

# BÁO CÁO

## KIỂM THỬ HỆ THỐNG

MSSV:23021665

Họ tên: Nguyễn Thành Phước

Lớp: 2526I\_INT3111\_1

Link Github: [Game LTNC SDL2](#)

### Mục lục

1.	Đặc tả.....	3
1.1.	Mô tả dự án.....	3
1.1.1.	Tên dự án: “A Day As A UET Student” .....	3
1.1.2.	Giới thiệu: .....	3
1.1.3.	Chức năng chính: .....	3
1.1.4.	Công nghệ sử dụng:.....	3
1.2.	Bảng thuật ngữ .....	3
1.3.	Đặc tả Use Case.....	4
1.3.1.	UC01:Khởi động game .....	4
1.3.2.	UC02:Xem menu .....	4
1.3.3.	UC03a:Xem high score .....	5
1.3.4.	UC03b:Xem tutorial.....	5
1.3.5.	UC04:Chơi game.....	6
1.3.6.	UC05:Thu thập gift .....	7
1.3.7.	UC06:Tránh fail .....	7
1.3.8.	UC07:Vượt deadline .....	8
1.3.9.	UC08:Xem kết quả.....	8
1.3.10.	UC09:Thoát game .....	9
2.	Unit Testing.....	10
2.1.	Các hàm cần test.....	10
2.2.	Unit Test.....	10
3.	Kiểm thử tích hợp.....	13
3.1.	Tích hợp hệ thống khởi tạo SDL .....	13
3.2.	Tích hợp LoadTexturevà SDL_render .....	13
3.3.	Tích hợp Character và SDL_event.....	14
3.4.	Tích hợp Collision System .....	14
3.5.	Tích hợp Game State Management .....	15
3.6.	Tích hợp Audio và Game Event.....	17
3.7.	Tích hợp Spawn System.....	18
3.8.	Thống kê kiểm thử tích hợp .....	19
4.	Kiểm thử chức năng hệ thống .....	19

4.1.	Các Use Case của hệ thống .....	19
4.2.	Kết quả kiểm thử chức năng hệ thống.....	19
4.2.1.	Khởi động game (UC01).....	19
4.2.2.	Xem menu(UC02).....	20
4.2.3.	Xem high score(UC03a) .....	22
4.2.4.	Xem tutorial(UC03b) .....	22
4.2.5.	Chơi game(UC04).....	23
4.2.6.	Thu thập gift(UC05).....	25
4.2.7.	Tránh fail(UC06).....	26
4.2.8.	Vượt deadline(UC07).....	26
4.2.9.	Xem kết quả(UC08) .....	27
4.2.10.	Thoát game(UC09).....	29
4.2.11.	Thống kê kiểm thử chức năng hệ thống .....	29
5.	<i>Tỷ lệ lỗi</i> .....	29

# 1. Đặc tả

## 1.1. Mô tả dự án

1.1.1. Tên dự án: “A Day As A UET Student”

1.1.2. Giới thiệu:

- Dự án phát triển game, project môn lập trình nâng cao năm 1
- Nơi người chơi điều khiển nhân vật để ăn điểm A+, tránh điểm F và Deadline
- Điểm cao nhất sẽ được lưu vào hệ thống như kỷ lục mới

1.1.3. Chức năng chính:

- Hệ thống menu với các lựa chọn
- Điều khiển nhân vật di chuyển
- Nhân vật thu thập A+ và tránh F, Deadline
- Hệ thống tính điểm và lưu điểm cao nhất
- Tăng độ khó theo cấp độ điểm số
- Hiện thị kết quả game over với tùy chọn chơi lại

1.1.4. Công nghệ sử dụng:

- Ngôn ngữ C++
- Thư viện đồ họa SDL\_2
- Thư viện hỗ trợ SDL\_2 image, mixer và ttf

## 1.2. Bảng thuật ngữ

Thuật ngữ	Định nghĩa	Giải thích ngữ cảnh
Character	Nhân vật	Nhân vật người chơi điều khiển
Gift (A+)	Món quà	Vật phẩm rơi xuống, đại diện cho điểm A+, khi thu thập sẽ được cộng điểm
Fail(F)	Điểm thất bại	Vật phẩm rơi xuống, đại diện cho điểm F, va chạm thì sẽ thua
Deadline	Hạn chót nộp bài	Vật phẩm di chuyển, đại diện cho, va chạm thì sẽ thua
Score	Điểm số	Điểm tích lũy trong game hiện tại
High Score	Điểm cao nhất	Điểm tích lũy trong game cao nhất từng đạt được
Level	Cấp độ	Độ khó của game( có 3 mức 1-3)
Frame	Khung hình	Đơn vị thời gian trong animation của nhân vật
Collision	Va chạm	Sự kiện nhân vật chạm vào 1 vật phẩm hay 1 chướng ngại vật
Veclocity(Vel)	Vận tốc	Tốc độ di chuyển hoặc vật thể

Render	Vẽ/Hiển thị	Quá trình hiển thị hình ảnh lên màn hình
Texture	Kết cấu hình ảnh	Dữ liệu hình ảnh được tải vào bộ nhớ GPU
SDL	Simple Direct Media Layer	Thư viện đa phương tiện đa nền tảng cho game 2D
TTF	TrueType Font	Định dạng phông chữ
Mix Music	Nhạc nền	Chạy nhạc nền dài
Mix Chunk	Âm thanh ngắn	Hiệu ứng âm thanh
gRenderer	Global Renderer	Đối tượng toàn cục để render đồ họa
gWindow	Global Window	Cửa sổ chính của ứng dụng

### 1.3. Đặc tả Use Case

#### 1.3.1. UC01: Khởi động game

Thành phần	Nội dung
Tên Use Case	Khởi động game
Actor	Người chơi
Mô tả	Khởi động ứng dụng game và load tài nguyên
Tiền điều kiện	Đã cài đặt SDL2 và các thư viện liên quan Tất cả các file tài nguyên đã tồn tại
Luồng chính	<ol style="list-style-type: none"> <li>1. Người chơi khởi động file thực thi</li> <li>2. Hệ thống khởi tạo SDL, SDL_image, SDL_mixer, SDL_ttf</li> <li>3. Hệ thống tạo window và renderer</li> <li>4. Hệ thống load tất cả textures</li> <li>5. Hệ thống load tất cả fonts</li> <li>6. Hệ thống load tất cả âm thanh</li> <li>7. Hệ thống đọc High Score từ file MaxScore.txt</li> <li>8. Hệ thống chuyển sang UC02: Xem Menu</li> </ol>
Luồng thay thế	<ol style="list-style-type: none"> <li>3a. Lỗi khởi tạo SDL: <ol style="list-style-type: none"> <li>1. Hiển thị thông báo lỗi ra console</li> <li>2. Thoát chương trình</li> </ol> </li> <li>4a. Lỗi load texture: <ol style="list-style-type: none"> <li>1. Hiển thị thông báo lỗi cụ thể</li> <li>2. Thoát chương trình</li> </ol> </li> <li>7a. File MaxScore.txt không tồn tại: <ol style="list-style-type: none"> <li>1. Khởi tạo highscore = 0</li> </ol> </li> </ol>
Hậu điều kiện	Hiển thị Menu

