

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



NHÓM GỒM 2 THÀNH VIÊN:

- | | |
|----------------------------|-----------------------|
| 1. LÝ HÁN ĐỒNG | MSSV: 18120318 |
| 2. NGUYỄN QUỐC DŨNG | MSSV: 18120333 |

BÁO CÁO ĐỒ ÁN 2

ĐỒ ÁN HƯỚNG ĐỐI TƯỢNG

Môn học: Phương Pháp Lập Trình Hướng Đối Tượng
Lớp: 18CTT3A

Thành phố Hồ Chí Minh – 2019

MỤC LỤC

CHƯƠNG 0: NHẬN XÉT ĐỒ ÁN TRƯỚC.....	3
0.1. Giao diện:.....	3
0.2. Xử lý:.....	3
CHƯƠNG I: TIẾN ĐỘ ĐỒ ÁN.....	4
CHƯƠNG II: PHÂN CHIA CÔNG VIỆC.....	5
CHƯƠNG III: SƠ ĐỒ LỚP (UML).....	7
CHƯƠNG IV: TRÌNH BÀY VÀ GIẢI THÍCH.....	8
4.1. Đối tượng CFOOD:.....	8
4.1.1. Các thuộc tính:.....	8
4.1.2. Các chức năng:.....	8
4.2. Đối tượng CSAVE:.....	11
4.3. Chỉnh sửa CGAME:.....	11
CHƯƠNG V: LƯU ĐỒ THUẬT TOÁN.....	13
CHƯƠNG VI: HƯỚNG DẪN SỬ DỤNG.....	14
6.1. Hướng dẫn sử dụng:.....	14

CHƯƠNG 0: NHẬN XÉT ĐỒ ÁN TRƯỚC

0.1. Giao diện:

Giao diện console chưa được đẹp, chớp nháy tạo cảm giác khó chịu khi chơi, không có hướng dẫn chơi trong game khiến bước đầu tiếp cận khó khăn.

0.2. Xử lý:

Không tạo tiến trình song song để bắt sự kiện nhận phím gây khó khăn cho việc lập trình lẫn người chơi.

Quả bóng không tuân theo va chạm vật lý khi chạm vào thanh trượt người chơi, một số trường hợp dù thanh trượt đỡ được nhưng vẫn đi xuyên qua dẫn đến thua.

Đặt phần xử lý kết thúc game ngay trong phần vẽ là hơi bất hợp lý.

Không có menu tùy chọn lúc bắt đầu game.

Không làm cho máy tự động chơi được.

CHƯƠNG I: TIẾN ĐỘ ĐỒ ÁN

Yêu cầu của đồ án	Tiến độ (%)
- Hiện màn hình chơi cùng các vật phẩm của game	100
- Xuất hiện vật phẩm thưởng dạng random	100
- Save game và bảng thành tích	100
- Xử lý thắng thua và kết thúc game	100
- Tăng tốc độ khi va chạm vào thanh trượt người chơi	100
- Thiết kế giao diện với màu sắc đẹp	80
- Cho thanh trượt tự động chạy để hứng quả bóng	100

CHƯƠNG II: PHÂN CHIA CÔNG VIỆC

Ghi chú:

- **18120318: Lý Hán Đồng (đảm nhiệm)**

- **18120333: Nguyễn Quốc Dũng (đảm nhiệm)**

Công việc	Mô tả	18120318	18120333
Công tác chuẩn bị và lên ý tưởng	<ul style="list-style-type: none"> - Phân tích đề, hiểu đề. - Đưa ra ý kiến về các đối tượng cần phân chia là: CFOOD, CSAVE. - Đọc hiểu, tìm hiểu về phần source code được tiếp nhận để phát triển. 	X	X
Thiết kế đồ họa cho game	- Thiết kế đồ họa cho các đối tượng vật phẩm trong game.	X	
	- Chỉnh màu sắc cho game		X
Thiết kế đồ họa cho giao diện	- Thiết kế menu cho giao diện khởi đầu		X
	- Thiết kế giao diện kết thúc, bảng thành tích	X	
Code	<ul style="list-style-type: none"> - CFOOD (Đối tượng vật phẩm) - CSAVE (Đối tượng dùng quản lý file save) 	X	

	<ul style="list-style-type: none"> - Chỉnh sửa CGAME để thêm chế độ mới vào game. - Quản lý các đối tượng của chế độ mới 		X
Kiểm thử, dò lỗi	<ul style="list-style-type: none"> - Xem bản Release có hoạt động ổn không. - Phát hiện các lỗi để sửa chữa kịp thời 	X	X
Viết báo cáo	<ul style="list-style-type: none"> - Sơ đồ lớp - Lưu đồ thuật toán - Hướng dẫn sử dụng - Giải tích, trình bày 	X	X

