**Mục lục**

[**LỜI MỞ ĐẦU 3**](#_Toc504641056)

[**CƠ SỞ LÝ THUYẾT 4**](#_Toc504641057)

[**PHÂN TÍCH & THIẾT KẾ TRÒ CHƠI 6**](#_Toc504641058)

[1. Lập bảng ghi thông tin các thành viên trong nhóm. 6](#_Toc504641059)

[2. Lập bảng ghi thông tin mỗi thành viên chịu trách nhiệm hiện thực hay kiểm tra chức năng. 6](#_Toc504641060)

[3. Các hàm xử lý chính trong file thực hiện chức năng X. 7](#_Toc504641061)

[**3.1 Chức năng 1: Xây dựng khung chương trình và các quang cảnh cần thiết trong trò chơi: 7**](#_Toc504641062)

[**3.2 Chức năng 2: Xây dựng các đối tượng cần thiết trong trò chơi 8**](#_Toc504641063)

[4. Các ưu điểm và hạn chế khi làm bài tập nhóm 10](#_Toc504641064)

[5. Các kinh nghiệm học được từ bài tập nhóm 10](#_Toc504641065)

[6. Hướng dẫn sử dụng game 11](#_Toc504641066)

[**TƯ LIỆU THAM KHẢO 12**](#_Toc504641067)

**NHẬN XÉT CỦA GIÁO VIÊN**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………… ………………………………………………………………………………………… …………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………… ………………………………………………………………………………………… …………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………… ………………………………………………………………………………………… …………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

# LỜI MỞ ĐẦU

Ngày nay, công nghệ thông tin được xem là ngành mũi nhọn, là động lực quan trọng thúc đẩy sự phát triển của các lĩnh vực khác của đời sống kinh thế , văn hóa, xã hội… Đặc biệt trong thời gian gần đây, chúng ta có thể nhận thấy sự phát triển từng ngày của công nghệ thông tin đang ngày càng trở nên thuận tiện hơn trong cuộc sống: làm việc, mua sắm, cập nhập tin tức hằng ngày.... Và đăc biệt không thể thiếu đó là nhu cầu giải trí ngày càng cao của con người: chơi game, xem phim, đọc truyện...Con người có thể thực hiện mọi thứ mà không cần thiết phải bước ra khỏi nhà bằng cách sử dụng mạng internet. Có rất nhiều loại mô hình đã được triển khai rộng rãi trên các nước đã phát triển, tuy nhiên ở nước ta do mặt bằng hạ tầng viễn thông chưa phát triển lắm nên việc triển khai còn gặp nhiều khó khăn, đặc biệt đó là ngành lập trình game.

Nhu cầu giải trí game đang là sự quan tâm hàng đầu của các bạn trẻ. Những người có khả năng nhanh nhẹn, sáng tạo tốt. Việc tạo ra nhiều trò chơi hay, bổ ích là điều rất cần thiết. Nổi trong số đó là là các game trí tuệ nhanh tay nhanh mắt. Nó tạo sự nhanh nhẹn, khéo léo, không bạo lực, tinh thần vui vẻ. Nhưng để có một game hay thì cần phải có am hiểu, biết nhu cầu, ý muốn của người chơi, phải tạo được một game đẹp, thân thiện, thu hút được người chơi. Để làm được điều đó chúng ta phải khảo sát va phân tích thiết kế game thật chi tiết trước khi tạo ra nó. Hôm nay nhóm mình sẽ cùng các bạn thiết kế lập trình trò chơi và mô phỏng game Bắn tàu vũ trụ và chúng ta sẽ thấy được điều đó.

# CƠ SỞ LÝ THUYẾT

**Game là gì?** Chắc các bạn đã từng chơi game rồi nhỉ ?

Về cơ bản mà nói thì Game(trò chơi) nó cũng là một ứng dụng(ứng dụng của nó là giải trí chẳng hạn).

Game có khá nhiều thể loại, ví dụ như: game trí tuệ, game hành động, đối kháng, đua xe, bla bla...

Điểm chung của tất cả các game đó là nó chứa một *vòng lặp "vô tận"* thực ra thì lúc nào game over thì nó cũng out vòng lặp và kết thúc ứng dụng luôn, nhưng vô tận ở đây có nghĩa là khi game còn chạy thì nó còn lặp). Vòng lặp chính là vòng lặp game (hay main loop). Các bạn có thể xem hình ảnh dưới đây để hiểu được cơ bản cách thức mà một game vận hành:

* Lúc đầu tiên game của chúng ta sẽ được khởi động lên
* Tiếp theo sẽ là vòng lặp của game, trong vòng lặp này sẽ là các tác vụ xử lý như: update lại trạng thái game (ví dụ chuyển động của nhân vật, điểm của người chơi, vv...), sau khi update thì nó sẽ tiến hành vẽ lại.
* Cuối cùng là kết thúc game(có thể là game over hoặc người chơi thoát chẳng hạn)  **Làm thế nào để tạo được một game?**

Đây cũng là câu hỏi mà lúc trước mình thắc mắc, vậy để làm được một game cơ bản ta cần những gì?

