



Machine à état en C# pour Unity



La Citation du jour :

“Mesurer les progrès de la programmation en lignes de code, c’est comme mesurer les progrès de la construction d’un avion en poids.” Bill Gates (qui a aussi dit qu’internet était une mode qui ne durerait pas !)



Et pour commencer :

- **Rappel des épisodes précédents**
- **Et les projets alors ?**
- **Et aujourd'hui on fait quoi ?**



Programme de la journée:

- **Nouveautés du jour : les Enums**
- **Nouveautés du jour : les Machines à état**
- **Un projet : SoloTank**



Les enums

Définition : Ensemble de constantes nommées. exemple : au lieu de créer un Array de string on peut faire une enum contenant : {Spring, Summer, Autumn, Winter}.

Intérêt :

- Une enum n'est pas un array ou une liste ! On ne fait pas de boucle dedans, on ne modifie pas les valeurs. Une enum est un ensemble de CONSTANTES !
- Permet d'associer des valeurs à des string facilement, pour des listes déroulantes par exemple.
- Peut être utilisé pour des opérations binaires (pas vu dans ce cours)

Utilité pour notre projet : Gestion des états. Nous allons utiliser une enum pour gérer les états des tourelles dans le solo tank.

Déclaration : On déclare une enum et on définit immédiatement les noms des constantes :

```
enum Season
{
    Spring,
    Summer,
    Autumn,
    Winter
}
```



Les machine à états

Définition : Une machine à nombre d'états fini (Final State Machine ou FSM) sert à modéliser le comportement séquentiel d'un objet. Elle comporte un nombre limité et défini d'états, d'un nombre de transitions possible entre chaque état et d'actions associées à chaque état et à chaque transitions.

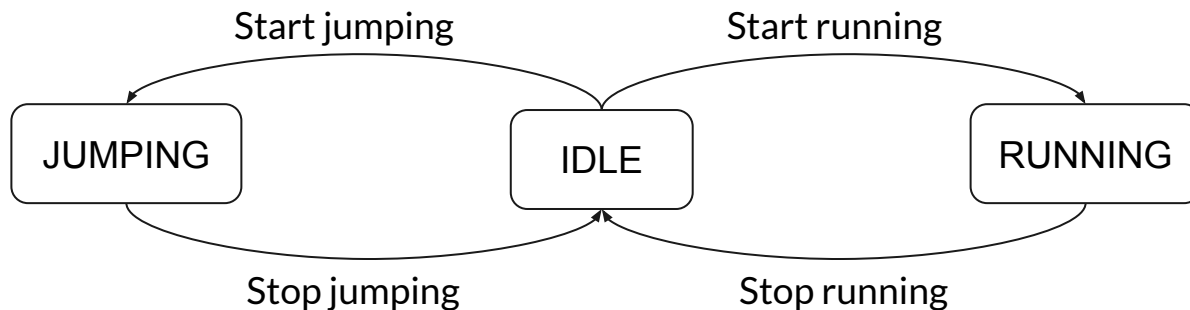
Un état correspond à la description que l'on pourrait faire en commence par “en train de ...”, et une transition est définie par un verbe conjugué.

Intérêt :

- Nous allons pouvoir gérer chaque état de manière indépendante.
- Nous allons définir les conditions de transition, ainsi que les actions associées

Les machine à états

Exemple : Un personnage de plateforme connaît trois états : Idle (immobile), Running (en train de courir), Jumping (en train de sauter). et 4 transitions : Start running (commence à courir), Stop running (s'arrête de courir), Start Jumping (saute), Stop jumping (atterris).





Les machine à états

Exemple : Dans le projet d'exemple vous pouvez voir le déplacement d'une balle comme dans un Roll_A_Ball. le script est toutefois très différent.

Exercice :

1. Ajouter un déplacement selon l'axe vertical avec la même vitesse que sur l'horizontal
2. Ajouter une transition pour que l'on puisse sauter en state Running

Et la question finale : cela vous rappelle-t-il une autre fonctionnalité d'Unity qui est gérée de la même manière ?



Les machine à états

Application concrète de base : Modifier le SoloTank.

L'objectif est d'appliquer les états aux tourelles pour quelles soient en mode Détection ou Alerte.

L'état Détection doit trouver le joueur s'il est à moins de XXm de la tourelle.

L'état Alerte doit tirer sur le joueur en le suivant du regard tant qu'il est dans sa zone de détection.

Une transition est possible de la détection vers le tir : si le joueur entre dans la zone de détection.

Une autre transition est possible du tir vers la détection si le joueur sort de la zone de détection.



Les machine à états

Application concrète avancée: Modifier le SoloTank.

L'objectif cette fois est de faire des tourelles qui se déplacent. Il faut donc créer une nouvelle classe, avec les états Patrol / Alerte.

L'état Patrol doit déplacer la tourelle sur un chemin prévu et trouver le joueur s'il est à moins de XXm de la tourelle.

L'état Alerte doit tirer sur le joueur en le suivant tant qu'il est dans sa zone de détection.

Une transition est possible de la patrol vers l'alerte : si le joueur entre dans la zone de détection.

Une autre transition est possible de l'alerte vers la patrol si le joueur sort de la zone de détection.



Les machine à états

Application concrète plus avancée: Modifier le SoloTank.

L'objectif cette fois est de faire des tourelles qui fuient si elles sont trop blessées et se font réparer dans leur zone de réparation. Il faut donc modifier la classe précédente en ajoutant un état RunAway, et un état Repairing.

L'état RunAway doit déplacer la tourelle un point de fuite prédésigné.

L'état Repairing doit remonter les pv de la tourelle régulièrement jusqu'au PV max.

Une transition est possible des deux autres états (Patrol et Alerte) si la tourelle descends en dessous de 25% de ses PV.

Si la tourelle retrouve ses PV Max elle repasse en état Patrol.