CHƯƠNG III: SƠ ĐỒ LỚP (UML)

C SAVE
<u>+ Save(CBALL* , CBAR* , int , vector<CFOOD> , vector<CFOOD> , vector<CFOOD> , int) : void</u> <u>+ Load_game() : vector<int></u> <u>+ High_scores(int , string) : void</u> <u>+ Load_highscores() : vector<string></u>

CFOOD
- _topLeft_x : int - _topLeft_y : int - _botRight_x : int - _botRight_y : int - console : HANDLE
<u>+ GotoXY(int , int) : void</u> <u>+ Generate() : vector<CFOOD></u> <u>+ Obstacles() : vector<CFOOD></u> <u>+ Prize() : vector<CFOOD></u> <u>+ AddFoods(vector<CFOOD>& , int) : void</u> <u>+ AddPrizes(vector<CFOOD>& , int) : void</u> + TopLeftX() : int + SetTopLeftX() : int + TopLeftY() : int + SetTopLeftY() : int + BotRightX() : int + SetBotRightX() : int + BotRightY() : int + SetBotRightY() : int + CFOOD(int , int , int , int) + Check_collision(CBALL*) : eDir + IsCollisionball(CBALL*) : bool + IsCollision(CFOOD) : bool + Draw_food(int , int) : bool + Draw_obstacles(int , int) : bool + Draw_prize(int , int) : bool

CHƯƠNG IV: TRÌNH BÀY VÀ GIẢI THÍCH

4.1. Đối tượng C FOOD:

Dùng để biểu thị vật phẩm trong game.

4.1.1. Các thuộc tính:

Tên	Mô tả
_topLeft_x	Tọa độ x góc trái trên
_topLeft_y	Tọa độ y góc trái trên
_botRight_x	Tọa độ x góc phải dưới
_botRight_y	Tọa độ y góc phải dưới
console	Handle của console

4.1.2. Các chức năng:

- Các hàm lấy và gán giá trị cho các thuộc tính.

- **C FOOD(int, int, int, int):** Hàm chạy khi khởi tạo một C FOOD với các tọa độ

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
int : giá trị _topLeft_x	Không có
int : giá trị _topLeft_y	
int : giá trị _botRight_x	
int : giá trị _botRight_y	

- **Check_collision(CBALL*)**: Kiểm tra va chạm với quả bóng, nếu va chạm trả về hướng đi của quả bóng.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
CBALL* : quả bóng hiện đang xem xét	eDir : hướng đi của quả bóng

- **IsCollision(CFOOD)**: Kiểm tra va chạm với CFOOD khác

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
CFOOD	bool : kết quả xác nhận có va chạm hay không.

- **Draw_food(int, int)**: Kiểm tra tại tọa độ này có thành phần của food hay không, nếu có thì vẽ thành phần tại tọa độ đó.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
int : tọa độ x trên màn hình int : tọa độ y trên màn hình	bool : kết quả là đã vẽ hay không

- **Draw_obstacles(int, int)**: Kiểm tra tại tọa độ này có thành phần của obstacle hay không, nếu có thì vẽ thành phần tại tọa độ đó.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
int : tọa độ x trên màn hình int : tọa độ y trên màn hình	bool : kết quả là đã vẽ hay không

- **Draw_prize(int, int):** Kiểm tra tại tọa độ này có thành phần của prize hay không, nếu có thì vẽ thành phần tại tọa độ đó.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
int : tọa độ x trên màn hình int : tọa độ y trên màn hình	bool : kết quả là đã vẽ hay không

- **Generate():** Khởi tạo random vị trí các vật phẩm food trong game.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
Không có	vector<CFOOD>

- **Obstacles():** Khởi tạo vị trí các vật cản trong game.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
Không có	vector<CFOOD>

- **Prize():** Khởi tạo random vị trí các vật phẩm thưởng trong game.

Giá trị nhận vào (theo thứ tự)	Giá trị trả ra
Không có	vector<CFOOD>

4.2. Đối tượng CSAVE:

Dùng để quản lý lưu dữ liệu game ra file và load lại.

