ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ────────



**BÀI TẬP LỚN**

MÔN HỆ NHÚNG

TÊN ĐỀ TÀI: XÂY DỰNG TRÒ CHƠI SPACE INVADERS

Giảng viên hướng dẫn: TS. Ngô Lam Trung

Danh sách sinh viên:

|  |  |
| --- | --- |
| **Họ tên sinh viên** | **Mã số sinh viên** |
| Trịnh Quốc Đạt | 20194248 |
| Trần Đức Hải | 20194270 |
| Tô Đức Hiệp | 20194278 |

***Hà nội, tháng 08 năm 2022***

**MỤC LỤC**

[Chương 1. Mô tả đề tài 3](#_Toc111375421)

[I. Mô tả bài toán 3](#_Toc111375422)

[**1.** **Giới thiệu bài toán** 3](#_Toc111375423)

[**2.** **Các chức năng** 3](#_Toc111375424)

[II. Phân công công việc 3](#_Toc111375425)

[Chương 2. Phân tích thiết kế 4](#_Toc111375426)

[I. Phần mềm 4](#_Toc111375427)

[**1.** **Phân tích chức năng** 4](#_Toc111375428)

[**2.** **Phân tích cấu trúc** 4](#_Toc111375429)

[**3.** **Phân tích hành vi** 6](#_Toc111375430)

[II. Phần cứng 7](#_Toc111375431)

[**1.** **Thành phần** 7](#_Toc111375432)

[**2.** **Kỹ thuật sử dụng** 10](#_Toc111375433)

[Chương 3. Kết luận 14](#_Toc111375434)

[I. Kết quả thực hiện 14](#_Toc111375435)

[II. Kết luận 18](#_Toc111375436)

[III. Hướng phát triển 19](#_Toc111375437)

[TÀI LIỆU THAM KHẢO 19](#_Toc111375438)

# Chương 1. Mô tả đề tài

1. **Mô tả bài toán**
2. **Giới thiệu bài toán**

Mục đích của project là mô phỏng lại trò chơi arcade nổi tiếng Space Invaders sử dụng kit STM32F429ZIT6 ARM Cortex-M4, người chơi điều khiển con tàu của mình (lên, xuống, sang phải, sang trái) bằng cách nghiêng màn hình, mục tiêu là bắn trúng các kẻ thù ở phía trên màn hình và đạt được điểm số cao nhất có thể. Kẻ thù cũng có thể bắn đạn, con tàu của người chơi phải né đạn của kẻ thù để tránh bị mất mạng. Nếu người chơi mất 5 mạng hoặc đánh bại được boss ở màn chơi cuối cùng, trò chơi sẽ kết thúc.

Project được lập trình bằng ngôn ngữ C/C++ sử dụng nền tảng phát triển STM32CubeIDE, hỗ trợ bởi framework phần mềm đồ họa TouchGFX để thiết kế đồ họa nhanh và dễ dàng hơn (thông qua kéo-thả).

1. **Các chức năng**

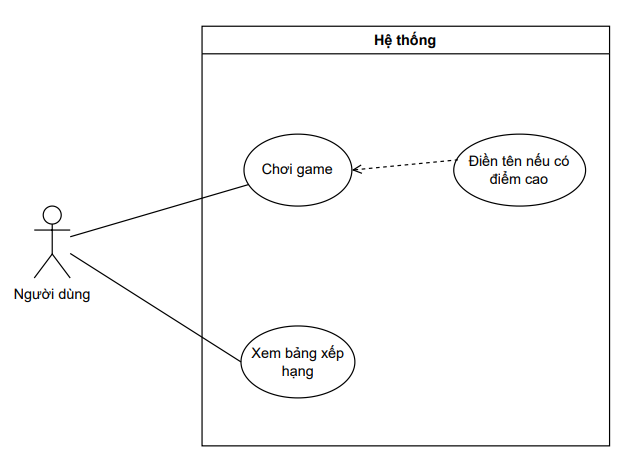
* Người dùng có thể chơi game
* Người dùng có thể xem bảng xếp hạng
* Người dùng nhập tên khi đạt được điểm cao
* Trò chơi có đồ họa đẹp, gây hứng thú cho người chơi

1. **Phân công công việc**

|  |  |
| --- | --- |
| **Nhiệm vụ** | **Người tham gia** |
| Tìm hiểu, thiết kế hệ thống | Trịnh Quốc Đạt |
| Lập trình lớp ship | Tô Đức Hiệp |
| Lập trình các lớp enemy | Trần Đức Hải |
| Lập trình các giai đoạn game | Tô Đức Hiệp |
| Xử lý dữ liệu từ cảm biến | Trịnh Quốc Đạt |
| Lập trình, thiết kế giao diện | Trần Đức Hải  Tô Đức Hiệp |
| Xử lý nhập, xuất bảng xếp hạng | Trịnh Quốc Đạt |

