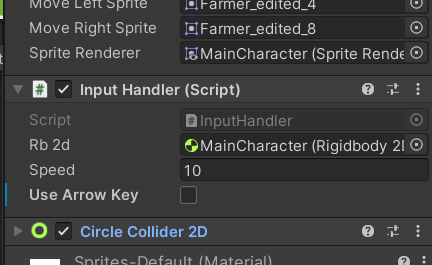
FarmVille

# Design Pattern

## {Huyền} [Behavioral Pattern] Command Pattern: *Main Character Movement - Keyboard*

* Code trc khi có Command:
  + 
* => Sử dụng Command Pattern:
  + Có 2 cách di chuyển:
    - Dùng phím mũi tên
    - Dùng WASD
  + => Ng dùng có thể tự do chuyển đổi giữa 2 cách di chuyển = cách sửa “Use Arrow Key”
    - 
* **Implementation**:
  + Command: abstract class cho các Command implement
  + MoveRight, MoveLeft, MoveUp, MoveDown, Stop, DoNothing: các Command implement, tương ứng với từng hành động di chuyển
  + InputHandler: Sử dụng Command Pattern:
    - Định nghĩa các Object Command
    - -> Tùy theo người dùng lựa chọn sử dụng cách di chuyển nào mà khởi tạo Command tương ứng cho từng Object
    - -> Mỗi khi ng dùng bấm 1 Button -> Gọi đến method thực thi hành động của Command tương ứng

## {Giang} [Behavioral Pattern] State Design Pattern: *Main Character Animation Transition*

* Sử dụng Pattern để quản lý Animation của Nhân vật chính:
  + Nhân vật sẽ có những State di chuyển khác nhau: lên, xuống, trái, phải
  + Mỗi State lại có hành động khác nhau: thể hiện Sprite tương ứng
  + => Sử dụng State Design Pattern để quản lý và thay đổi State di chuyển cho Nhân vật mỗi khi Nhân vật có hướng di chuyển mới -> từ đó thể hiện Sprite tương ứng phù hợp
* **Implementation**:
  + IMoveState: interface cho các States implement
  + MoveLeftState, MoveRightState, MoveUpState, MoveDownState: các States khác nhau của Animation, implement hành động (chuyển sprite) tương ứng cho state đó
  + MoveStateContext: dùng để lưu trữ State hiện tại của GameObject và chuyển state
  + MoveController: điều khiển các State = cách sử dụng Context
  + Sử dụng Pattern: gọi đến Controller và qua đó gọi đến từng method chuyển state khác nhau (Kết hợp với Command Pattern -> Code bên trong từng Command)

# Algorithm

## {Giang} Dijkstra Shortest Path Algorithm: *Main Character Movement – Mouse Click*

# Data Structure

## {Giang} Undirected Weighted Graph: Used in: [Dijkstra Shortest Path Algorithm]

* **Implementation**:
  + GraphNode:
  + Graph:

## {Giang} Sorted Linked List: Used in: [Dijkstra Shortest Path Algorithm]

# Others

## {Giang} Background & Map Creation:

[***https://learn.unity.com/tutorial/introduction-to-tilemaps***](https://learn.unity.com/tutorial/introduction-to-tilemaps)

**Tilemap - Guide**:

* 1. What is Tilemap?
* 2. Installing the Tilemap Editor
  + Ko cần
* 3. Create a Tilemap
  + **Hierarchy**: 2D Objects -> Tilemap -> Rectangular
* 4. About the Tilemap system
  + Ở phần Grid có thể chỉnh các thuộc tính:
    - Cell Size
    - Call Gap
    - Call Layout
    - Call Swizzle
* 5. Configuring the Tilemap
  + *Tilemap Component*
  + *Tilemap Renderer Component*
* 6. Exploring the **Tile Palette**
  + *Window -> 2D -> Tile Palete*
  + Tools:
    - Selection
    - Move
    - Bursh
    - Fill Selection
    - Tile Sampler
    - Eraser
    - Fill
* 7. Using the Tile Palette
  + Create New Tile Palette
* 8. Using Tilemap Collider 2D
  + Tilemap Renderer -> Add Component: **Tilemap Collider 2D**

**Tilemap – Implementation:**

* **Tilemap**:
  + **TileMap\_Background**: vẽ những Object nằm ở Background
    - Gồm: Đường đi
    - **Sorting Layer**: *Background*
  + **TileMap\_Platformer:** vẽ những Object trên Map, có **Collider**
    - Gồm: Ô đất, Tường bao quanh Map
    - **Sorting Layer**: *Platformer*
* **Tile Palettes:**
  + Folder **Palettes**: Lưu trữ các Palette khác nhau
    - *Background\_Road\_FarmPlots*
    - *Background\_Road\_FarmPlots\_2*
  + Folder **Tiles**: Lưu trữ các Tiles trong Palette
* **Grid**:
  + *Cell Size: 0.6* (để phù hợp với kích cỡ của Palette)