Dưới đây theo mình là những kiến thức cần thiết:

* Sự sáng tạo: để làm được một game thì điều đầu tiên chúng ta cần một ý tưởng (có thể là ý tưởng hay hoặc chưa tốt).
* Kiến thức lập trình cơ bản: Bây giờ ngôn ngữ lập trình rất chi là phong phú, bạn nên lựa chọn cho mình một nguôn ngữ lập trình phù hợp với bản thân. Một số ngôn ngữ lập trình game phổ biến hiện nay như: C++, Java, C#,... rất được ưa chuộng

**Java** là một [ngôn ngữ lập trình](http://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](http://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng)(OOP) và dựa trên các lớp

(class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì [biên dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_bi%C3%AAn_d%E1%BB%8Bch) [mã nguồn](http://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) thành [mã máy](http://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) hoặc [thông dịch](http://vi.wikipedia.org/wiki/Tr%C3%ACnh_th%C3%B4ng_d%E1%BB%8Bch) mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành [bytecode,](http://vi.wikipedia.org/w/index.php?title=Bytecode&action=edit&redlink=1) bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và [C++,](http://vi.wikipedia.org/wiki/C%2B%2B) nhưng sau này nhờ công nghệ “biên dịch tại chỗ” – Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như [Python,](http://vi.wikipedia.org/wiki/Python) [Perl,](http://vi.wikipedia.org/wiki/Perl) [PHP](http://vi.wikipedia.org/wiki/PHP) gấp nhiều lần. Java chạy tương đương so với [C#,](http://vi.wikipedia.org/wiki/C) một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy.

[Cú pháp](http://vi.wikipedia.org/wiki/C%C3%BA_ph%C3%A1p_h%E1%BB%8Dc) Java được vay mượn nhiều từ [C](http://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) & [C++](http://vi.wikipedia.org/wiki/C%2B%2B) nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lí bởi Java Virtual Machine (JVM) bằng cách tự động “dọn dẹp rác”. Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sở dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

Dùng các phần mềm lập trình thông dụng như: Java, netbeans.....

Một số phần mềm chỉnh sửa ảnh: Photoshop

Một số phần mềm âm thanh: Sound Force

# PHÂN TÍCH & THIẾT KẾ TRÒ CHƠI

## 1. Lập bảng ghi thông tin các thành viên trong nhóm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stt | Mã SV | Họ Tên | SĐT | Email |
| 1 | 14ĐC150 | Đặng Thị Thanh Nhàn | 0976321840 | dangthithanhnhan1996@gmail.com |
| 2 | 14ĐC063 | Huỳnh Diệp Phụng | 01635300108 | huynhdiepphung109@gmail.com |
| 3 | 14ĐC090 | Nguyễn Công Toàn | 0919970611 | congtoan0101@gmail.com |

## 2. Lập bảng ghi thông tin mỗi thành viên chịu trách nhiệm hiện thực hay kiểm tra chức năng.

|  |  |  |
| --- | --- | --- |
| Tên thành viên | Chức năng thực hiện | Kiểm tra chức năng |
| Đặng Thị Thanh Nhàn | Xây dựng khung chương trình và các quang cảnh cần thiết trong trò chơi | Xây dựng các đối tượng cần thiết trong trò chơi |
| Huỳnh Diệp Phụng | Xây dựng các đối tượng cần thiết trong trò chơi | Xây dựng các mức độ trò chơi khi người dùng chơi với máy tính và nhiều người cùng chơi với nhau |
| Nguyễn Công Toàn | Xây dựng các mức độ trò chơi khi người dùng chơi với máy tính và nhiều người cùng chơi với nhau | Xây dựng khung chương trình và các quang cảnh cần thiết trong trò chơi |
| Huỳnh Diệp Phụng, Đặng Thị Thanh Nhàn, Nguyễn  Công Toàn | Xây dựng các setting cần thiết để người chơi điều chỉnh. |  |

## 3. Các hàm xử lý chính trong file thực hiện chức năng X.

### 3.1 Chức năng 1: Xây dựng khung chương trình và các quang cảnh cần thiết trong trò chơi:

**Các hàm xử lý chính** public void init() { } public void drawBackGroundWithTileImage(int w, int h, Graphics2D g2) { } public void drawDemo(int w, int h, Graphics2D g2) { } public class Island { } public void update(int w, int h) { } public void draw(Graphics g, ImageObserver obs) { }