# Chương 2. Phân tích thiết kế

1. **Phần mềm**
2. **Phân tích chức năng**



Hình 1. Biểu đồ ca sử dụng tổng quan

* Chức năng chơi game: Người dùng có thể chọn chơi game, trong game người dùng sẽ điều khiển phi thuyền, bắn các con quái để chiến thắng. Hệ thống sẽ lưu lại điểm của người dùng. Nếu khi game kết thúc, người dùng có điểm cao hơn điểm trong bảng xếp hạng, người dùng sẽ thêm tên mình vào bảng xếp hạng.
* Xem bảng xếp hạng: Khi người dùng chọn xem bảng xếp hạng, hệ thống sẽ hiện thị bảng xếp hạng gồm danh sách các người chơi có số điểm cao nhất

1. **Phân tích cấu trúc**

Trong quá trình, phân tích thiết kết game, nhóm đã quyết định trò chơi sẽ có phi thuyền(Ship) và 3 loại quái(Enemy), mỗi loại quái sẽ có hành vi khác nhau.

* 1. Biểu đồ lớp enemy

Diagram

Description automatically generated

Hình 2. Biểu đồ các lớp enemy

Các lớp enemy được thiết kế với một lớp interface tăng tính tái sử dụng và sự thuận tiện cho việc triển khai. Các lớp con sẽ ghi đè phương thức move() để mỗi enemy sẽ di chuyển khác nhau. Lớp enemy1 và lớp boss sẽ có thêm phương thức fireBullet() để bắn đạn.

* 1. Biểu đồ lớp ship

Table

Description automatically generated

Hình 3. Biểu đồ lớp Ship

Lớp Ship có các thuộc tính cơ bản là số mạng(lives). Lớp ship có thể di chuyển hoặc bắn đạn.

1. **Phân tích hành vi** 
   1. Biểu đồ trạng thái enemy

Diagram, schematic

Description automatically generated

Hình 4. Biều đồ trạng thái enemy

Các đối tượng Enemy có 5 trạng thái:

* OOB: Đợi ngoài màn hình.
* ENTER: Bắt đầu vào màn hình.
* RETREAT: Rút lui khi hết thời gian màn chơi.
* ATTACK: Tấn công theo thời điểm được định trước.
* DEAD: Chết khi nhận sát thương đủ lớn.
  1. Biểu đồ trạng thái ship

Diagram

Description automatically generated

Hình 5. Biểu đồ trạng thái ship

Các đối tượng Ship có 4 trạng thái:

* OOB: Đợi ngoài màn hình.
* ALIVE: Trạng thái hiển thị còn sống khi đang tham gia màn chơi.
* IMMUNE: Trạng thái không nhận xát thương khi vào màn chơi mới hoặc sau khi chết.
* DEAD: Trạng thí khi chết.

1. **Phần cứng**
2. **Thành phần**

Hệ thống gồm 3 thành phần chính:

* 32F429IDISCOVERY Discovery kit
* Cảm biến gyroscope I3G4250D
* Ngoại vi RTC
  1. . 32F429IDISCOVERY

32F429IDISCOVERY là một kit phát triển được cung cấp bởi STMicroelectronics. Nó có bộ vi điều khiển cực kỳ mạnh mẽ STM32F429ZIT6 với 32bit arm cortex M4.

A close-up of a circuit board

Description automatically generated with medium confidence

Hình 6. STM32F429I-DISC1

* Các tính năng chính được sử dụng trong bài tập lớn:
  + Bộ vi điều khiển STM32F429ZIT6 có 2 Mbyte bộ nhớ Flash, 256 Kbytes RAM
  + Màn hình LCD 2,4 inch QVGA
  + 64-Mbit SDRAM
  + Cảm biến chuyển động I3G4250D, ST-MEMS Con quay hồi chuyển đầu ra kỹ thuật số 3 trục [1].
  1. . Gyroscope I3G4250D

Cảm biến Gyro 3 trục I3G4250D được sử dụng để đo tốc độ góc quay của vật thể gắn cảm biến theo ba trục x, y, z, cảm biến có độ phân giải rất cao (16 bit) có để đo ở tốc độ 245/500/2000 độ/giây (dps) ổn định mà không bị ảnh hưởng bởi nhiệt độ. I3G4250D có thể hoạt động ổn định từ -40 ° C đến +85 ° C. Để giao tiếp với cảm biến, ta có thể sử dụng chuẩn giao tiếp SPI hoặc I2C. Cụ thể trong bài tập lớn này ta sẽ sử dụng chuẩn giao tiếp SPI. [3]

Diagram, schematic

Description automatically generated

Hình 7. Schematic ngoại vi [1]

Diagram

Description automatically generated

Hình 8. Schematic MCU [1]

Ở hình 7, có thể xem như là schematic của I3G4250D. Qua schematic của STM32F429I-DISC1 ta có thể thấy cảm biến gyro kết nối với MCU qua SPI5, trong đó chân PC1 là chân Chip Select.

