**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**THIẾT KẾ HỆ THỐNG**

***Version 1.1***

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**MỤC LỤC**

[1 Thông tin nhóm 2](#_Toc445498386)

[2 Mô hình quan niệm 3](#_Toc445498387)

[3 Thiết kế kiến trúc 4](#_Toc445498388)

[3.1 Sơ đồ kiến trúc 4](#_Toc445498389)

[3.2 Sơ đồ lớp (Class Diagram) 6](#_Toc445498390)

[3.3 Đặc tả các lớp đối tượng 6](#_Toc445498391)

[3.3.1 Lớp C1 6](#_Toc445498392)

[3.3.2 Lớp C2 7](#_Toc445498393)

[4 Thiết kế dữ liệu 8](#_Toc445498394)

[4.1 Sơ đồ dữ liệu 8](#_Toc445498395)

[4.2 Đặc tả dữ liệu 9](#_Toc445498396)

**TÀI LIỆU THIẾT KẾ HỆ THỐNG**

Tài liệu tập trung vào các chủ đề:

* Tạo ra tài liệu thiết kế hệ thống.
* Hoàn chỉnh tài liệu thiết kế hệ thống với các nội dung:

Hiển thị dữ liệu phức tạp

* DataGridView
  + Mô hình quan niệm
  + Thiết kế kiến trúc
  + Thiết kế dữ liệu
* Đọc hiểu tài liệu thiết kế hệ thống.

# Thông tin nhóm

**Website / Facebook nhóm:** <https://www.facebook.com/>Group

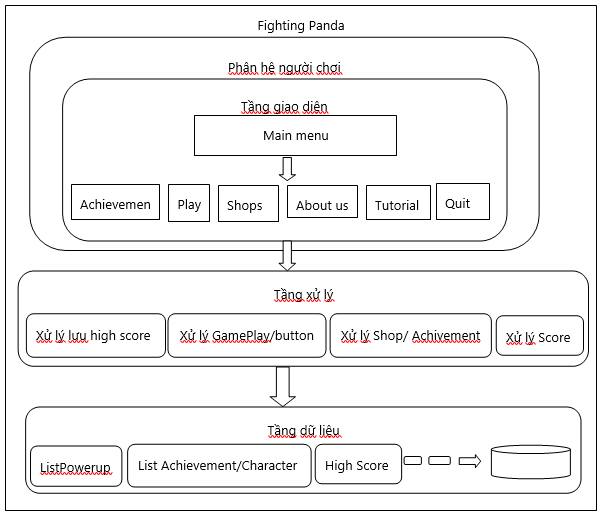
|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **Email** | **Điện thoại** |
| 1412619 | Lâm Tuấn Tú | tulam1503@gmail.com | 01667535638 |
| 1412593 | Đỗ Thanh Trúc |  |  |
| 1412557 | Phạm Đức Toàn |  |  |
| 1412563 | Phan Như Trà |  |  |
| 1412604 | Lữ Hoàng Phi Tuấn |  |  |