### 3.2 Chức năng 2: Xây dựng các đối tượng cần thiết trong trò chơi

**Hàm xử lý chính của đối tượng máy bay 1:**  public class MyPlane extends GameObject implements Observer

{ public boolean collision(int x, int y, int w, int h) { } public void fire() { } public void draw(Graphics g, ImageObserver obs) { } public void update(Observable obj, Object arg) { } public void update(Gm1942 vm) { }

}

**Hàm xử lý chính của đối tượng máy bay 2 :**

public class MyPlane2 extends GameObject implements Observer

{ public boolean collision(int x, int y, int w, int h) { } public void fire() { } public void draw(Graphics g, ImageObserver obs) { } public void update(Observable obj, Object arg) { } public void update(Gm1942 vm) { }

}



**Hàm xử lý đối tượng địch** public class EnemyPlane extends GameObject

{ public boolean collision(int x, int y, int w, int h) { } public void fire() { } public void draw(Graphics g, ImageObserver obs) { } public void update(Observable obj, Object arg) { } public void update(Gm1942 vm) { }

}



**Hàm xử lý đối tượng viên đạn**

public class Bullet extends GameObject

{ public void setBulletType(int type){ } public boolean collision(int x, int y, int w, int h) { } public void draw(Graphics g, ImageObserver obs) { } public void update(Gm1942 vm) { } public void checkCollision(int x, int y, int sizeX, int sizeY){ } }



**Hàm xử lý thanh năng lượng** public class PowerUp extends GameObject

{ public boolean collision(int x, int y, int w, int h) { } public void draw(Graphics g, ImageObserver obs) { } public void update(Gm1942 vm) { }

}



## 4. Các ưu điểm và hạn chế khi làm bài tập nhóm

Ưu điểm

* Có sự hợp tác tốt giữa các thành viên để làm ra một sản phẩm game
* Biết cách xây dựng một game cơ bản bằng Java

Nhược điểm

* Chưa xây dựng đầy đủ các chức năng như yêu cầu
* Giao diện chưa được đẹp mắt hấp dẫn người chơi

## 5. Các kinh nghiệm học được từ bài tập nhóm

Thông qua bài tập nhóm này chúng em đã học được nhiều thứ. Từ cách làm việc nhóm thế nào cho hiệu quả đạt kết quả tốt. Học được thêm kiến thức java để làm một game. Cách làm việc có xây dựng lịch trình rõ ràng, sự tương tác tốt giữa các nhóm với nhau.

## 6. Hướng dẫn sử dụng game

Khi mở lên game bắt đầu liền

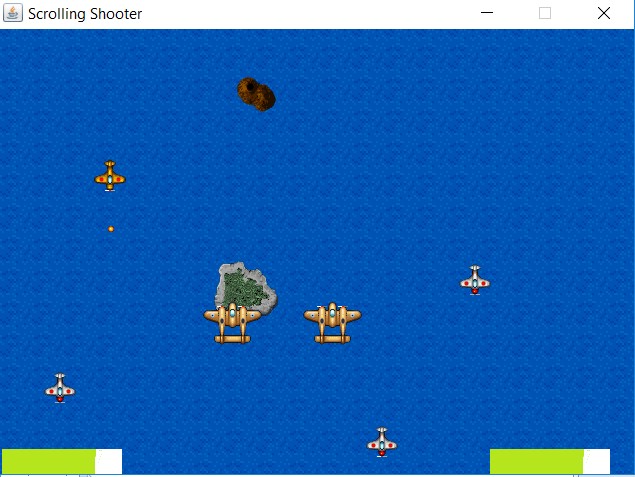
Máy bay 1: Dùng các nút nút điều hướng lên, xuống , trái phải để điều khiển máy bay.

Dùng phím enter để bắn máy bay địch. Thanh máu sẽ bị giảm dần nếu bị máy bay địch bắn trúng

Máy bay 2: Dùng các nút W, A, S, D để điều khiển máy bay. Dùng phím space để bắn máy bay địch.

Một trong hai thanh năng lượng cạn thì máy bay còn lại vẫn tiếp tục được chơi.

Khi hai máy bay đều hết năng lượng thì game kết thúc



# TƯ LIỆU THAM KHẢO

[1] <http://gamestudio.vn/tin-tuc/kinh-nghiem-lam-game/lap-trinh-game-voi-nguoi-moi-bat-dau-225.html>

[2]<https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)>

[3] https://www.youtube.com/watch?v=1gir2R7G9ws