#### 1.3.2. UC02: Xem menu

Thành phần	Nội dung
Tên Use Case	Xem menu
Actor	Người chơi
Mô tả	Hiển thị menu chính với các lựa chọn
Tiền điều kiện	Hệ thống đã khởi tạo thành công

Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị background menu</li> <li>2. Hệ thống hiển thị tiêu đề "A Day As A UET Student!"</li> <li>3. Hệ thống phát nhạc nền menu (loop)</li> <li>4. Hệ thống hiển thị các nút: <ul style="list-style-type: none"> <li>- Start Game (hoặc Continue nếu đã chơi)</li> <li>- High Score</li> <li>- Tutorial</li> </ul> </li> <li>5. Người chơi click chuột vào một trong các nút</li> <li>6. Hệ thống phát âm thanh Choice.mp3</li> <li>7. Hệ thống chuyển đến màn hình tương ứng</li> </ol>
Luồng thay thế	<ol style="list-style-type: none"> <li>5a. Click Start/Continue: Include UC04: Chơi Game</li> <li>5b. Click High Score: Include UC03a: Xem High Score</li> <li>5c. Click Tutorial: Include UC03b: Xem Tutorial</li> <li>5d. Click nút X (đóng cửa sổ): Include UC09: Thoát Game</li> </ol>
Hậu điều kiện	Chuyển sang màn hình được chọn

#### 1.3.3. UC03a:Xem high score

Thành phần	Nội dung
Tên Use Case	Xem High Score
Actor	Người chơi
Mô tả	Hiển thị điểm cao nhất đạt được
Tiền điều kiện	Đang ở màn hình menu
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị background menu</li> <li>2. Hệ thống hiển thị High Score với font lớn (GameOver font, 50pt)</li> <li>3. Hệ thống hiển thị text "Try your best to break this record!"</li> <li>4. Hệ thống hiển thị nút Back</li> <li>5. Người chơi click Back</li> <li>6. Hệ thống phát âm thanh Choice.mp3</li> <li>7. Quay lại UC02: Xem Menu</li> </ol>
Luồng thay thế	---
Hậu điều kiện	Quay lại menu chính

#### 1.3.4. UC03b:Xem tutorial

Thành phần	Nội dung
Tên Use Case	Xem Tutorial
Actor	Người chơi
Mô tả	Hiển thị hướng dẫn cách chơi
Tiền điều kiện	Đang ở màn hình menu
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị background menu</li> <li>2. Hệ thống hiển thị hình ảnh</li> </ol>

	Tutorial text.png (hướng dẫn chi tiết) 3. Hệ thống hiển thị hình ảnh UETER.png ở góc dưới trái 4. Hệ thống hiển thị nút Back 5. Người chơi đọc hướng dẫn 6. Người chơi click Back 7. Hệ thống phát âm thanh Choice.mp3 8. Quay lại UC02: Xem Menu
Luồng thay thế	---
Hậu điều kiện	Quay lại menu chính

#### 1.3.5. UC04:Chơi game

Thành phần	Nội dung
Tên Use Case	Chơi game
Actor	Người chơi
Mô tả	Người chơi điều khiển nhân vật chơi game
Tiền điều kiện	Đã chọn Start/Continue từ menu
Luồng chính	1. Hệ thống hiển thị background gameplay (MoonUET) 2. Hệ thống hiển thị nhân vật ở vị trí ban đầu (giữa màn hình, đáy) 3. Hệ thống hiển thị text "Press Space To Play" 4. Hệ thống hiển thị Score: 0, High Score, "Press M: Menu" 5. Người chơi nhấn Space 6. Hệ thống bắt đầu phát nhạc nền gameplay (loop) 7. Hệ thống bắt đầu spawn Gift, Fail, Deadline theo timer 8. Người chơi điều khiển nhân vật: - Left Arrow: Di chuyển trái - Right Arrow: Di chuyển phải - Up Arrow: Nhảy 9. Include UC05: Thu thập Gift 10. Include UC06: Tránh Fail 11. Include UC07: Vượt Deadline 12. Hệ thống cập nhật điểm số liên tục 13. Hệ thống kiểm tra và tăng level khi đủ điểm 14. Lặp lại bước 8-13 cho đến khi Game Over
Luồng thay thế	5a. Nhấn M trước hoặc trong khi chơi: 1. Hệ thống tạm dừng gameplay 2. Include UC02: Xem Menu 3. Người chọn Continue: Resume gameplay 14a. Xảy ra Game Over: Include UC08: Xem kết quả

Hậu điều kiện	<ul style="list-style-type: none"> <li>- Game đang chạy</li> <li>- Hoặc chuyển sang Game Over</li> <li>- Hoặc quay về Menu</li> </ul>
---------------	---

#### 1.3.6. UC05:Thu thập gift

Thành phần	Nội dung
Tên Use Case	Thu thập Gift (A+)
Actor	Người chơi
Mô tả	Nhân vật va chạm với Gift để tăng điểm
Tiền điều kiện	Đang trong gameplay, Gift đã spawn
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống spawn Gift tại vị trí random</li> <li>2. Gift rơi xuống với tốc độ theo level</li> <li>3. Nhân vật di chuyển đến vị trí Gift</li> <li>4. Hệ thống phát hiện va chạm (AABB collision)</li> <li>5. Hệ thống tăng Score +1</li> <li>6. Hệ thống cập nhật High Score nếu Score &gt; High Score</li> <li>7. Hệ thống phát âm thanh <u>Ting_CharacterStone.mp3</u></li> <li>8. Hệ thống xóa Gift khỏi danh sách</li> <li>9. Hệ thống cập nhật hiển thị Score</li> </ol>
Luồng thay thế	3a. Gift rơi chạm đáy màn hình: <ol style="list-style-type: none"> <li>1. Hệ thống xóa Gift</li> <li>2. Không tăng điểm</li> </ol>
Hậu điều kiện	Score tăng, Gift biến mất

#### 1.3.7. UC06:Tránh fail

Thành phần	Nội dung
Tên Use Case	Tránh Fail
Actor	Người chơi
Mô tả	Nhân vật tránh va chạm với Fail
Tiền điều kiện	Đang trong gameplay, Fail đã spawn
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống spawn Fail tại vị trí random (không trùng Gift)</li> <li>2. Fail rơi xuống với tốc độ nhanh hơn Gift</li> <li>3. Người chơi điều khiển nhân vật tránh xa Fail</li> <li>4. Fail rơi chạm đáy màn hình</li> <li>5. Hệ thống xóa Fail khỏi danh sách</li> </ol>
Luồng thay thế	4a. Nhân vật va chạm Fail: <ol style="list-style-type: none"> <li>1. Hệ thống set youlose = true</li> <li>2. Include UC08: Xem kết quả (Game Over)</li> </ol>
Hậu điều kiện	<ul style="list-style-type: none"> <li>- Fail biến mất an toàn</li> <li>- Hoặc Game Over</li> </ul>