# Mô hình quan niệm

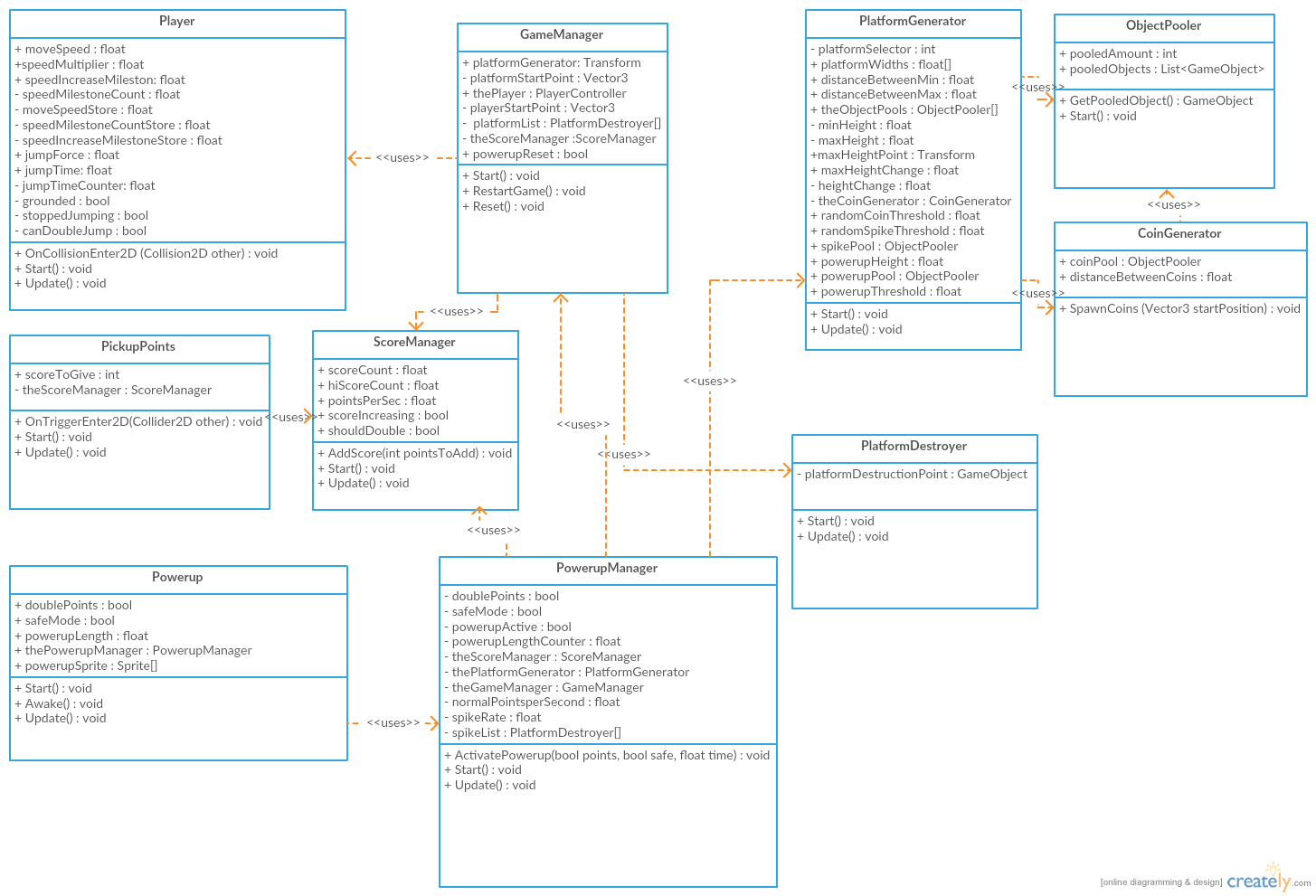
# 

# Thiết kế kiến trúc

## Sơ đồ kiến trúc

**

## Sơ đồ lớp (Class Diagram)

**

## Đặc tả các lớp đối tượng

### Lớp Player

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ moveSpeed : float** | * **Tốc độ di chuyển của nhân vật** |
| **+speedMultiplier : float** | * **Tốc độ di chuyển tăng theo khoảng chạy nhất định** |
| **+ speedIncreaseMileston: float** | * **Mốc chạy, mỗi lần đạt mốc chạy thì tốc độ tăng 1 khoảng \*= speedMultiplier** |
| **- speedMilestoneCount : float** | * **Biến đếm khi nào đạt mốc chạy** |
| **- moveSpeedStore : float**  **- speedMilestoneCountStore : float**  **speedIncreaseMilestoneStore : float** | * **Lưu các giá trị moveSpeed, speedMilestoneCount, speedIncreaseMilestone ban đầu để trả về khi Restart Game** |
| **+ jumpForce : float** | * **Lực nhảy của nhân vật** |
| **+ jumpTime: float** | * **Thời gian cố định cho nhân vật ở trên không, hết thời gian phải hạ xuống đất** |
| **- jumpTimeCounter: float** | * **Biến đếm đến jumpTime** |
| **- grounded : bool** | * **Kiểm tra xem nhân vật đã chạm đất chưa** |
| **- stoppedJumping : bool** | * **Kiểm tra không cho nhảy khi quá jumpTime hoặc đã double Jump** |
| **- canDoubleJump : bool** | * **Kiểm tra nhân vật có được cho thực hiện double Jump hay không** |
| **+ OnCollisionEnter2D (Collision2D other) : void** | * **Kiểm tra va chạm của nhân vật với những thứ gây chết nhân vật (gai, rớt khỏi màn hình)** |
| **+ Start() : void** | * **Tìm Rigidbody2D gắn với class**   **Lưu giá trị cho moveSpeedStore, speedMilestoneCountStore.**  **speedIncreaseMilestoneStore** |
| **+ Update() : void** | * **Tăng tốc độ nếu đạt mốc** * **Kiểm tra phím nhấn người dùng** |