Các chức năng:

- **Save(CBALL*, CBAR*, int, vector<CFOOD>, vector<CFOOD>, vector<CFOOD>, int):** Hàm lưu trạng thái game xuống file

- **Load_game():** Hàm load dữ liệu từ file save, trả về vector<string>

- **High_scores(int, string):** Hàm lưu thành tích người chơi vào file

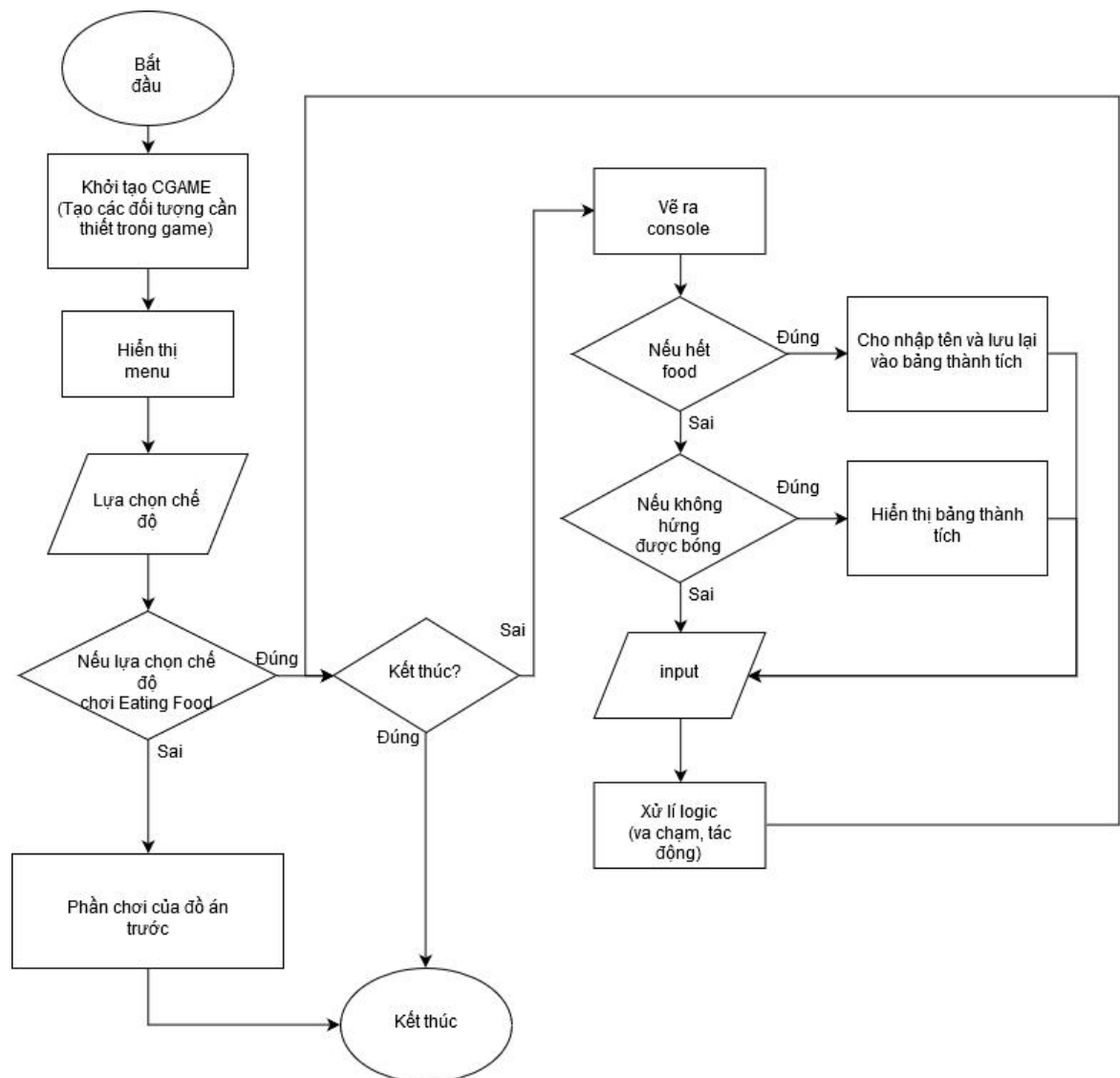
- **Load_highscores():** Hàm load dữ liệu từ file save của highscore, trả về vector<string>