### 1.3.8. UC07: Vượt deadline

Thành phần	Nội dung
Tên Use Case	Vượt qua Deadline
Actor	Người chơi
Mô tả	Nhân vật nhảy lên hoặc tránh qua Deadline
Tiền điều kiện	Đang trong gameplay, Deadline đã spawn
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống spawn Deadline ở bên phải màn hình (<math>X=1200-\text{width}</math>, <math>Y=725</math>)</li> <li>2. Deadline di chuyển sang trái với tốc độ theo level</li> <li>3. Người chơi thấy Deadline đang tiến lại</li> <li>4. Người chơi nhấn Up để nhảy</li> <li>5. Nhân vật nhảy lên và rơi xuống (<math>\text{vely} &gt; 0</math>)</li> <li>6. Hệ thống phát hiện va chạm khi nhân vật đang rơi</li> <li>7. Hệ thống tăng Score +1</li> <li>8. Hệ thống phát âm thanh <code>Ting_CharacterStone.mp3</code></li> <li>9. Hệ thống xóa Deadline</li> </ol>
Luồng thay thế	<ol style="list-style-type: none"> <li>6a. Va chạm khi nhân vật không rơi xuống: <ol style="list-style-type: none"> <li>1. Set <code>youlose = true</code></li> <li>2. Include UC08: Game Over</li> </ol> </li> <li>6b. Nhân vật vượt qua Deadline (<math>x &gt; \text{deadline\_x} + \text{width}</math>): <ol style="list-style-type: none"> <li>1. Tăng Score +1</li> <li>2. Phát âm thanh</li> <li>3. Xóa Deadline</li> </ol> </li> <li>3a. Deadline di chuyển ra khỏi màn hình trái: <ol style="list-style-type: none"> <li>1. Xóa Deadline (không tăng điểm)</li> </ol> </li> </ol>
Hậu điều kiện	<ul style="list-style-type: none"> <li>- Score tăng, Deadline biến mất</li> <li>- Hoặc Game Over</li> </ul>

### 1.3.9. UC08: Xem kết quả

Thành phần	Nội dung
Tên Use Case	Xem kết quả Game Over
Actor	Người chơi
Mô tả	Hiển thị kết quả khi game kết thúc
Tiền điều kiện	<code>youlose = true</code> (va chạm Fail hoặc Deadline sai cách)
Luồng chính	<ol style="list-style-type: none"> <li>1. Hệ thống dừng nhạc nền gameplay</li> <li>2. Hệ thống kiểm tra: <math>\text{Score} \geq \text{High Score}</math>?</li> <li>3a. Nếu YES: <ul style="list-style-type: none"> <li>- Phát âm thanh <code>Wingame.mp3</code></li> <li>- Hiển thị icon <code>Record.png</code></li> </ul> </li> <li>3b. Nếu NO: <ul style="list-style-type: none"> <li>- Phát âm thanh <code>LoseGame.mp3</code></li> <li>- Hiển thị icon <code>YOUFAIL.png</code></li> </ul> </li> </ol>



	4. Hệ thống hiển thị background GameOver.jpg 5. Hệ thống hiển thị bảng kết quả ShowResultBang.jpeg 6. Hệ thống hiển thị Score (màu đỏ, font lớn) 7. Hệ thống hiển thị Max Score (màu đỏ) 8. Hệ thống hiển thị "Press 'P' to replay" 9. Hệ thống lưu High Score vào file MaxScore.txt 10. Người chơi chọn hành động
Luồng thay thế	10a. Nhấn P: 1. Reset score = 0 2. Reset space = false 3. Clear tất cả objects (gifts, fails, deadlines) 4. Set youlose = true (để trigger lại gameplay) 5. Include UC04: Chơi Game 10b. Click X (đóng cửa sổ): 1. Set quit = true 2. Thoát vòng lặp Game Over 3. UC09: Thoát Game
Hậu điều kiện	- Chơi lại game mới - Hoặc thoát game

#### 1.3.10. UC09:Thoát game

Thành phần	Nội dung
Tên Use Case	Thoát Game
Actor	Người chơi
Mô tả	Giải phóng tài nguyên và đóng ứng dụng
Tiền điều kiện	Game đang chạy
Luồng chính	1. Người chơi click nút X hoặc thoát từ menu 2. Hệ thống dừng tất cả âm thanh 3. Hệ thống giải phóng tất cả Mix_Music và Mix_Chunk 4. Hệ thống đóng SDL_mixer 5. Hệ thống đóng tất cả TTF_Font 6. Hệ thống giải phóng tất cả SDL_Texture 7. Hệ thống destroy SDL_Renderer 8. Hệ thống destroy SDL_Window 9. Hệ thống quit IMG, TTF, SDL
Luồng thay thế	---
Hậu điều kiện	Kết thúc game

## 2. Unit Testing

### 2.1. Các hàm cần test

Nhóm	Hàm cần kiểm tra
Logic va chạm	Bool checkCollision(SDL_Rect a, SDL_Rect b)
Logic nhân vật	Void Character::move(); Void Character::Solve_Event(SDL_Event &e); void Character::reset();
Logic Game Objects	void AddGift(); void UpdateGift(); void AddFailScore(); void UpdateFailScore(); void AddDeadline(); void UpdateDeadline();
Logic Coliision với Vật	void Check_Character_Gift_Collision(Character& character); void Check_Character_FailScore_Collision(Character& character); void CheckCharacter_DeadlineCollision(Character& character);
Logic Điểm và Level	void Change_level(); void Save_High_Score(string path); void Load_High_Score(string path);

### 2.2. Unit Test

Hàm cần kiểm tra	Testcase	Expected Output
Bool checkCollision(SDL_Rect a, SDL_Rect b)	<ul style="list-style-type: none"> <li>- {0, 0, 50, 50}, {100, 100, 50, 50}</li> <li>- {0, 0, 100, 100}, {25, 25, 50, 50}</li> <li>- {0, 0, 60, 60}, {40, 40, 60, 60}</li> <li>- {0, 0, 50, 50}, {50, 0, 50, 50}</li> <li>- {0, 100, 50, 50}, {0, 0, 50, 50}</li> <li>- {0, 0, 50, 50}, {0, 100, 50, 50}</li> <li>- {0, 0, 50, 50}, {100, 0, 50, 50}</li> <li>- {100, 0, 50, 50}, {0, 0, 50, 50}</li> <li>- {50, 50, 100, 100}, {50, 50, 100, 100}</li> <li>- {0, 0, 60, 60}, {50, 50, 60, 60}</li> <li>- {0, 0, 0, 50}, {0, 0, 50, 50}</li> <li>- {0, 0, 50, 0}, {0, 0, 50, 50}</li> <li>- {-50, -50, 100, 100}, {0, 0, 50, 50}</li> <li>- {0, 0, 50, 50}, {49, 49, 50, 50}</li> <li>- {0, 0, 50, 50}, {50, 50, 50, 50}</li> <li>- {0, 0, 10000, 10000}, {5000, 5000, 1000, 1000}</li> </ul>	<ul style="list-style-type: none"> <li>- False</li> <li>- True</li> <li>- True</li> <li>- False</li> <li>- False</li> <li>- False</li> <li>- False</li> <li>- False</li> <li>- False</li> <li>- True</li> <li>- True</li> <li>- False</li> <li>- False</li> <li>- True</li> <li>- True</li> <li>- False</li> <li>- True</li> </ul>
Void Character::move();	<ul style="list-style-type: none"> <li>- x_Vel=-20, gọi move()</li> <li>- x_Vel=20, gọi move()</li> <li>- up_pressed=true, gọi move()</li> <li>- jump=true, y_Vel=-10, gọi move()</li> <li>- jump=true, y_Vel=10, gọi move()</li> <li>- x_Vel=-40, gọi move()</li> <li>- x_Vel=0, gọi move()</li> <li>- x_Vel=-1000, gọi move()</li> <li>- x_Vel=1000, gọi move()</li> <li>- jump=true, y_Vel=-10, gọi move() 5 lần</li> <li>- Khởi tạo Character mới</li> </ul>	<ul style="list-style-type: none"> <li>- x_position giảm 20</li> <li>- x_position tăng 20</li> <li>- jump=true, y_Vel=-16</li> <li>- y_Vel=-9</li> <li>-</li> <li>y_position=SCREEN_HEIGHT-75, jump=false, y_Vel=0</li> <li>- x_position giảm 40</li> <li>- x_position không đổi</li> <li>- x_position &gt;= 0</li> <li>- x_position+width &lt;= SCREEN_WIDTH</li> </ul>