### Lớp ScoreManager

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ scoreCount : float** | * **Tính điểm của game đang chơi** |
| **+ hiScoreCount : float** | * **Giữ giá trị highscore, nếu người chơi đang chơi game vượt highscore thì highscore thay đổi theo score** |
| **+ pointsPerSec : float** | * **Điểm cộng thêm theo thời gian** |
| **+ scoreIncreasing : bool** | * **Cho phép có tăng điểm hay ngừng** |
| **+ shouldDouble : bool** | * **Có nhân đôi điểm hay không (nhận powerup x2 score)** |
| **+ AddScore(int pointsToAdd) : void** | * **Hàm Cộng thêm pointsToAdd vào điểm** |
| **+ Start() : void** | * **Kiểm tra có dữ liệu Highscore nào được lưu chưa, chưa có thì tạo** |
| **+ Update() : void** | * **Update score, highscore trong quá trình chơi** |

### Lớp PickupPoints

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ scoreToGive : int** | * **Điểm nhận được khi ăn tiền** |
| **- theScoreManager : ScoreManager** | * **Gọi để lấy phương thức AddScore(int pointsToAdd) để cộng điểm vào score** |
| **+ OnTriggerEnter2D(Collider2D other) : void** | * **Kiểm tra tiền có va chạm với nhân vật hay không** |

### Lớp PowerupManager

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **- doublePoints : bool** | * **Kiểm tra phải powerup x2 không** |
| **- safeMode : bool** | * **Kiểm tra phải powerup safemode không** |
| **- powerupActive : bool** | * **Kiểm tra có còn kích hoạt powerup hay không, sẽ bị false nếu hết thời gian sử dụng** |
| **- powerupLengthCounter : float** | * **Thời gian sử dụng powerup** |
| **- theScoreManager : ScoreManager** | * **Lấy phương thức AddScore** |
| **- thePlatformGenerator : PlatformGenerator** | * **Để lấy mức độ ra gai từ PlatformGenerator và chỉnh thông số shouldDouble và lấy tỉ lệ ra gai** |
| **- theGameManager : GameManager** | * **Sử dụng để Reset powerup khi restart** |
| **- normalPointsperSecond : float** | * **Lưu để trả về sau khi hết powerup x2** |
| **- spikeRate : float** | * **Mức độ ra gai nhọn, lưu lại để trả về sau khi hết thời gian sử dụng powerup** |
| **- spikeList : PlatformDestroyer[]** | * **List các gai nhọn sẽ bị xóa** |
| **+ ActivatePowerup(bool points, bool safe, float time) : void** | * **Kích hoạt powerup** |
| **+ Update() : void** | * **Kiểm tra nếu là safemode thì** thePlatformGenerator.randomSpikeThreshold = 0; * **Nếu là doublePoints thì** theScoreManager.pointsPerSec = normalPointsperSecond \* 2; theScoreManager.shouldDouble = true; * **Kiểm tra hết thời gian sử dụng powerup chưa** |

### Lớp Powerups

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ doublePoints : bool** | * **Đây có phải là powerup x2 không** |
| **+ safeMode : bool** | * **Đây có phải là powerup safemode không** |
| **+ powerupLength : float** | * **Thời gian sử dụng powerup** |
| **+ thePowerupManager : PowerupManager** | * **Gọi PowerupManager để xử lý powerup** |
| **+ powerupSprite : Sprite[]** | * **List các sprite powerup** |
| **+ Awake() : void** | * **Random powerup** |
| **+ OnTriggerEnter2D (Collider2D other) : void** | * **Kiểm tra va chạm với nhân vật** |

### ObjectPooler

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ pooledAmount : int** | * **Số lượng Object được tạo sẵn** |
| **+ pooledObjects : List<GameObject>** | * **List các object** |
| **+ GetPooledObject() : GameObject** | * **Chọn trong list các object được tạo để đặt vào game, nếu còn thiếu object thì tạo thêm** |
| **+ Start() : void** | * **Khởi tạo số Object theo pooledAmount** |

### CoinGenerator

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ coinPool : ObjectPooler** | * **List coin (ObjectPooler)** |
| **+ distanceBetweenCoins : float** | * **Khoảng cách giữa các coin với nhau** |
| **+ SpawnCoins (Vector3 startPosition) : void** | * **Đặt coin vào game theo vị trí Vector3** |