* 1. . RTC

RTC hay Real Time Clock là ngoại vi sử dụng như một bộ đồng hồ – lịch thời gian thực. RTC cung cấp lịch , thời gian trong ngày, hai Alarm có thể lập trình và lập trình định kỳ cờ wakeup với khả năng ngắt. RTC cũng bao gồm một đơn vị đánh thức tự động để quản lý Low Power Modes. Nhưng trong bài tập lớn, chúng ta sẽ không sử dụng chức năng lập lịch hay báo thức, mà chúng ta sẽ sử dụng các thanh ghi backup của RTC như 1 Eprom. [2]

Graphical user interface, application, table

Description automatically generated

Hình 9. RTC backup registers [2]

RTC cung cấp 20 thanh ghi 32bit để lưu trữ 80 byte cho chương trình ứng dụng người dùng. Các thanh ghi backup của RTC đều thuộc backup domain được cung cấp nguồn bởi VDD hoặc VBAT khi VDD bị tắt. Chính vì vậy chúng sẽ không bị xóa khi hệ thống reset.

1. **Kỹ thuật sử dụng**
   1. . Điều khiển Ship



Đây là luồng dùng để đọc dữ liệu từ cảm biến gyro sau đó gửi lệnh cho luồng giao diện để điều khiển ship:

* Lời gọi hàm GyroscopeDrv->GetXYZ(pfData) dùng để lấy dữ liệu từ cảm biến, lưu vào mảng 3 phần tử pfData. Ở đây chúng ta chỉ cần hai trục x và y để điều khiển ship nên sẽ chỉ sử dụng pfData[0] và pfData[1].
* ABS(pfData[0]) > ABS(pfData[1]). Do trong các thao tác thực tế khi di chuyển 1 trục thì các trục còn lại còn thay đổi, nên ta cần xác định trục nào thay đổi chính bằng cách so sánh giá trị tuyệt đối.
* Đồng thời, trong thực tế, tùy thuộc môi trường mà dữ liệu cảm biến có thể bị ảnh hưởng bởi nhiễu. Vì vậy trong bài tập lớn này chúng ta sẽ sử dụng mốc 50000.0f để khử nhiễu.
* Khi đó, nếu giá trị trục x lớn hơn 50000.0f n thì sẽ gửi lệnh di chuyển lên cho ship qua messageQueue Queue1Handle, nếu giá trị x bé hơn -50000.0f thì sẽ gửi lệnh di chuyển xuống. Nếu giá trị y lớn hơn 50000.0f thì sẽ gửi lệnh di chuyển sang phải, nếu giá trị y bé hơn -50000.0f thì sẽ gửi lệnh di chuyển sang trái.



Ở luồng giao diện, ta sẽ nhận thông điệp qua messageQueue Queue1Handle rồi di chuyển ship theo thông điệp nhận được.

* 1. . Lưu bảng xếp hạng

Để lưu bảng xếp hạng, ta sẽ chia 20 ngăn nhớ 32bit của RTC thành 5 nhóm tương ứng với 5 chỉ mục của bảng xếp hạng, mỗi nhóm gồm 4 ngăn nhớ. Trong mỗi nhóm, 3 ngăn nhớ đầu hay 96 bit đầu sẽ dùng để lưu tên người chơi, ngăn nhớ cuối sẽ dùng để lưu điểm. Dưới đây là hàm dùng để lấy bảng xếp hạng.



* Vì Touchgfx sử dụng UnicodeChar có độ lớn là 16 bit để biểu diễn Text nên mỗi tên của người chơi sẽ có 6 ký tự.
* Để lấy tên và điểm của các người chơi thì chúng ta chỉ cần đọc ngăn nhớ tương ứng với vị trí của người chơi đó qua hàm HAL\_RTCEx\_BKUPRead và hàm memcpy. Trong đó RTC\_BKP\_DR0 là ngăn nhớ đầu tiên.

# Chương 3. Kết luận

1. **Kết quả thực hiện**

Project đã thành công trong việc hoàn thành các mục tiêu đặt ra mặc dù các chức năng có thể được cải thiện và tinh chỉnh, tối ưu hơn nữa. Các lỗi gặp phải trong quá trình lập trình đều đã được sửa.

