**Bài tập lớn nhóm HAC**

**Tên game: Catch the candy**

**Lớp:** INT2215 25

**Giảng viên:** Đặng Cao Cường

**Thành viên:** Nguyễn Hoàng Trung

Nguyễn Phạm Minh Anh

Thân Hùng Cường

**Contact:** Nguyễn Hoàng Trung

**Mail**: [shp2582002@gmail.com](mailto:shp2582002@gmail.com)

**Phone:** 0836579638

Phân chia công việc:

|  |  |
| --- | --- |
| Nguyễn Hoàng Trung | Đưa ra ý tưởng, Phân chia thực hiện từng phần của Project, viết Common Function, BaseObject, MainObject, ThreatsObject, BulletObject, Debug |
| Nguyễn Phạm Minh Anh | Đưa ra ý tưởng, Design ảnh, game\_map, PlayerPower, Debug, Quay video demo sản phẩm |
| Thân Hùng Cường | Đưa ra ý tưởng, Tìm âm nhạc cho game, TextObject, Menu Game, Sound Game, Debug |

**A. Giới thiệu**

Catch the candy là game 2D được lập trình dựa trên ngôn ngữ C++ và sử dụng thư viện hỗ trợ: SDL. Catch the candy là một game có gameplay rất đơn giản: người chơi sẽ điều kiển nhân vật để hứng được những viên kẹo từ rơi từ trên xuống, trò chơi sẽ kết thúc nếu người chơi làm rớt quá 3 viên kẹo. Với mỗi một viên kẹo hưng thành công người chơi sẽ được cộng 5 điểm. Tốc độ của trò chơi sẽ tăng dần theo số điểm đạt được tương ứng với mỗi 100 điểm vận tốc sẽ tăng lên.

Trò chơi có cơ chế lưu điểm và điểm cao nhất giúp cho người chơi có thể dễ dàng theo dõi điểm của mình và điểm của đối thủ từ đó đánh bại và chiếm vị trí cao nhất.

**B. Về Game**

**I. Nền tảng**

Sử dụng ngôn ngữ lập trình C++, thư viện SDL 2.0 và các thư viện tiêu chuẩn

Game này được hoàn thành dựa trên sự tham khảo từ các nguồn:

Website: <http://lazyfoo.net/>

Website: <https://phattrienphanmem123az.com/>

**II. Code**

Để thuận tiện cho việc code, debug và teamwork game đã được chia thành các:

+ File Common Fuction

+ Lớp BaseObject

+ Các lớp đối tượng

+ Main

**1, File Common Fuction**

Là một file cơ bản nhất của Project có thể sử dụng ở mọi nơi của Project. Đồng thời đây cũng là nơi để gọi các file chưa các thư viện càn sử dụng đơn cử như:

- iostream

- fstream

- vector

- string

- SDL.h

- SDL\_image.h

- SDL\_ttf.h

- SDL\_mixer.h

- windows.h

Trong Common Fuction có 2 hàm:

bool CheckCollision(const SDL\_Rect& object1, const SDL\_Rect& object2): Kiểm tra vị trí của Rect với vị trí x, y, được đưa vào

bool CheckForcusWithRect(const int& x, const int& y, const SDL\_Rect& rect): Kiểm tra va chạm giữa 2 vật thể

**2, Lớp Base Object**

Đây là một lớp cha, đồng thời là lớp cơ sở đại diện hình ảnh được hiển thị trên màn hình.

Base Object được xây dựng là một class bên trong chứa:

SDL\_Texture\* p\_object\_;

SDL\_Rect rect\_;

Các biến có tác dụng lưu trữ dữ liệu và thông tin về ảnh như kích thước, màu sắc và giá trị của ảnh

Kèm theo còn có các hàm:

void SetRect(const int& x, const int& y): Đẩy vị trí vào trong BaseObject

SDL\_Rect GetRect(): lấy ra giá trị từ rect\_ trong BaseObject

bool LoadImg(std::string path, SDL\_Renderer\* screen): nhận địa chỉ ảnh, kiết xuất, lấy màu sắc và đưa ảnh vào biến nhớ

void Render(SDL\_Renderer\* des, const SDL\_Rect\* clip = NULL): Kiết xuất đối tượng

void Freee(): Giải phóng các đối tượng khi không còn sử dụng nữa

**3, Các lớp đối tượng**

Là những đối tượng hình ảnh được kế thừa từ lớp BaseObject. Mỗi class sẽ thể hiện những tính chất khác nhau

