

# TD n° 5

## Programmation avec Unreal Engine 2

### 1 Installation d'un Environnement de Programmation pour UT 2004

Vous trouverez, comme d'habitude, l'ensemble des programmes à installer pour ce TD à l'adresse suivante :

<http://trolen.polytech.unice.fr/cours/maje/td5/>

#### 1.1 Installation d'Unreal Engine 2

Commencez par installer Unreal Engine 2 (UE2) qui est le moteur et l'éditeur d'Unreal Tournament, gratuit pour toute utilisation à but non commercial ou pour l'enseignement. **Lors de l'installation, sélectionnez l'ensemble des options disponibles (installation du code source des scripts, association d'UE2 avec les fichiers urm, etc.)**

Cette installation vous permettra d'avoir accès à deux outils :

- UnrealEngine 2 Runtime (UE2R) : le moteur d'Unreal Tournament 2004 qui permet de tester un développement ;
- UnrealEngine 2 Editor (UE2E) : l'éditeur de niveau identique à celui que nous avons pu utiliser dans UT lors de la précédente séance.

Vous devez lancer UnrealEngine Runtime afin de vérifier que l'installation s'est bien passée et pour créer les fichiers nécessaires lors du premier lancement (création du fichier UE2Runtime.ini au premier lancement). Cette fois-ci de toute façon, il n'est pas possible de jouer, on peut uniquement se promener, donc cela ne devrait pas durer longtemps ! 😊

Vous passez du mode vue subjective au mode caméra (vous permettant de voir le personnage) en utilisant la touche B.

#### 1.2 Installation des fichiers du nouvel exemple

Installez le paquetage `partie1.zip` contenant les fichiers d'un nouvel exemple pour UE2 sur lequel nous allons travailler. Cette opération a pour résultat d'ajouter les fichiers suivants :

```
Animations/UDN_CharacterModels_K.ukx (fichier des modèles)
Textures/UDN_CharacterModels_T.utx (fichiers des textures associés aux modèles)
MAJE/Classes/NewPawn.uc (fichiers de code source Unreal Script)
```

#### 1.3 Installation d'un IDE pour Unreal : WOTgreal

Installez le programme WOTGreal sur votre ordinateur. Après l'installation, lors du premier lancement du programme, un assistant de configuration vous proposera le réglage de différentes options. **Vous veillerez à bien sélectionner la configuration de l'outil pour Unreal Engine et non, pour Unreal Tournament.** Si vous avez correctement installé et configuré votre outil, vous devriez voir apparaître dans la liste le paquetage MAJE contenant la classe `NewPawn`. Les procédures d'installation étant terminées, passons maintenant à des choses plus intéressantes.

A l'aide de cet éditeur, vous modifierez le fichier `UE2Runtime.ini` pour faire en sorte que l'application démarre en mode fenêtré, dans une résolution de 800x600.

### 2 1...2...3... Codez !

Tous les exemples que nous allons ajouter vont, comme la semaine dernière, être créés dans la package MAJE. Vous veillerez à déclarer ce nouveau paquetage dans la liste des paquetages utilisés (et à compiler par WOTgreal).

## TD n° 5

# Programmation avec Unreal Engine 2

## 2.1 De nouveaux personnages pour votre jeu favori

### 2.1.1 Etude de la classe `NewPawn`

Vous commencerez par examiner la classe `NewPawn` qui implémente le comportement de notre nouvelle classe de personnage. Cette classe définit certains comportements comme la marche, la course, la nage, etc. Etudiez un peu le code pour vérifier que vous êtes bien capable d'identifier ces différentes parties. Ceci vous permettra d'identifier les structures à créer et les valeurs que vous devrez mettre lors de la prochaine étape.

### 2.1.2 Création d'une nouvelle classe `NewBoy` ou `NewGirl`

Le but de cet exercice est de créer une classe `NewBoy` (ou `NewGirl`) qui vous permettra d'associer une représentation graphique à votre nouvelle classe `NewPawn` qui implémente le comportement d'un nouveau type de personnage. Pour arriver à vos fins, il faudra donc créer cette classe `NewBoy` (ou `NewGirl`) à partir de la classe `NewPawn` (par héritage). En l'absence de définition d'un comportement spécifique pour ces nouveaux personnages, le travail à réaliser consiste donc à associer les animations qui vous ont été fournies aux différentes variables et structures qui sont utilisées pour l'affichage du personnage dans la classe `NewPawn`. Cette association est réalisée dans la section `defaultproperties` de votre classe. Pour connaître les valeurs à associer, vous chargerez `UDN_CharactersModel_K.ukx` dans l'outil de visualisation des animations de l'éditeur.

