư thế ở trong Sprite image ở trên. Tạm gọi là khung cắt
- Do các tư thế xếp cách đều nhau nên việc xác định chính xác khung cắt hết sức dễ dàng. Cả sprite image được xem như 1 ma trận 2 chiều các tư thế
- Khi cần nhân vật chuyển động. giả sử cần nhân vật chuyển động từ dưới lên trên. Ta sẽ cho chỉ số dòng của khung cắt bằng 0 đồng thời cho chỉ số cột lặp lại từ 0->4 và chỉ stop khi nhận được lệnh stop từ chương trình.
- Như vậy tập hợp các tư thế nhân vật chuyển động từ dưới lên được load trong thời gian ngắn làm cho chúng ta có cảm giác nhân vật bước đi từ dưới lên như thật vậy.
- Với các hướng chuyển động khác thì cũng làm tương tự nhưng phải thay đổi chỉ số dòng của khung cần cắt cho đúng với hướng chuyển động mà ta mong muốn.

CHƯƠNG 3: ỨNG DỤNG XNA GAME STUDIO VÀO PHÁT TRIỂN GAME

3.1 Tổng quan

3.1.1 Khảo sát các sản phẩm XNA GameStudio



3.1.2 Bối cảnh nghiên cứu

Hiện nay trên đà phát triển bùng nổ của công nghệ, các thiết bị số như Ipad, smartphone, PC, Laptop...ngày càng được người dùng quan tâm, đặt biệt là trong lĩnh vực giải trí.

3.1.3 Mục tiêu nghiên cứu

- Trò chơi được thực hiện với XNA có thể chạy trên Windows, Xbox 360 và Windows Phone 7, với các dự án khác như **MonoXNA** và **MonoGame** cố gắng đưa trò chơi đến các nền tảng khác.
- Nâng cao kỹ năng lập trình, phát triển ứng dụng đa nền tảng.
- Tiếp cận công mới, ứng dụng thực tiễn, tạo nền tảng tư duy cho các công trình nghiên cứu.
- Đáp ứng yêu cầu môn học, tạo ra game có tính giải trí phù hợp.

3.2 Thiết kế và lập trình game

3.2.1 Tổng quan

Ý tưởng:

Game lấy đề tài chiến tranh không gian, trong vai một người phi công, bạn phải điều khiển chiếc phi thuyền chiến đấu để bắn hạ từng đợt xâm lăng của hạm đội địch. Mặc dù chúng dùng một tuyệt chiêu duy nhất, nhưng chúng lại cực kì thông minh và bài bản. Từng lớp, từng hạm đội được sắp xếp theo những đội hình khác nhau để có thể tấn công Sao Hỏa. Bạn phải nhìn kĩ để có thể chọn cho mình một chỗ đứng an toàn và sau đó, cuộc chiến sẽ bắt đầu.

Thể loại game:

Game thuộc thể loại hành động, nhịp độ trận chiến bùng nổ, thách thức khả năng phản xạ nhanh, kỹ thuật điều luyện của người chơi.

Đối tượng hướng đến:

Game hành động nhập vai chiến thuật, không có cảnh bạo lực mạnh, hình ảnh vui nhộn, nên có thể hướng đến mọi đối tượng.

Quá trình hoạt động của game:



Cảm nhận về game:

Nhìn nhận ban đầu, game sẽ mang tới thách thức cho người chơi qua các màn đấu súng với mức độ khó dễ khác nhau, âm thanh sống động và hiệu ứng mượt mà là những gì mà game sẽ mang lại cho người chơi.

3.2.2 Game play và Mechanics

3.2.2.1 Game play

Tiến trình game:

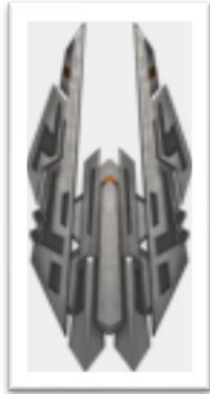
Nhiệm vụ duy nhất của người chơi trong game là chiến đấu đến cùng trước hạm đội xâm lược và bảo vệ Sao Hỏa.

Độ thách thức trong game:

Người chơi phải nhanh nhẹn né tránh và xử lý tình huống bất ngờ để tiêu diệt phi thuyền địch, giành chiến thắng cuối cùng.

Các đối tượng

- Spaceship Player



+ Phi thuyền chính mà người chơi điều khiển trong game, nó có thể di chuyển ngang dọc trong màn hình.

+ Phi thuyền được trang bị pháo laser tối tân.

- Spaceship Enemy



+ Phi thuyền hạm đội địch, bay theo đội hình không cố định.