Các class bao gồm:

+ Main Object

+ Threats Object

+ BulletObject

+ Text Object

+ Game Map

+ PlayerPower

**a. Main Object**

Là Class dùng để xử lý những hình ảnh và tương tác của nhân vật do người chơi điều kiển

Các biến thành viên:

std::vector<BulletObject\*> p\_bullet\_list\_: dánh sách vật thả rơi(sẽ được đề cập ở dưới)

float x\_val\_;

float y\_val\_;

float x\_pos\_;

float y\_pos\_;

int width\_frame\_;

int height\_frame\_;

Các biết hỗ trợ xây dựng chuyển động của nhân vật

SDL\_Rect frame\_clip\_[8]: chia ra làm 8 cử động nhỏ để nhân vật di chuyển thật hơn

Input input\_type\_;

int frame\_: lưu nhân vật đang thực hiện cử động thứ.

int status\_: trạng thái của nhân vật

Các hàm thành viên:

virtual bool LoadImg(std::string path, SDL\_Renderer\* screen): đẩy từ file của đối tượng nhân vật lên

void Show(SDL\_Renderer\* des): Thể hiện hình ảnh khi nhân vật di chuyển qua trái hoặc phải

void HandelInputAction(SDL\_Event events, SDL\_Renderer\* screen): Nhận lệnh phím và đưa trạng trái dịch chuyển sang trái hoặc phải tương ứng theo phím được ấn xuống

void set\_clips(): tải các cử động di chuyển của nhân vật

void DoPlayer(Map& map\_data): dịch chuyển nhân vật sang trái hoạc phải tuỳ theo trạng thái đã được lưu một đoạn được đặt trước

void CheckToMap(Map& map\_data): kiểm tra nhân vật có di chuyển ra khỏi giới hạn màn hình hay chưa

SDL\_Rect GetRectFrame(): lấy thông số của ảnh

**b. Threats Object**

Là Class dùng để xử lý những hình ảnh và tương tác của nhưng đối tượng có nhiệm vụ là thả kẹo rơi xuống

Các biến thành viên:

int map\_x\_;

int map\_y\_;

float x\_val\_;

float y\_val\_;

float x\_pos\_;

float y\_pos\_;

bool on\_ground\_;

int come\_back\_time\_;

SDL\_Rect frame\_clip\_[THREAT\_FRAME\_NUM]: danh sách các đối tượng

int width\_frame\_;

int height\_frame\_;

int frame\_;

std::vector<BulletObject\*> bullet\_list: danh sách kẹo được thả xuống

Các hàm thành viên:

void set\_clips();

bool LoadImg(std::string path, SDL\_Renderer\* screen): Tải lên hình ảnh đối tượng

void Show(SDL\_Renderer\* des): đẩy hình ảnh lên screen

void DoPlayer(Map& gMap): đưa vật thể vào vị trí trên bản đồ

void CheckToMap(Map& gMap): kiểm tra va chạm theo chiều ngang

void RemoveBullet(const int& idx): xoá đạn sau khi đi hết map

SDL\_Rect GetRectFrame(): lấy giá trị vị trí được set up từ trước

**c. BulletObject**

Chịu trách nhiệm quản lý, xử lý thả rơi viên kẹo đồng.

Các biến thành viên:

int x\_val\_;

int y\_val\_;

int count\_;

int valtoend\_;

bool is\_move\_: kiểm tra trạng thái đã di chuyển hay chưa

unsigned int bullet\_dir\_: hướng di chuyển của viên kẹo

unsigned int bullet\_type\_: loại kẹo

Các hàm thành viên:

bool LoadImgBullet(SDL\_Renderer\* des): tải hình ảnh của viên kẹo từ file lên

void HandleMove(const int& x\_border, const int& y\_border): sinh ngẫu nhiên các viên kẹo bằng rand().