		- y_Vel=-5 - y_position=SCREEN_HEIGHT-75, jump=false
Void Character::Solve_Event(SDL_Event &e);	- Event KEYDOWN SDLK_LEFT - Event KEYDOWN SDLK_RIGHT - Event KEYDOWN SDLK_UP (not jumping) - Event KEYUP SDLK_LEFT - Event KEYUP SDLK_RIGHT - Event KEYUP SDLK_UP	- isLeft=true, x_Vel=-Vmax_Character - isLeft=false, x_Vel+=Vmax_Character - jump=true, up_pressed=true - x_Vel+=Vmax_Character - x_Vel=-Vmax_Character - up_pressed=false
void Character::reset();	Gọi reset() sau khi Character di chuyển (x_Vel=20)	x_position=SCREEN_WIDTH/2, y_position=SCREEN_HEIGHT-75, x_Vel=0, y_Vel=0, jump=false, up_pressed=false
void AddGift();	Gọi AddGift()	gifts.size()=1, y_position=0, x_position=random(0 to SCREEN_WIDTH-GIFT_WIDTH), y_Vel=random(3 to 7)
void UpdateGift();	- gifts={100, 100, 5}, gọi UpdateGift() - gifts={100, SCREEN_HEIGHT+10, 5}, gọi UpdateGift() - gifts={100, 100, 0}, gọi UpdateGift() - gifts={100, 100, -5}, gọi UpdateGift() - gifts có 3 items (2 valid, 1 off-screen), gọi UpdateGift() - gifts={100, SCREEN_HEIGHT-100, 500}, gọi UpdateGift()	- y_position=105 - gifts.size()=0 - y_position=100 - y_position=95 - gifts.size()=2 - gifts.size()=0
void AddFailScore();	Gọi AddFailScore()	Fscores.size()=1, y_position=0, x_position=random(0 to SCREEN_WIDTH-FAIL_WIDTH), y_Vel=random(5 to 9)
void UpdateFailScore();	- Fscores={200, 150, 7}, gọi UpdateFailScore() - Fscores={200, SCREEN_HEIGHT+20, 7}, gọi UpdateFailScore() - Fscores={100, SCREEN_HEIGHT, 5}, gọi UpdateFailScore() - Fscores={100, SCREEN_HEIGHT-6, 5}, gọi UpdateFailScore()	- y_position=157 - Fscores.size()=0 - Fscores.size()=0 - Fscores.size()=1, y_position=SCREEN_HEIGHT-1
void AddDeadline();	Gọi AddDeadline()	deadlines.size()=1, x_position=SCREEN_WIDTH-DEADLINE_WIDTH, y_position=SCREEN_HEIGHT-DEADLINE_HEIGHT, x_Vel=2
void UpdateDeadline();	- deadlines={500, SCREEN_HEIGHT-DEADLINE_HEIGHT, 4}, gọi UpdateDeadline() - deadlines={-10, SCREEN_HEIGHT-DEADLINE_HEIGHT, 2}, gọi	- x_position=496 - deadlines.size()=0 - deadlines.size()=0 - deadlines.size()=1, x_position=0

	UpdateDeadline() - deadlines={0, SCREEN_HEIGHT-DEADLINE_HEIGHT, 2}, gọi UpdateDeadline() - deadlines={1, SCREEN_HEIGHT-DEADLINE_HEIGHT, 1}, gọi UpdateDeadline()	
void Check_Character_Gift_Collision(Character& character);	- Character tại vị trí gift, gọi hàm - Character cách xa gift (500,500), gọi hàm - Character tại screen edge, gift gần đó, gọi hàm - gifts.clear(), gọi hàm - Character với 2 gifts va chạm + 1 gift (500,500), gọi hàm	- score++, gifts.size()==0 - score=0, gifts.size()==1 - score++ - score=0, gifts.size()==0 - score=2, gifts.size()==1
void Check_Character_FailScore_Collision(Character& character);	- Character tại vị trí FailScore, gọi hàm - Character cách xa FailScore (500,500), gọi hàm - Fscores={posx+1, posy+1, 5}, gọi hàm	- youlose=true - youlose=false - Kiểm tra collision detection
void CheckCharacter_DeadlineCollision(Character& character);	- Character y_Vel=5 (>0) va chạm deadline, gọi hàm - Character y_Vel=-5 (<0) va chạm deadline, gọi hàm - Character.x > deadline.x+w, gọi hàm - deadline={posx, posy, 0}, x_Vel=0, gọi hàm - deadline={posx, posy, 2}, gọi hàm	- score++, deadlines.size()==0 - youlose=true - score++ - youlose=true - youlose=true
void Change_level();	- score=7 - score=14 - score=31 - score=5 - score=6 - score=13 - score=21 - score=30 - score=0 - score=100	- level=2 - level=2 - level=3 - level=1 - level=1 - level=2 - level=2 - level=2 - level=1 - level=3
void Save_High_Score(string path);	- highscore=100, Save("test_score.txt") - highscore=0, Save("test_zero_score.txt") - highscore=999999, Save("test_large_score.txt")	- File chứa "100" - File chứa "0" - File chứa "999999"
void Load_High_Score(string path);	- File chứa "250", Load("test_score_load.txt") - Save(500) rồi Load("test_cycle.txt") - File empty "", Load("test_empty_score.txt")	- highscore=250 - highscore=500 - highscore không đổi

### 3. Kiểm thử tích hợp

#### 3.1. Tích hợp hệ thống khởi tạo SDL

Tên	SDL Core với SDL_image
Mô tả	Kiểm tra khả năng tải hình ảnh PNG sau khi khởi tạo SDL
Tiền điều kiện	SDL_Init được khởi tạo thành công IMG_Init được khởi tạo thành công
Các bước thực hiện	Khởi tạo game
Kết quả mong đợi	Texture được load thành công Không báo lỗi trong console
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp SDL Core với SDL_ttf
Mô tả	Kiểm tra khả năng load fonts sau khi khởi tạo SDL
Tiền điều kiện	SDL_Init được khởi tạo thành công TTF_Init được khởi tạo thành công
Các bước thực hiện	Khởi tạo game
Kết quả mong đợi	Tất cả các font được load thành công Không có thông báo lỗi trong console
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp SDL Core với SDL_mixer
Mô tả	Kiểm tra khả năng phát âm thanh sau khi khởi tạo SDL
Tiền điều kiện	SDL_Init được khởi tạo thành công initAudio được khởi tạo thành công
Các bước thực hiện	Khởi tạo game
Kết quả mong đợi	Mix_Audio trả về $\geq 0$ Tất cả Mix_Music và Mix_Chunk $\neq$ null Âm thanh phát ra được
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