* Các đối tượng trong game
* Tàu của người chơi: đạn sẽ được bắn tự động khoảng 4 viên/giây



Hình 10. Ship

* Kẻ thù: có 3 loại từ trái qua phải: loại 0 (màu tím, không thể bắn đạn), loại 1 (màu vàng, có thể bắn đạn), boss (có thể bắn đạn)

A picture containing text

Description automatically generated

Hình 11. Enemy

* Giao diện khởi đầu: người chơi chọn nút Start để chơi game, chọn Leaderboard để xem bảng xếp hạng

**Graphical user interface, application

Description automatically generated**

Hình 12. Màn hình menu

* Chạy màn chơi thứ nhất:
* Có 4 kẻ thù loại 0 (màu tím) đứng cố định, không bắn đạn
* Có 1 kẻ thù loại 1 (màu vàng) di chuyển từ phải sang trái và bắn đạn, người chơi cần né đạn từ kẻ thù này
* Khi hết màn 1 (kéo dài khoảng 10 giây), các kẻ thù loại 0 mà chưa bị tiêu diệt sẽ lùi lại đến hết màn hình và biến mất, nếu người chơi hết mạng, trò chơi sẽ kết thúc, nếu không màn 2 sẽ bắt đầu.

Background pattern

Description automatically generated

Hình 13. Phase 1

* Chạy màn chơi thứ hai:
* Có 10 kẻ thù loại 0 đứng cố định, 3 kẻ thù loại 1 (2 trong số 3 kẻ thù này sẽ xuất hiện từ phải sang trái, 1 sẽ xuất hiện từ trái sang phải)
* Khi hết màn 2, các kẻ thù loại 0 mà chưa bị tiêu diệt sẽ lao thẳng xuống dưới màn hình, nếu tàu của người chơi bị đâm trúng sẽ mất 1 mạng.



Hình 14. Phase 2

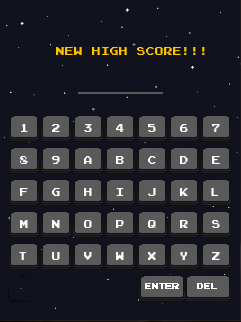
* Chạy màn chơi thứ ba:
* Màn chơi này sẽ không có giới hạn thời gian. Người chơi sẽ phải đối đầu với con boss có thể bắn 2 loại đạn , cứ khoảng 5 giây 1 lần sẽ có kẻ thù loại 1 bay từ phải qua trái và bắn đạn

A screenshot of a video game

Description automatically generated

Hình 15. Phase 3

* Nếu đạt điểm cao, người dùng sẽ chuyển đến giao diện nhập tên, nhập tên để thêm tên mình vào bảng xếp hạng. Giao diện nhập tên:



Hình 16. Màn hình điểm cao mới

* Bảng xếp hạng gồm có danh sách các người chơi có số điểm cao nhất. Giao diện bảng xếp hạng:

A screenshot of a video game

Description automatically generated

Hình 17. Màn hình bảng xếp hạng

1. **Kết luận**

Nhóm chúng em tin rằng đây là một project thú vị về một trò chơi arcade những năm 70. Project có hiệu ứng cháy nổ, giao diện dễ nhìn và cách chơi khá đơn giản. Phiên bản Space Invaders này có chút khác biệt với phiên bản gốc, về hình ảnh lẫn cách chơi nhưng vẫn giữ được các yếu tố cơ bản của trò chơi arcade kinh điển này.

Nhóm chúng em muốn gửi lời cảm ơn tới thầy giáo Ngô Lam Trung đã hướng dẫn chúng em trong quá trình làm project. Và mong rằng thầy và các bạn trong lớp sẽ cảm thấy thích thú với trò chơi này.

1. **Hướng phát triển**

Về thiết kế trò chơi, có thể thêm các power-up, thêm các màn chơi hơn, tăng độ thú vị của trò chơi.

Về di chuyển ship, do cảm biến gyro chỉ có thể đo được gia tốc góc của thiết bị nên việc điều khiển ship còn khá đuồi. Có thể kết hợp cảm biến gyro với cảm biến gia tốc để xác định chính xác yaw, pitch, roll khiến việc điều khiển mượt mà hơn.

Hiện tại, dù dữ liệu bảng xếp hạng vẫn tồn tại khi thiết bị bị reset nhưng nếu rút nguồn thì dữ liệu bảng xếp hợp sẽ bị xóa hết. Do đó, có thể kết nối thêm pin bên ngoài qua chân Vbat, khi đó bảng xếp hạng sẽ không bị xóa.

Ngoài ra, RTC chỉ có 20 ngăn nhớ 32bit nên trong bài tập lớn này chúng ta chỉ có thể lưu được 5 chỉ mục của bảng xếp hạng. Nếu muốn lưu nhiều hơn chúng ta có thể sử dụng SRAM làm backup.

# TÀI LIỆU THAM KHẢO

1. UM1670 User manual
2. RM0090 Reference manual
3. I3G4250D Datasheet