Dans un premier temps, vous devez au minimum associer les bonnes valeurs aux variables suivantes: `Mesh`, `RunningAnims`, `StandingTurnAnims`, `StandIdle`, ainsi que les variables suivant le type de personnage aux valeurs:

`NewBoy`

```
BaseEyeHeight=110
CollisionRadius=+00040.000000
CollisionHeight=+00113.000000
CrouchHeight=+80.0
```

`NewGirl`

```
BaseEyeHeight=100
CollisionRadius=+00040.000000
CollisionHeight=+00100.000000
CrouchHeight=+72.0
```

Pour réaliser vos tests, vous configurerez UE2R pour faire en sorte d'utiliser le nouveau personnage disponible. Pour effectuer cette opération, il suffit de modifier le fichier de configuration `User.ini` (à l'aide de `WOTgreal`) qui se situe dans le répertoire `System`. Dans la section `[DefaultPlayer]`, il suffit de modifier la variable `Class` pour spécifier :

```
Class=MAJE.NewBoy
```

ou

```
Class=MAJE.NewGirl
```

Relancez l'exécution pour vérifier que la modification a bien été prise en compte.

D'autres méthodes peuvent être utilisées pour faire cette opération :

- Au lancement du programme, en utilisant la commande suivante :

```
UE2Runtime EM_Runtime?Class=MAJE.NewBoy
```

- Lorsque le programme est lancé, en utilisant la console pour exécuter l'instruction suivante :

```
Open EM_Runtime?Class=MAJE.NewBoy
```

Après avoir testé que l'association des animations est bien réalisée correctement, vous pourrez continuer à associer les autres animations pour marcher, voler, nager, ramper, etc. Vous utiliserez la carte à disposition permettant au moins de tester la marche (debout ou accroupi), la course, la nage, quand le personnage reste statique dans une position, etc. Vous utiliserez la touche `F5` pour passer de vue subjective en vue à la troisième personne (où l'on voit le personnage avec lequel on joue).

## TD n° 5

# Programmation avec Unreal Engine 2

### 2.2 Redéfinition d'un comportement d'attente pour le personnage

Ajoutez à votre classe afin de donner une attitude plus « naturelle » à votre personnage. Pour cela, vous devrez ajouter une fonction à votre classe pour faire en sorte de varier les comportements quand votre personnage attend. Au bout d'un certain temps (toutes les 1 ou 2 animations correspondant à `IdleLong`), vous intercalerez une animation ou le personnage se gratte le nez, s'étire, etc. Vous devrez pour cela ajouter les variables et structures nécessaires et pourrez faire appel à la fonction `Rand(int Max)` pour retourner un nombre aléatoire en 0 et `Max - 1`.

### 2.3 NewBoy, NewGirl

Vous ferez en sorte pour cette dernière partie concernant les personnages de modifier votre code pour avoir deux classes de personnages (un garçon et une fille) qui héritent tous deux du comportement que vous venez de définir.

## 3 Modification du contrôle du personnage

### 3.1 Définition de nouvelles classes : NewGameInfo, NewHUD, NewScoreBoard

Récupérez le deuxième paquetage `partiez.zip` qui vous installera de nouvelles classes utiles à la suite du développement de notre projet. Cette opération a pour résultat d'ajouter les fichiers suivants :

```
Textures/NewFonts_T.utx
    NewHUD_T.utx
MAJE/Classes/NewGameInfo.uc
    NewHUD.uc
    NewScoreboard.uc
```

Modifier le fichier `UE2Runtime.ini` afin de ne plus utiliser la classe `RTGameInfo` mais la classe `NewGameInfo`. Avant de pouvoir tester, vous prendrez soin de définir la classe `NewController` que nous allons implémenter dans la partie suivante.

### 3.2 Définition d'une classe de contrôle : NewController

En vous basant sur la classe `Runtime.RTPlayerController.uc`, vous implémenterez une nouvelle classe `NewController` qui permettra de faire danser votre personnage (le comportement est déjà implémenté dans la classe `NewPawn` et vous aurez pris soin d'associer une animation à l'action « danser » dans la classe implémentant le personnage que vous utilisez. Dans cette classe, vous ajouterez une fonction

```
exec function DoDance()
```

Dans cette classe, vous vérifierez que le type de personnage est un `NewPawn`, que l'on se trouve bien dans le modèle physique terrestre (marche) et que l'on n'est pas en train de danser. Sous ces conditions, on pourra déclencher l'animation de danse que vous aurez choisie.

Il vous suffira alors d'associer la fonction `DoDance` que vous venez de définir à une touche du clavier. Pour cela vous avez deux possibilités :

- Modifiez le fichier `User.ini` et lui ajouter la ligne suivante :

```
G=DoDance
```

ou

- Invoquez le mode console dans le moteur (touche « `ù` ») pour y taper une commande du type :

```
set input g DoDance
```