+ Phi thuyền trang bị hệ thống tàng hình, xuất hiện bất ngờ tấn công người chơi.

+ Trang bị tia laze siêu nhanh.

- Asteroid

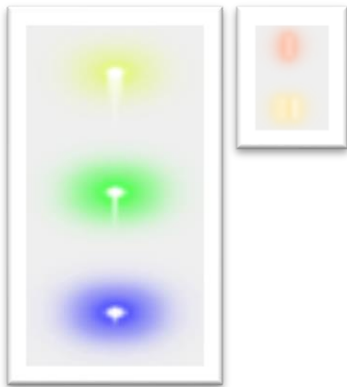


+ Thiên thạch không gian, chúng bay theo hướng không xác định, bay rất nhanh trong không gian.

+ Thiên thạch có thể va chạm với phi thuyền người chơi và phát nổ.

+ Là kẻ thù gián tiếp.

- Bullet, Space



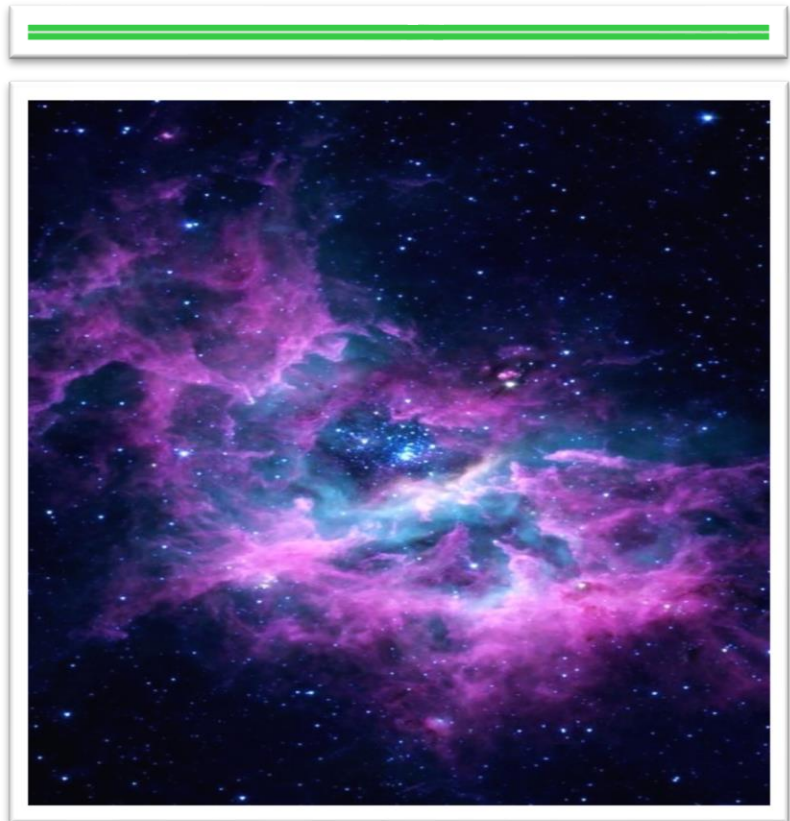
Pháo laser kết cấu ba tầng công phá.

Tia laze siêu nhanh xuyên thấu vỏ địa chất.

Không gian game mờ ảo, kích thích thị giác người chơi.

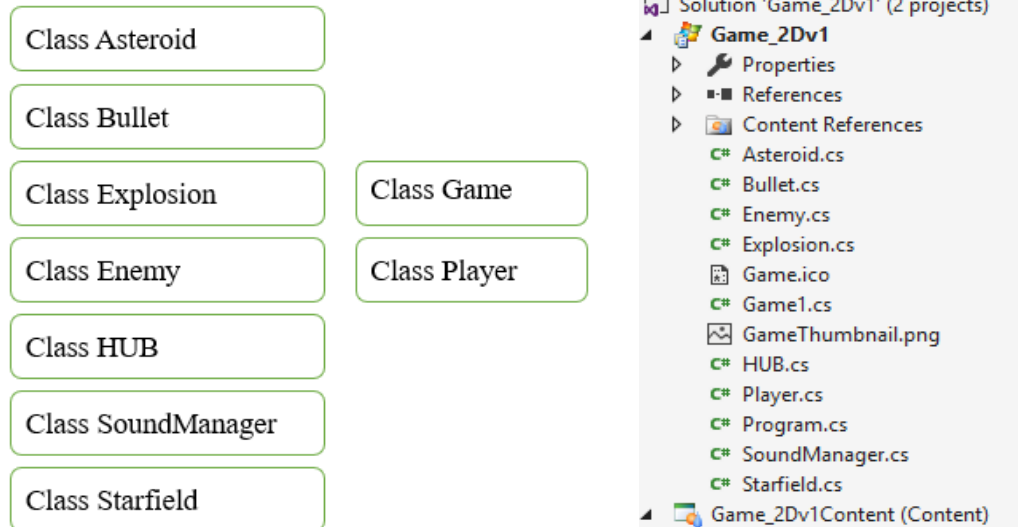
- Health player

Thanh năng lượng, nếu thanh năng lượng cạn mất thì GameOver.