#### 3.2. Tích hợp LoadTexture và SDL\_render

Tên	Tích hợp LoadTexture::LoadImage() với SDL_Renderer
Mô tả	Kiểm tra việc tải texture và render lên màn hình
Tiền điều kiện	SDL và SDL_image đã được khởi tạo gRenderer $\neq$ Null
Các bước thực hiện	Khởi tạo game, chờ 1s
Kết quả mong đợi	Texture được load thành công Hình ảnh hiển thị đúng trên màn hình Không có lỗi render trong console log
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp LoadTexture với SDL_Rect (clipping)
Mô tả	Kiểm tra render với clipping cho animation
Tiền điều kiện	gCharacter được load thành công Các khung ảnh của Character được load thành công
Các bước thực hiện	Khởi tạo game, vào chơi game và xem vị trí, chuyển động của nhân vật
Kết quả mong đợi	Các frame animation hiển thị đúng Flip hoạt động chính xác Không có lỗi trong console
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.3. Tích hợp Character và SDL\_event

Tên	Tích hợp Character::Solve_Event() với SDL_Event
Mô tả	Kiểm tra xử lý sự kiện bàn phím cho Character
Tiền điều kiện	gCharacter được khởi tạo Game đang chạy
Các bước thực hiện	Khởi tạo Game Chạy game Di chuyển lên sang trái, phải để kiểm tra hướng đi của nhân vật
Kết quả mong đợi	Velocity thay đổi theo đúng phím bấm Jump flag được đánh dấu Không có lỗi xử lý event
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp Character::move() với Game Loop
Mô tả	Kiểm tra cập nhật vị trí Character trong mỗi frame
Tiền điều kiện	Game được khởi tạo Character đã được khởi tạo
Các bước thực hiện	Khởi tạo Game Chạy game Kiểm tra nhân vật move nhiều lần, rơi xuống, dừng lại ở đáy màn hình
Kết quả mong đợi	Vị trí cập nhật theo đúng vel Gravity hoạt động để vel_y tăng lên Nhân vật không rơi đáy màn hình Jump=false khi chạm đất
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.4. Tích hợp Collision System

Tên	Tích hợp checkCollision() với Character và Gift
Mô tả	Kiểm tra phát hiện va chạm giữa Character và Gift
Tiền điều kiện	Character và Gift đã được khởi tạo

	Cả hai đang được render
Các bước thực hiện	Khởi tạo game Chạy Game Di chuyển nhân vật cho chạm A+
Kết quả mong đợi	Va chạm được phát hiện chính xác Gift biến mất Score tăng 1 Âm thanh phát ra
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp checkCollision() với Character và Fail
Mô tả	Kiểm tra game over khi Character chạm Fail
Tiền điều kiện	Character và Fail đã được khởi tạo Cả hai đang được render
Các bước thực hiện	Khởi tạo game Chạy Game Di chuyển nhân vật cho chạm F
Kết quả mong đợi	Va chạm được phát hiện chính xác Game Over được trigger Game Play dừng lại Âm thanh phát ra
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp checkCollision() với Character và Deadline
Mô tả	Kiểm tra logic phức tạp của Deadline collision
Tiền điều kiện	Character và Deadline đã được khởi tạo Cả hai đang được render
Các bước thực hiện	Khởi tạo game Chạy Game Di chuyển nhân vật cho -Chạm deadline -Nhảy lên deadline -Nhảy qua deadline
Kết quả mong đợi	Nhảy đúng cách: score tăng, không thua Va chạm sai: Game Over Vượt qua: Score tăng Âm thanh collision phát ra chính xác
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.5. Tích hợp Game State Management

Tên	Tích hợp Menu với Gameplay
Mô tả	Kiểm tra chuyển đổi giữa Menu và Gameplay
Tiền điều kiện	Game đã khởi động Menu đang hiển thị
Các bước thực hiện	Khởi động game

	Chơi game bằng ấn space Nhấn M trong khi chơi để Menu hiển thị lại Click continue để kiểm tra gameplay resume
Kết quả mong đợi	Chuyển đổi mượt mà giữa Menu và Gameplay Nhạc nền chuyển đúng State giữ nguyên khi pause/resume Không bị crash
Kết quả thực tế	Nhạc phát trước, và màn hình bị đơ mất 0.3->0.5s khi ấn continue, và nhân vật tự động bị chạy theo hướng nút trước khi ấn M để hiện menu
Trạng thái	Fail

Tên	Tích hợp Gameplay với Game Over
Mô tả	Kiểm tra chuyển đổi từ Gamplay sang Game over
Tiền điều kiện	Game đang chạy
Các bước thực hiện	Khởi tạo game Chơi game Cho character va chạm với fail hoặc deadline
Kết quả mong đợi	Game over trigger đúng Điểm hiển thị chính xác High score được lưu (nếu đó là kỉ lục)
Kết quả thực tế	Màn hình bị lỏng vào nhau, vị trí không chính xác trên những màn hình có kích thước khác nhau
Trạng thái	Fail

Tên	Tích hợp Game State với Level System
Mô tả	Kiểm tra thay đổi level khi đạt điểm
Tiền điều kiện	Đang chơi game Score<7
Các bước thực hiện	Chơi đến score=7 Kiểm tra màn hình có hiện Speed up! Chơi đến score=14 Kiểm tra tốc độ có tăng lên Chơi đến score=21 Kiểm tra tốc độ có tăng thêm Chơi đến score =31 Kiểm tra level==3
Kết quả mong đợi	Level tăng đúng thời điểm Tốc độ objects tăng theo level Text Speed Up! Hiện thị rõ ràng Gameplay khó hơn
Kết quả thực tế	Thoả mãn
Trạng thái	Pass



### Tích hợp File I/O và Game State

Tên	Tích hợp Load_High_Score() với init()
Mô tả	Kiểm tra load high score khi khởi động game
Tiền điều kiện	File MaxScore.txt tồn tại với nội dung "15"
Các bước thực hiện	Khởi tạo game Vào phần High Score trong Menu Kiểm tra xem kết quả hiển thị == 15
Kết quả mong đợi	High score được đọc đúng từ file Hiển thị đúng trong menu
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp Save_High_Score() với Game Over
Mô tả	Kiểm tra lưu high score khi game over
Tiền điều kiện	High Score hiện tại =10 Đang chơi game
Các bước thực hiện	Đang chơi game đạt score >10 Trigger Game over Kiểm tra high score được cập nhật Kiểm tra file Save_High_Score được gọi, cập nhật
Kết quả mong đợi	High score được cập nhật đúng File được ghi đúng
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp File I/O với trường hợp file không tồn tại
Mô tả	Kiểm thử xử lý khi file không tồn tại
Tiền điều kiện	Xoá file Max_Score.txt
Các bước thực hiện	Khởi động game Chơi game đạt score=5 Trigger game over Kiểm tra thông tin hiển thị
Kết quả mong đợi	Không crash khi không tồn tại High score mặc định =0 File được tạo mới khi lưu
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.6. Tích hợp Audio và Game Event

