J05 – Outils – Outils Graphique

C#

Struct

- stocker des type primitifs qui vont être souvent manipulés
- Inférieur a 16 octets
- dans le stack => passage par copy
- Peut utiliser ref pour palier a ce probleme

Généricité

- Classes ou méthodes indépendants du type.
  List<GameObject>
  Dictionnary<K,V>

- Sur méthode

```
public void MyParameter<P>(P p1, P p2) {
}
```

- KeyWord default

Généricité

- Contrainte sur les génériques

Peut ajouter de la précision sur les génériques
where T : struct -> Le type générique doit être un type valeur
where T : class -> Le type générique doit être un type référence
where T : new() -> Le type générique doit posséder un constructeur par défaut
where T : IMonInterface -> Le type générique doit implémenter l'interface ImonInterface
where T : MaClasse -> Le type générique doit dériver de la classe MaClasse
where T1 : T2 -> Le type générique doit dériver du type générique T2

Généricité

- Sur les class

On utilisera la contrainte new() si on a besoin d initialiser le générique dans la classe

```
public class ClassAvecAssociationEtNew<T> where T : new(){
    T t = new T();
}
```

Méthode extension

ajouter une méthodes à une classe déjà existante Vector3, String ...

- Méthode statique dans une classe statique
- Premier paramètre de la méthode doit être du même type que la classe
- Doit utiliser le mot clef this sur ce paramètre.
- Au moment de l’appel doit utiliser using et le namespace ou se trouve la classe statique

Delegate

- Delegate est un pointeur sur un tableau de fonction
- Permettre de définir un Type qui représente une signature de méthode.
- On pourra ajouter alors n’importe quelle méthode qui respecte cette signature dans le tableau.
- Ajoute une méthode
- Plusieurs méthodes
- Peut utiliser une fonction anonyme
- Existe Delegate Générique
- Pattern Observé / Observateur