3.2.2.2 Mechanics – Cơ chế game

Kiến trúc game



Hiệu ứng

Các hiệu ứng được sử dụng trong game, mang đến trải nghiệm khác lạ cho người chơi, các đối tượng không cứng nhắc tại một chỗ, mà di động lên xuống qua lại kết hợp, hòa quyện vào nhau tạo nên một thế giới game sinh động.

- Spaceship player

+ Di chuyển spaceship player

Điều khiển spaceship player bằng các phím (W, S, A, D), khi các phím này sử dụng, Rectangle của spaceship player sẽ được update vị trí trên màn hình game.

```
//getting keyboard state
KeyboardState keyState = Keyboard.GetState();

//boundingBox for playerShip
boundingBox = new Rectangle((int)position.X, (int)position.Y,
    texturn.Width, texturn.Height);
```

Thiết lập giới hạn di chuyển cho spaceship player không được di chuyển ra khỏi màn hình game play.

Chiều rộng của màn hình game 900, chiều cao 740.

```
public Game1()
{
    //constructor
    graphics = new GraphicsDeviceManager(this);
    graphics.IsFullScreen = false;
    graphics.PreferredBackBufferHeight = 740;
    graphics.PreferredBackBufferWidth = 900;
    this.Window.Title = "XNA Gamer";
    Content.RootDirectory = "Content";
    enemybulletDamage = 10;
    menuImage = null;
    gameoverImage = null;
}

//keep ship_1 in screen
if (position.X <= 0)
    position.X = 0;

if (position.X >= 900 - texture.Width)
    position.X = 900 - texture.Width;

if (position.Y <= 0)
    position.Y = 0;

if (position.Y >= 720 - texture.Height)
    position.Y = 720 - texture.Height;
```

+ Pháo laser của spaceship player

Xây dựng class Bullet, class Player riêng biệt. class Player sẽ kế thừa từ class Bullet, xây dựng một mảng đối tượng Bullet.

Khởi tạo ban đầu cho số lượng pháo laser thông qua bulletDelay.

```
//Shooting ship
public void Shoot()
{
    if (bulletDelay >= 0)
        bulletDelay--;

    if (bulletDelay <= 0)
    {
        sn.playerShootSound.Play();
        Bullet newBullet = new Bullet(bulletTexture);
        newBullet.position = new Vector2(position.X + 32 -
        newBullet.texture.Width /
        2, position.Y + 30);
        newBullet.isVisible = true;
    }
}
```

```
//bullet visible
if (bulletList.Count < 20)
    bulletList.Add(newBullet);
}

//reset bullet
if (bulletDelay == 0)
    bulletDelay = 10;
}
```

Xử lý trạng thái tang biến của pháo laser khi chạm mục tiêu hoặc chạm giới hạn trên của màn hình game.

```
foreach (Bullet b in bulletList)
{
    //BoundingBox for every bullet in listbullet
    b.boundingBox = new Rectangle((int)b.position.X, (int)b.position.Y,
b.texture.Width,
b.texture.Height);
    //xet vi tri bullet
    b.position.Y -= b.speed;
    //xoa bullet khi cham top
    if (b.position.Y <= 0)
        b.isVisible = false;
}
```

Xây dựng class `Explosion` , quản lý vụ nổ khi va chạm giữa các đối tượng spaceship player, spaceship enemy, Asteroid...



```
//check collision ennemy vs ship's player
if (e.boundingBox.Intersects(p.boundingBox))
{
    p.health -= 40;
    e.isVisible = false;
}

//check collision bullet's enemy vs ship's player
for (int i = 0; i < e.bulletList.Count; i++)
{
    if (p.boundingBox.Intersects(e.bulletList[i].boundingBox))
    {
        p.health -= enemybulletDamage;
        e.bulletList[i].isVisible = false;
    }
}
```

```
//check collision bullet's ship's player vs ennemy
```



```

for (int i = 0; i < p.bulletList.Count; i++)
{
    if (p.bulletList[i].boundingBox.Intersects(e.boundingBox))
    {
        sn.explodeSound.Play();
        listExplosion.Add(new
        Explosion(Content.Load<Texture2D>("Explosion"), new
        Vector2(e.position.X, e.position.Y)));
        hub.playerScore += 20;
        p.bulletList[i].isVisible = false;
        e.isVisible = false;
    }
}