4.3. Chỉnh sửa CGAME:

- Thêm menu để vào có thể phân biệt các chế độ chơi.
- Thêm chức năng tự di chuyển cho thanh trượt của người chơi.
- Bỏ xóa màn hình để ngăn hiện tượng hình ảnh giật lag
- Thêm tăng tốc 10% cho mỗi lần va chạm vào thanh trượt người chơi.
- Chuyển random hướng khi va chạm người chơi thành tuân thủ quy tắc vật lý.
- Thêm tác động đổi hướng phù hợp với sự di chuyển của thanh trượt.
- Thêm các vector lưu trữ và quản lý các đối tượng mới CFOOD

- Xử lý các tình huống va chạm giữa quả bóng với CFOOD
- Thêm các trường hợp thưởng của prize: x2 điểm, x3 điểm, thêm 3 food, thêm 2 prize, tạo vật cản, bật chế độ phá hủy vật cản.
- Thêm giới hạn thời gian chơi.
- Thay đổi việc vẽ cho phù hợp với chế độ mới.
- Bổ sung kết thúc game, nếu thắng (khi không còn food nào) thì cho nhập tên và lưu vào bảng thành tích, nếu thua (khi không đỡ được bóng hoặc hết thời gian) thì chỉ hiện bảng thành tích.
- Thêm vào phần xử lý input khi nhấn phím 'L' thì sẽ lưu game lại ra file.

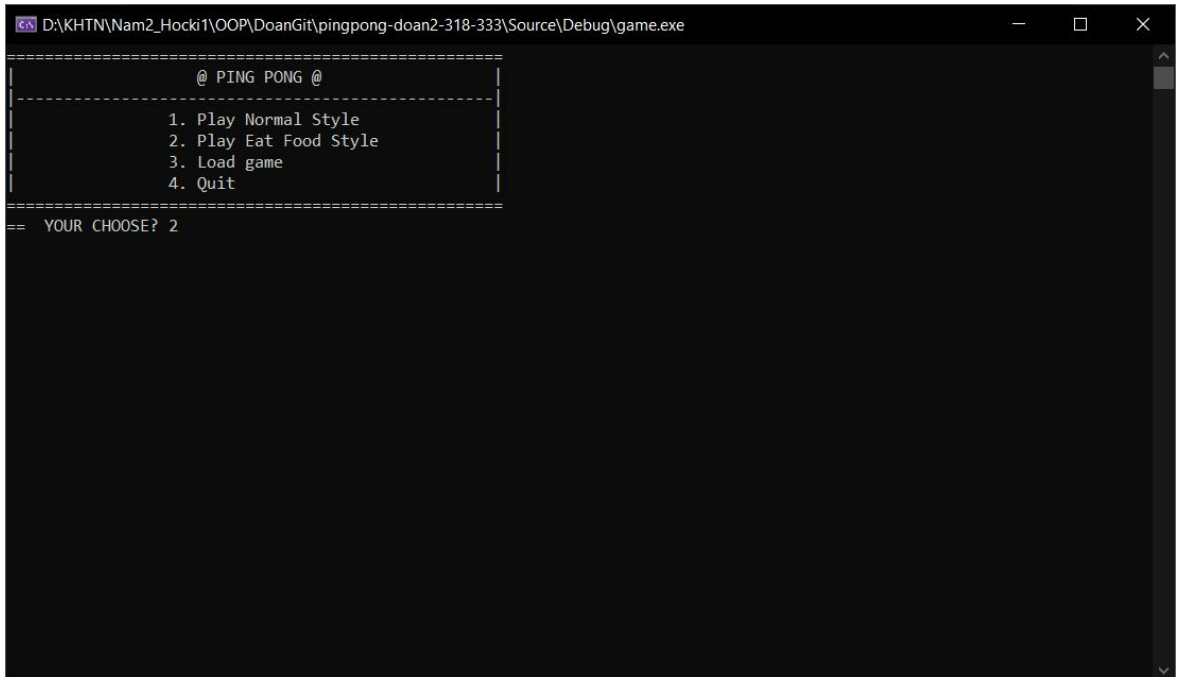
CHƯƠNG V: LƯU ĐỒ THUẬT TOÁN



CHƯƠNG VI: HƯỚNG DẪN SỬ DỤNG

6.1. Hướng dẫn sử dụng:

Giao diện ban đầu thể hiện menu, muốn vào chơi chế độ của đồ án 2 thì chúng ta nhập vào tùy chọn 2



Vào phần chơi, nhấn phím bất kì để bắt đầu, sử dụng phím W để di chuyển lên và S để di chuyển xuống, L để lưu lại.



Sau khi ăn hết thức ăn hoặc hết giờ hoặc không đỡ được bóng thì trò chơi sẽ kết thúc với bảng thành tích.