### PlatformGenerator

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **- platformSelector : int** | * **List coin (ObjectPooler)** |
| **+ platformWidths : float[]** | * **Mảng độ dài các platform** |
| **+ distanceBetweenMin : float**  **+ distanceBetweenMax : float**  **+ distanceBetween : float** | * **Khoảng cách min/max giữa 2 platform** * **Random distanceBetween trong khoảng này** |
| **+ theObjectPools : ObjectPooler[]** | * **List các platform sử dụng** |
| **- minHeight : float**  **- maxHeight : float** | * **Độ dài min/max của platform** |
| **+maxHeightPoint : Transform** | * **Mốc điểm độ cao max** |
| **+ maxHeightChange : float** | * **2 platform kế nhau sẽ có chiều cao chênh lệch max là giá trị này** |
| **- heightChange : float** | * **Độ cao của platform** |
| **- theCoinGenerator : CoinGenerator** | * **Tạo tiền** |
| **+ randomCoinThreshold : float**  **+ randomSpikeThreshold : float**  **+ powerupThreshold : float** | * **Giá trị random của Tiền và Gai và Powerup** |
| **+ spikePool : ObjectPooler**  **+ powerupPool : ObjectPooler** | * **Tạo Gai, Powerup** |
| **+ powerupHeight : float** | * **Độ cao powerup** |
| **+ Start() : void** | * **Lấy minHeight, maxHeight, List platformWidths** |
| **+ Update() : void** | * **Tạo platform, tiền, gai, powerup** |

### PlatformDestroyer

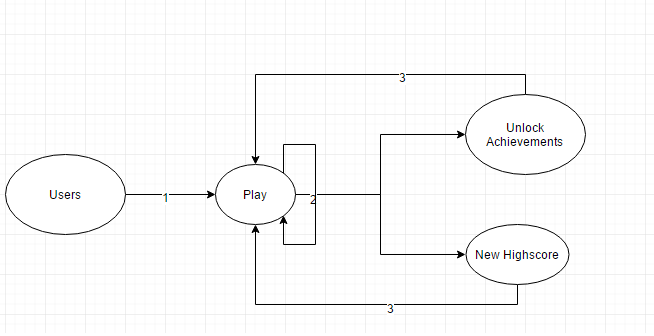
|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **- platformDestructionPoint : GameObject** | * **Mốc inactive object, đặt bên trái màn hình, đi qua mốc này thì inactive** |
| **+ Update() : void** | * **Kiểm tra vị trí đi qua để inactive các object** |

### GameManager

|  |  |
| --- | --- |
| **Tên thuộc tính/phương thức** | **Ý nghĩa** |
| **+ platformGenerator: Transform** | * **Mốc active object, sẽ active cho đến khi đụng mức này** |
| **- platformStartPoint : Vector3** | * **Vị trí bắt đầu platform (min)** |
| **+ thePlayer : PlayerController** | * **Gọi player** |
| **- playerStartPoint : Vector3** | * **Vị trí bắt đầu của player** |
| **- platformList : PlatformDestroyer[]** | * **Gọi PlatformDestroyer** |
| **- theScoreManager :ScoreManager** | * **Gọi ScoreManager** |
| **+ powerupReset : bool** | * **Reset thời gian powerup khi restart game** |
| **+ Start() : void** | * **Lấy giá trị cho platformStartPoint, playerStartPoint** |
| **+ RestartGame() : void** | * **Ngưng tăng điểm, inactive player, bật màn hình khi chết** |
| **+ Reset() : void** | * **Reset lại thông số ban đầu** |

# Thiết kế dữ liệu

## Sơ đồ dữ liệu

**

## Đặc tả dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| *Bảng Users* | | | |
| *STT* | *Tên trường* | *Ràng buộc* | *Mô tả* |
| *1* | *Account ID* | *Khóa chính*  *Chuỗi(15)* | *Mã Account* |
| *2* | *Account Name* | *Chuỗi(100)* | *Tên Account* |
| *3* | *Basic info* | *Chuỗi(200)* | *Thông tin cơ bản* |
| *4* | *Level* | *Int* | *Cấp độ* |

|  |  |  |  |
| --- | --- | --- | --- |
| *Bảng Achievement* | | | |
| *STT* | *Tên trường* | *Ràng buộc* | *Mô tả* |
| *1* | *Account ID* | *Khóa chính*  *Char(15)* |  |
| *2* | *ID* | *Khóa chính*  *Char(15)* |  |
| *3* | *Tên Achievement* | *Char(100)* |  |
| *4* | *Type ID* | *Char(15)* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Bảng Type of Achievement*** | | | |
| *STT* | *Tên trường* | *Ràng buộc* | *Mô tả* |
| *1* | *Type ID* |  |  |
| *2* | *Type Name* |  |  |