```

Các đối tượng sau khi va chạm với spaceship player sẽ biến mất, đồng thời thanh năng lượng của spaceship player giảm 20 đơn vị.

```

//khi ship collision vs Asteroid => asteroid.isVisible=false no biến mất
if (a.boundingBox.Intersects(p.boundingBox))
{
    a.isVisible = false;
    p.health -= 20;
}
a.Update(gameTime);

```

- Starfield (Không gian)

Ứng dụng SpriteBatch chạy ảnh nền không gian theo chiều dọc tạo hiệu ứng đang bay của spaceship player theo trục tọa độ Oy.

Giới hạn update là -720

```

//Draw
public void Draw(SpriteBatch spriteBatch)
{
    spriteBatch.Draw(texturn, bgPos1, Color.White);
    spriteBatch.Draw(texturn, bgPos2, Color.White);
}

//Update
public void Update(GameTime gameTime)
{
    //setting speed for background scrolling
    bgPos1.Y += speed;
    bgPos2.Y += speed;
    //scrolling background
    if(bgPos1.Y>=720)
    {
        bgPos1.Y = 0;
        bgPos2.Y = -720;
    }
}

```

- Asteroid (Thiên thạch)

Random vị trí của thiên thạch rơi trên màn hình game play, tạo yếu tố bất ngờ cho người chơi.

```
randX = random.Next(0, 1000);  
randY = random.Next(-700, -60);
```

Tạo hiệu ứng thiên thạch xoay đánh lừa thị giác người chơi.

```
//Rotate Asteroid  
float elapsed = (float)gameTime.ElapsedGameTime.TotalSeconds;  
rotationAngel += elapsed;  
float circle = MathHelper.Pi * 2;  
rotationAngel = rotationAngel % circle;
```

-Spaceship Enemy, Asteroid, Explosion, HUB (điểm số) được quản lý thông qua List, được vẽ dựa trên spriteBatch

```
//Lists  
List<Asteroid> listAsteroid = new List<Asteroid>();  
List<Enemy> listEnemy = new List<Enemy>();  
List<Explosion> listExplosion = new List<Explosion>();  
  
//drawing playing state  
sf.Draw(spriteBatch);  
p.Draw(spriteBatch);  
foreach (Explosion ex in listExplosion)  
{  
    ex.Draw(spriteBatch);  
}  
  
foreach (Asteroid a in listAsteroid)  
{  
    a.Draw(spriteBatch);  
}  
  
foreach (Enemy e in listEnemy)  
{  
    e.Draw(spriteBatch);  
}  
  
hub.Draw(spriteBatch);
```

Cơ chế tính điểm

- Spaceship player bắn hạ hoặc va chạm với Spaceship enemy : Cộng 20 điểm.

```
//check collision bullet's ship's player vs ennemy
hub.playerScore += 20;
```

- Spaceship player bắn hạ hoặc va chạm với Asteroid (thiên thạch): Cộng 5 điểm.

```
//khi ship collision vs Asteroid => asteroid.isVisible=false no bien mat
hub.playerScore += 5;
```

- Khi spaceship player hết năng lượng (máu) điểm số được tính, và đặc lại bằng 0.

```
//reset health's player, score's player
p.health = 800;
hub.playerScore = 0;
gameState = State.Menu;
```

Cơ chế năng lượng (máu)

- Khởi tạo năng lượng cho spaceship player : 800
- Spaceship player bị bắn hạ hoặc va chạm với Spaceship enemy : Trừ 40 năng lượng.

```
//check collision ennemy vs ship's player
if (e.boundingBox.Intersects(p.boundingBox))
{
    p.health -= 40;
    e.isVisible = false;
}
```

- Spaceship player va chạm với Asteroid (thiên thạch): Trừ 20 năng lượng.