Tên	Tích hợp Audio với Collision Events
Mô tả	Kiểm tra âm thanh phát khi có va chạm
Tiền điều kiện	Audio system đã khởi tạo
Các bước thực hiện	Khởi tạo Game Chơi game Thu thập gift Nhảy lên deadline, vượt qua deadline

	Kiểm tra âm thanh collision phát ra
Kết quả mong đợi	Âm thanh phát đúng thời điểm Không bị delay Không bị overlap lỗi
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp Audio với State Transitions
Mô tả	Kiểm tra nhạc nền khi đổi state
Tiền điều kiện	Audio system đã được khởi tạo
Các bước thực hiện	Vào Menu-Kiểm tra Background_menu Vào Start Game-Kiểm tra Menu dừng Ấn Space-Kiểm tra music_background Nhấn M – Kiểm tra gameplay pause Quay về Menu-Kiểm tra nhạc Menu bắt đầu lại Click Continue-Kiểm tra nhạc Gameplay liên tục Game Over(win)- kiểm tra nhạc Wingame.mp3 Game Over (lose)-Kiểm tra LoseGame.mp3
Kết quả mong đợi	-Nhạc nền chuyển đổi đúng theo màn hình -Âm thanh phù hợp trạng thái
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.7. Tích hợp Spawn System

Tên	Tích hợp Timer với AddGift()
Mô tả	Kiểm tra Gift spawn đúng thời gian
Tiền điều kiện	Đang chơi Nhấn Space để chơi game
Các bước thực hiện	Đặt timegift=0 Chạy game loop Đếm số frame khi timeGift=timeframeGift Kiểm tra gifts.size() tăng lên với từng level Đợi thêm 100frames Kiểm tra Gift spawn lại
Kết quả mong đợi	Gift spawn đều đặn mỗi 100 frames Số lượng Gift tương ứng với level Timer reset đúng
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

Tên	Tích hợp AddGift() với AddFailScore() (overlap check)
Mô tả	Kiểm tra Gift và Dail không spawn trùng vị trí
Tiền điều kiện	Đang chơi game

	Đã có gift hoặc fail trong vector
Các bước thực hiện	Chơi game Kiểm tra các deadline, fail, gift có bị spawn trùng nhau hay không.
Kết quả mong đợi	Gift và fail không có spawn trùng nhau Logic overlap hoạt động đúng
Kết quả thực tế	Xuất hiện Gift và Fail hiện chồng đè lên nhau, có cả 2 gift, 2fail đè lên nhau
Trạng thái	Fail

Tên	Tích hợp AddDeadline với timer
Mô tả	Kiểm tra Deadline spawn đúng thời gian
Tiền điều kiện	Đang chơi Nhấn Space để chơi game
Các bước thực hiện	Chơi game Kiểm tra deadline có gen đều không Kiểm tra vị trí spawn deadline có đúng không
Kết quả mong đợi	Deadline spawn không đều(random) Deadline spawn đúng vị trí Timer reset đúng
Kết quả thực tế	Thoả mãn
Trạng thái	Pass

### 3.8. Thống kê kiểm thử tích hợp

Tổng số test	Test Fail	Test Pass
21	3	18

## 4. Kiểm thử chức năng hệ thống

### 4.1. Các Use Case của hệ thống

UC	Tên UC
UC01	Khởi động game
UC02	Xem menu
UC03a	Xem high score
UC03b	Xem tutorial
UC04	Chơi game
UC05	Thu thập gift
UC06	Tránh fail
UC07	Vượt deadline
UC08	Xem kết quả
UC09	Thoát Game

### 4.2. Kết quả kiểm thử chức năng hệ thống

#### 4.2.1. Khởi động game (UC01)

Test Case ID	TC-UC01-01
Tên	Khởi động game chạy thành công
Mục đích	Kiểm tra game khởi động bình thường
Tiền điều kiện	Đã cài đặt SDL2 và các thư viện

	File tài nguyên đầy đủ File MaxScore.txt đã tồn tại
Bước thực hiện	Chạy file thực thi game Quan sát quá trình khởi tạo và màn hình hiển thị
Kết quả mong đợi	Window mở với title 23021665_NguyenThanhPhuoc Hiển thị menu chính Nhạc nền menu phát High Score được load từ file
Kết quả	Pass

Test Case ID	TC-UC01-02
Tên	Khởi động khi thiếu file MaxScore.txt
Mục đích	Kiểm tra xử lý khi file điểm không tồn tại
Tiền điều kiện	Đã cài đặt SDL2 và các thư viện Xoá file MaxScore.txt
Bước thực hiện	Xoá file MaxScore.txt Khởi động game Kiểm tra giá trị highScore
Kết quả mong đợi	Game khởi động thành công highScore =0 console.log hiện: Unable to open file
Kết quả	Pass

Test Case ID	TC-UC01-03
Tên	Khởi động khi thiếu texture
Mục đích	Kiểm tra xử lý lỗi khi thiếu tài nguyên hình ảnh
Tiền điều kiện	Xoá 1 file ảnh bất kỳ (A+,Fail,...)
Bước thực hiện	Xoá file image/A+.png
Kết quả mong đợi	Console.log: hiển thị lỗi cụ thể Chương trình thoát an toàn
Kết quả	Pass

Test Case ID	TC-UC01-01
Tên	Khởi động khi thiếu font
Mục đích	Kiểm tra xử lý lỗi khi thiếu font chữ
Tiền điều kiện	Xoá file font juice.ttf
Bước thực hiện	Xoá file font juice.ttf Khởi động game
Kết quả mong đợi	Console.log: Failed to load font.. Chương trình thoát
Kết quả	Pass

#### 4.2.2. Xem menu(UC02)

Test Case ID	TC-UC02-01
Tên	Hiển thị menu lần đầu
Mục đích	Kiểm tra menu hiển thị đúng khi chưa chơi

Tiền điều kiện	Game mới khởi động Chưa bắt đầu chơi (times=0)
Bước thực hiện	Khởi động game Quan sát Menu
Kết quả mong đợi	Hiển thị nút Start game Hiển thị High Score Hiển thị Tutorial Nhạc menu nền chạy loop Tiêu đề hiện “A Day As A UET Student”
Kết quả	Pass

Test Case ID	TC-UC02-02
Tên	Click nút Start Game
Mục đích	Kiểm tra chuyển sang gameplay
Tiền điều kiện	Đang ở menu chính
Bước thực hiện	Click vào nút Start Game Quan sát màn hình
Kết quả mong đợi	Phát âm thanh Choice.mp3 Chuyển sang gameplay Nhạc menu dừng Hiển thị “Press Space To Play”
Kết quả	Pass

Test Case ID	TC-UC02-03
Tên	Click nút High Score
Mục đích	Kiểm tra chuyển sang màn high score
Tiền điều kiện	Đang ở menu chính
Bước thực hiện	Click vào High Score
Kết quả mong đợi	Phát âm thanh Choice.mp3 Hiển thị high score Hiển thị nút Back
Kết quả	Pass

Test Case ID	TC-UC02-04
Tên	Click nút Tutorial
Mục đích	Kiểm tra hiển thị hướng dẫn
Tiền điều kiện	Đang ở menu chính
Bước thực hiện	Click vào nút Tutorial
Kết quả mong đợi	Phát âm thanh Choice.mp3 Chuyển sang trang Tutorial Nhạc nền Menu
Kết quả	Pass