**d. Text Object**

Lớp quản lý hình dạng, font, màu sắc, vị trí của chữ

Các biến thành viên:

std::string str\_val\_;

SDL\_Color text\_color\_;

SDL\_Texture\* texture\_;

int width\_;

int height\_;

Các hàm thành viên:

enum Textcolor: Lưu màu sắc theo bảng màu RGB

bool LoadFromRenderText(TTF\_Font\* font, SDL\_Renderer\* screen): set up font chữ

void Free(): giải phỏng Text được tạo ra

void SetColor(int type): đặt màu theo bảng đã được lưu trừ trước

void RenderText(SDL\_Renderer\* screen, int xp, int yp,SDL\_Rect\* clip = NULL, double angle = 0.0, SDL\_Point \* center = NULL, SDL\_RendererFlip flip = SDL\_FLIP\_NONE): kêt xuất chữ ra để đưa lên hình ảnh

**e. Game map**

Sử dụng kỹ thuật Title map vào trong game để hỗ trợ kiểm tra va chạm một cách chính xác nhất. Class này quản lý kịch thước toạ độ và bố trí map.

Các biến thành viên:

Map game\_map\_;

TileMat tile\_mat\_[MAX\_TILES]: Lưu trữ hình dạng của từng loại ô

Input input\_type\_;

Các hàm thành viên:

void LoadMapTiles(SDL\_Renderer\* screen): đọc map từ file rồi đưa vào bộ nhớ

void LoadMap(char\* name): đưa map về dạng mảng 2 chiều để dễ dàng xử lý

void DrawMap(SDL\_Renderer\* des): dưa vào tính toàn trước đó tạo ra các ô với hình thái được định dạng từ trước

**f. PlayPower**

Lớp dùng để quản lý những đối tượng không tác động trên game vào. Ở đây là sinh mệnh của người chơi

Các biến thành viên:

int number\_: số lần chạm đất

std::vector<int> pos\_list\_: danh sách chưa vị trí biểu tượng

Các hàm thành viên:

void Init(SDL\_Renderer\* screen): Tải lên hoạt ảnh của sinh mạng người chơi

void Show(SDL\_Renderer\* screen): Hiện ra sinh mạng còn lại của người chơi

void Decrease(): Giảm sinh mạng của người chơi khi hứng trượt

**4. Main.cpp**

Đây là hàm chính của chương trình chịu trách nhiệm khởi tạo cũng như vận hành game:

Main.cpp sử dụng những hàm hỗ trợ sau

Bool Init(): Khởi tạo và thiết lập môi trường của game như: Lấy dữ liệu Highscore được

lưu trước trong file, khởi tạo cửa sổ game, màu sắc, tải lên font chữ từ file lên, tải lên âm

thanh từ trong file

bool loadBackground(): Tải lên hình ảnh nên của các đối tượng: Menu game, game và game over.

std::vector<ThreatsObject\*> MakeThreadList(): Khởi tạo các địa điểm thả kẹ

int loadMenu(): Tải lên Menu game với 3 lựa chọn: Start – vào chơi game, Highscore – xem top điểm cao của game, Exit – thoát game. Hàm sẽ trả về 3 giá trị tương ứng với từng lựa chọn khác nhau của người chơi

int highScore(): đưa ra top 10 người chơi đạt điểm cao nhất trong game ra màn hình

int checkEnd(): kiểm tra lựa chọn có chơi lại hay không của người chơi

void Export(): update highscore và đưa lại vào file

void close(): giải phóng toàn bộ các bộ nhớ được cấp phát cho game.

int main(): hàm vận hành game:

* Gọi hàm khởi tạo của màn hình, ảnh, menu, text, Tile map, của nhân vật chơi.
* Kiểm tra khởi tạo thành công
* Kiểm tra lựa chọn của menu và đưa ra theo click chuột
* Load nhạc chương trình, bắt đầu thả rơi vật, cho phép điều kiển nhân vật
* Liên tục kiểm tra sự va chạm giữa vật và nhân vật, đồng thời khoá không cho nhân vật di chuyển khỏi màn hình game
* Cộng điểm nếu người chơi hứng được vật và trừ mạng nếu người chơi làm rớt vật (kèm âm thành)
* Nếu người chơi để rớt quá 3 lần sẽ đưa ra số điểm của người chơi và kiểm tra người chơi có yêu cầu chơi lại không. Nếu có reset toàn bộ chỉ số và cho người chơi chơi lại
* Mỗi lần ghi nhận điểm liên tục update highScore
* Giải phóng hết bộ nhớ và kết thúc chương trình.