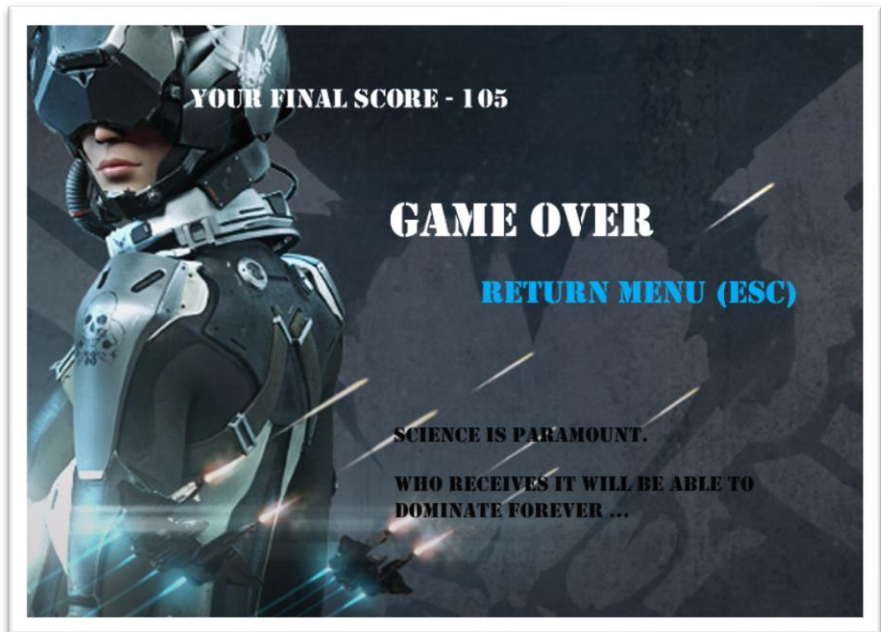
```
//khi ship collision vs Asteroid => asteroid.isVisible=false no bien mat
if (a.boundingBox.Intersects(p.boundingBox))
{
    a.isVisible = false;
    p.health -= 20;
}
```

- Khi spaceship player hết năng lượng (máu) tiến đến trạng thái GameOver.

3.2.3 Thiết kế giao diện

Hệ thống giao diện trực quan





Âm thanh, nhạc nền

Âm thanh trong game chia làm 2 phần:

- Nhạc nền
- Hiệu ứng game play: Bắn pháo laser, tia lazer, va chạm thiên thạch, các vụ nổ hạt nhân.

3.2.4 Môi trường và công cụ phát triển

- Môi trường phát triển:

Visual Studio 2017

XNA Game Studio 4.0

-Thư viện hỗ trợ:

XNA Framework 4.0

CHƯƠNG 4 TỔNG KẾT

4.1 Kết quả đạt được

Về mặt lý thuyết:

- Thông qua đồ án game, giúp sinh viên thực hiện củng cố kiến thức trong lĩnh vực lập trình game căn bản.
- Bên cạnh đó, sinh viên có cơ hội tiếp cận những công nghệ mới về lập trình game đa nền tảng.
- Xây dựng kiến thức nền tảng, tiếp cận công nghệ mới.

Về mặt thực tiễn:

Hoàn thành đồ án môn học.

Đáp ứng yêu cầu của môn học:

- Game chạy theo thời gian thực (real time)
- Sử dụng sprite cho tạo hình nhân vật
- Xây dựng cấu trúc dữ liệu để quản lí nhân vật, tài nguyên, điểm số.. trong game.
- Hàm main chia thành 3 phần chính: Load ảnh, Khởi tạo nhân vật – Cập nhật game (phần real time) bao gồm cập nhật trạng thái nhân vật, vẽ nhân vật ra màn hình (dựa trên cấu trúc dữ liệu quản lí nhân vật), xét va chạm giữa các nhân vật (nếu có) để tính điểm – Kết thúc game, dọn rác.

4.2 Khó khăn

Công nghệ:

Mới tiếp cận lần đầu, khó nắm bắt hết được kỹ thuật xử lý. Tài liệu có nhưng không chi tiết cụ thể. Tốn nhiều thời gian nghiên cứu.

Thiết kế và lập trình game:

- Do chưa có nhiều kinh nghiệm phát triển game nên một số chức năng chưa hoàn chỉnh.
- Chưa tạo được hiệu ứng cầu kỳ cho các đối tượng trong game.

4.3 Hướng phát triển

- Tiếp tục hoàn thiện đồ án game về chế độ game play, hiệu ứng nhân vật thêm sống động.
- Tạo ra nhiều chế độ chơi khác nhau.
- Phát triển đồ án game thành game thương mại.

TÀI LIỆU THAM KHẢO

<https://stackoverflow.com>

<https://www.codeproject.com>

<https://github.com>

[https://docs.microsoft.com/en-us/previous-versions/windows/xna/bb200104\(v=xnagamestudio.41\)](https://docs.microsoft.com/en-us/previous-versions/windows/xna/bb200104(v=xnagamestudio.41))