Test Case ID	TC-UC02-05
Tên	Hiển thị Continue khi đã chơi
Mục đích	Kiểm tra nút thay đổi sau lần đầu chơi
Tiền điều kiện	Đã chơi ít nhất 1 lần
Bước thực hiện	Chơi game 1 lần Ấn M để mở menu

	Quan sát màn hình về menu
Kết quả mong đợi	Hiện thị Continue thay vì Start Game
Kết quả	Pass

Test Case ID	TC-UC02-06
Tên	Click X để đóng cửa sổ
Mục đích	Kiểm tra thoát game từ menu
Tiền điều kiện	Đang ở Menu
Bước thực hiện	Click nút X góc phải Quan sát màn hình
Kết quả mong đợi	Game thoát an toàn Giải phóng tài nguyên Không crash
Kết quả	Pass

#### 4.2.3. Xem high score(UC03a)

Test Case ID	TC-UC03a-01
Tên	Hiện thị high score=0
Mục đích	Kiểm tra hiện thị khi chưa có điểm
Tiền điều kiện	High score=0 Đang ở menu
Bước thực hiện	Click High Score
Kết quả mong đợi	Hiện thị Max Score:0, màu đỏ, font 50px Text “Try your best to break this record” Hiện thị nút back
Kết quả	Pass

Test Case ID	TC-UC03a-02
Tên	Hiện thị gigh score có giá trị
Mục đích	Kiểm tra hiện thị điểm đã lưu
Tiền điều kiện	High Score >0 (đã chơi và lưu điểm)
Bước thực hiện	Chơi game đạt 10 điểm Game over
Kết quả mong đợi	Vào menu sẽ thấy hiện thị High Score
Kết quả	Pass

Test Case ID	TC-UC03a-03
Tên	Click Back từ High Score
Mục đích	Kiểm tra back từ menu
Tiền điều kiện	Đang ở màn hình High Score
Bước thực hiện	Click Back
Kết quả mong đợi	Phát âm thanh choice.mp3 Quay về menu chính
Kết quả	Pass

#### 4.2.4. Xem tutorial(UC03b)

Test Case ID	TC-UC03b-01
--------------	-------------

Tên	Hiện thị tutorial đầy đủ
Mục đích	Kiểm tra tất cả thông tin hướng dẫn
Tiền điều kiện	Đang ở menu
Bước thực hiện	Click Tutorial từ menu
Kết quả mong đợi	Hiện thị Tutorial_text.png ở giữa Hiện thị UETer.png góc dưới trái Nút Back Background Menu
Kết quả	Fail (“Sai chính tả hiện Totutorial”)

Test Case ID	TC-UC03b-02
Tên	Click Back từ Tutorial
Mục đích	Kiểm tra quay về từ Menu
Tiền điều kiện	Đang ở màn Tutorial
Bước thực hiện	Click nút back
Kết quả mong đợi	Phát âm choic.mp3 Quay về menu chính
Kết quả	Pass

#### 4.2.5. Chơi game(UC04)

Test Case ID	TC-UC04-01
Tên	Bắt đầu gameplay
Mục đích	Kiểm tra khởi tạo gameplay
Tiền điều kiện	Click Start/Continue từ menu
Bước thực hiện	Quan sát màn hình gameplay
Kết quả mong đợi	Background MoonUet Nhân vật ở giữa đáy màn hình Hiện Press Space To Play Score:0, High Score hiển thị
Kết quả	Pass

Test Case ID	TC-UC04-02
Tên	Nhấn Space để bắt đầu
Mục đích	Kiểm tra kích hoạt Gameplay
Tiền điều kiện	Đang trong màn “Press Space To Play”
Bước thực hiện	Nhấn Space Quan sát
Kết quả mong đợi	Nhạc nền gameplay bắt đầu phát Gift, Fail, Deadline bắt đầu spawn Text hướng dẫn biến mất
Kết quả	Pass

Test Case ID	TC-UC04-03
Tên	Di chuyển trái
Mục đích	Kiểm tra điều khiển nhân vật sang trái
Tiền điều kiện	Game đang chạy
Bước thực hiện	Nhấn phím left arrow Giữ và thả phím
Kết quả mong đợi	Nhân vật di chuyển trái với Vmax=20

	Animation chạy Flip Horizontal Dừng khi chạm biên trái
Kết quả	Pass

Test Case ID	TC-UC04-04
Tên	Di chuyển phải
Mục đích	Kiểm tra điều khiển nhân vật sang phải
Tiền điều kiện	Game play đang chạy
Bước thực hiện	Nhấn phím right arrow Giữ và thả phím
Kết quả mong đợi	Nhân vật di chuyển với $V_{max}=20$ Animation chạy Flip None Dừng khi chạm biên phải
Kết quả	Pass

Test Case ID	TC-UC04-05
Tên	Nhảy
Mục đích	Kiểm tra cơ chế nhảy
Tiền điều kiện	Nhân vật đang đứng trên mặt đất
Bước thực hiện	Nhấn phím up arrow
Kết quả mong đợi	Nhân vật nhảy lên với $V_{jump}=17$ Vely tăng dần Animation nhảy Rơi xuống và dừng ở đáy
Kết quả	Pass

Test Case ID	TC-UC04-06
Tên	Nhảy liên tục (double jump prevention)
Mục đích	Kiểm tra không nhảy 2 lần trên không
Tiền điều kiện	Nhân vật đang nhảy
Bước thực hiện	Nhấn Up Arrow khi nhân vật đang trên không Quan sát nhân vật
Kết quả mong đợi	Nhân vật không nhảy thêm
Kết quả	Pass

Test Case ID	TC-UC04-07
Tên	Nhấn M để pause và quay về menu
Mục đích	Kiểm tra chức năng pause
Tiền điều kiện	Đang chơi game
Bước thực hiện	Nhấn phím M Quan sát màn hình
Kết quả mong đợi	Hiện thị menu Gameplay tạm dừng Hiện các mục Continue, High Score, Tutorial
Kết quả	Pass



Test Case ID	TC-UC04-08
Tên	Animation nhân vật
Mục đích	Kiểm tra animation của nhân vật
Tiền điều kiện	Gameplay đang chạy
Bước thực hiện	Quan sát nhân vật khi đứng, chạy, nhảy
Kết quả mong đợi	Animation đổi frame mỗi 60ms 8 frames cho mỗi trạng thái Lặp lại
Kết quả	Pass

#### 4.2.6. Thu thập gift(UC05)

Test Case ID	TC-UC05-01
Tên	Spawn Gift
Mục đích	Kiểm tra Gift xuất hiện đúng cách
Tiền điều kiện	Gameplay đang chạy
Bước thực hiện	Chờ 100frames Quan sát
Kết quả mong đợi	Gift spawn ở vị trí random Số lượng=level Không trùng với fail Roi xuống với $vel=rand()\%5 + 3$
Kết quả	Pass

Test Case ID	TC-UC05-02
Tên	Va chạm với gift
Mục đích	Kiểm tra thu thập Gift tăng điểm
Tiền điều kiện	Gift đang rơi Nhân vật di chuyển
Bước thực hiện	Di chuyển nhân vật chạm Gift Quan sát
Kết quả mong đợi	Score +1 High score cập nhật nếu $> HighScore$ Phát âm Ting_Character.mp3 Gift biến mất
Kết quả	Pass

Test Case ID	TC-UC05-03
Tên	Gift rơi chạm đáy không va chạm
Mục đích	Kiểm tra Gift tự xóa khi ra ngoài màn hình
Tiền điều kiện	Gift đang rơi
Bước thực hiện	Không di chuyển nhân vật Để gift rơi xuống đáy
Kết quả mong đợi	Gift bị xóa khỏi vector Không tăng điểm Không có âm thanh
Kết quả	Pass

Test Case ID	TC-UC05-04
--------------	------------

Tên	Thu thập nhiều gift liên tiếp
Mục đích	Kiểm tra xử lý nhiều va chạm
Tiền điều kiện	Có nhiều Gift đang rơi
Bước thực hiện	Di chuyển chạm 3 Gift liên tiếp
Kết quả mong đợi	Score +3 Mỗi Gift phát âm thanh riêng Tất cả Gift biến mất
Kết quả	Pass

#### 4.2.7. Tránh fail(UC06)

Test Case ID	TC-UC06-01
Tên	Spawn Fail
Mục đích	Kiểm tra Fail xuất hiện đúng
Tiền điều kiện	Gameplay đang chạy
Bước thực hiện	Chờ 100 frames Quan sát Fail rơi
Kết quả mong đợi	Fail spawn random Số lượng = level Không trùng Gift Vel nhanh hơn Gift (rand()%5+5)
Kết quả	Pass

Test Case ID	TC-UC06-02
Tên	Va chạm với Fail để Game Over
Mục đích	Kiểm tra thua game khi chạm Fail
Tiền điều kiện	Fail đang rơi
Bước thực hiện	Di chuyển nhân vật chạm fail Quan sát
Kết quả mong đợi	Hiện âm thanh thua game Hiện màn hình game Over Hiện thị số điểm và high score nếu score > high score
Kết quả	Pass

Test Case ID	TC-UC06-03
Tên	Tránh Fail thành công
Mục đích	Kiểm tra Fail tự xóa khi ra màn hình
Tiền điều kiện	Fail đang rơi
Bước thực hiện	Di chuyển tránh xa Fail Để Fail rơi xuống
Kết quả mong đợi	Fail bị xóa Game tiếp tục Không game over
Kết quả	Pass

#### 4.2.8. Vượt deadline(UC07)

Test Case ID	TC-UC07-01
--------------	------------

Tên	Spawn Deadline
Mục đích	Kiểm tra Deadline xuất hiện
Tiền điều kiện	Gameplay đang chạy
Bước thực hiện	Chờ 200 frames (timefamedeadline) Quan sát
Kết quả mong đợi	Deadline spawn bên phải ( $x=1200-width$ ) $Y=725$ Di chuyển sang trái với $vel=2$
Kết quả	Pass

Test Case ID	TC-UC07-02
Tên	Nhảy qua Deadline thành công
Mục đích	Kiểm tra vượt Deadline khi đang rơi
Tiền điều kiện	Deadline di chuyển đến
Bước thực hiện	Nhấn up khi deadline gần Nhân vật nhảy và rơi xuống ( $vely>0$ ) Va chạm khi đang rơi
Kết quả mong đợi	Score tăng 1 Phát âm Tíng_CharacterStone.mp3 Deadline bị xoá Game tiếp tục
Kết quả	Pass

Test Case ID	TC-UC07-03
Tên	Va chạm Deadline khi đang lên
Mục đích	Kiểm tra khi thua khi chạm Deadline sai cách
Tiền điều kiện	Deadline di chuyển
Bước thực hiện	Nhân vật va chạm khi $vely\leq 0$
Kết quả mong đợi	Âm thanh va chạm fail Màn hình hiện game over Hiển thị High Score
Kết quả	Pass

Test Case ID	TC-UC07-04
Tên	Deadline ra khỏi màn hình trái
Mục đích	Kiểm tra Deadline tự xoá
Tiền điều kiện	Deadline đang di chuyển
Bước thực hiện	Không tương tác Để Deadline đi ra ngoài ( $x<0$ )
Kết quả mong đợi	Deadline bị xoá Điểm không tăng
Kết quả	Pass

#### 4.2.9. Xem kết quả(UC08)

Test Case ID	TC-UC08-01
Tên	Game Over với điểm mới cao hơn

Mục đích	Kiểm tra phá kỷ lục
Tiền điều kiện	Youlose= true Score>high score
Bước thực hiện	Chơi game đạt điểm >high score Chạm fail, deadline để game over
Kết quả mong đợi	Nhạc gameplay dừng Phát wingamne.mp3 Hiển thị Icon Record.png Hiển thị Game Over.png Hiển thị bảng showResultBang.jpeg Hiển thị chữ font đỏ Điểm được lưu vào MaxScore
Kết quả	Fail(Màn hình GameOver lỗi hiển thị với kích thước màn hình khác nhau)

Test Case ID	TC-UC08-02
Tên	Game Over với điểm thấp hơn kỷ lục
Mục đích	Kiểm tra không phá kỷ lục
Tiền điều kiện	youlose=true score < highscore
Bước thực hiện	Chơi đạt điểm thấp Thua Game
Kết quả mong đợi	Nhạc gameplay dừng Hiển thị Game Over.png Hiển thị chữ font đỏ
Kết quả	Fail(Màn hình GameOver lỗi hiển thị với kích thước màn hình khác nhau)

Test Case ID	TC-UC08-03
Tên	Nhấn P để replay
Mục đích	Kiểm tra chơi lại
Tiền điều kiện	Đang ở màn Game Over
Bước thực hiện	Nhấn phím P
Kết quả mong đợi	score=0 space=false Màn hình gameplay được hiện ra Tắt âm thanh nền
Kết quả	Pass

Test Case ID	TC-UC08-04
Tên	Hiển thị điểm đúng trên màn Game Over
Mục đích	Kiểm tra độ chính xác dữ liệu
Tiền điều kiện	Chơi đạt 12 điểm, highscore=20
Bước thực hiện	Thua game với score =20 điểm
Kết quả mong đợi	Score: 12, màu đỏ MaxScore: 29, màu đỏ FontGameOver(50px)
Kết quả	Fail(Màn hình GameOver lỗi hiển thị với kích thước màn hình khác nhau)

#### 4.2.10. Thoát game(UC09)

Test Case ID	TC-UC09-01
Tên	Thoát game từ menu
Mục đích	Kiểm tra giải phóng tài nguyên
Tiền điều kiện	Đang ở menu hoặc gameplay
Bước thực hiện	Click nút X Kiểm tra console
Kết quả mong đợi	Dừng tất cả âm thanh Giải phóng tài nguyên (Music,Texture...)
Kết quả	Pass

#### 4.2.11. Thống kê kiểm thử chức năng hệ thống

Tổng số test	Test Fail	Test Pass
39	4	35

## 5. Tỷ lệ lỗi

Game: 1357 dòng

Unit Test	Số lỗi kiểm thử tích hợp	Số lỗi kiểm thử hệ thống	Tổng
5	3	4	12

Tỷ lệ: 12 lỗi/ 1 KLOC