Create a following system :

In a random place on the level, place exactly one actor (seeker), who will be detecting other actors (wanted) on the map. After the detection, the seeker informs wanteds about the fact that they have been detected, and prints onto the screen (in any form) the distance from the seeker in order from the closest to the farthest. Additionally, the seeker picks two found actors, the closest and the farthest, and informs them accordingly about the distance.

Features of the seeker mechanism :

● Whole mechanism must be closed inside an Actor Component, so that you can reuse it in any actor desirable.

● Seeker can only detect Actors, which have the appropriate Actor Component.

● Seeker can only detect Actors, which have the appropriate distance from him.

● Seeker can only detect Actors, which are in his line of sight (are not hidden behind a wall).

Features of the wanted mechanism :

● Whole mechanism must be closed inside an Actor Component, so that you can reuse it in any actor desirable.

● Actor Component attached to wanted is required for him to be detected.

Assumptions :

● Task should be done on the newest Unreal Engine 4 version.

● System must be written in the C++ programming language.

● If there is a need, BVS (Blueprint Visual Scripting) language can be used for small tasks.

● System must be based on the Unreal Engine 4 API.

● Development configuration build can be done from the project that system was created in.

● Information flow should be seen in the log.

● Assets appearance is not relevant.

● System optimised for around 250 actors.

● Detection must be performed continuously in real time.

● Task should be done within 48 hours from the receivement

Utwórz następujący system:

W losowym miejscu na poziomie umieść dokładnie jednego aktora (poszukiwacza), który będzie wykrywał innych aktorów (poszukiwanych) na mapie. Po wykryciu poszukiwany informuje poszukiwanego o fakcie wykrycia oraz drukuje na ekranie (w dowolnej formie) odległość od poszukiwanego w kolejności od najbliższej do najdalszej. Dodatkowo poszukiwacz wybiera dwóch znalezionych aktorów, najbliższego i najdalszego, i odpowiednio informuje ich o odległości.

Cechy mechanizmu szukającego:

● Cały mechanizm musi być zamknięty w Actor Component, tak aby można go było ponownie wykorzystać w dowolnym pożądanym aktorze.

● Poszukiwacz może wykryć tylko Aktorów, którzy mają odpowiedni Komponent Aktora.

● Poszukiwacz może wykrywać tylko Aktorów, którzy znajdują się w odpowiedniej odległości od niego.

● Poszukiwacz może wykryć tylko Aktorów, którzy znajdują się w jego polu widzenia (nie są schowani za ścianą).

Cechy poszukiwanego mechanizmu:

● Cały mechanizm musi być zamknięty w Actor Component, tak aby można go było ponownie wykorzystać w dowolnym pożądanym aktorze.

● Komponent aktora dołączony do poszukiwanego jest wymagany do jego wykrycia.

Założenia :

● Zadanie należy wykonać na najnowszej wersji Unreal Engine 4.

● System musi być napisany w języku programowania C++.

● W razie potrzeby do drobnych zadań można użyć języka BVS (Blueprint Visual Scripting).

● System musi być oparty na API Unreal Engine 4.

● Konfigurację programistyczną można zbudować z projektu, w którym system został utworzony.

● Przepływ informacji powinien być widoczny w logu.

● Wygląd zasobów nie ma znaczenia.

● System zoptymalizowany dla około 250 aktorów.

● Wykrywanie musi odbywać się w sposób ciągły w czasie rzeczywistym.

● Zadanie należy wykonać w ciągu 48 godzin od otrzymania

* zmienić camere na pawn
* ograniczenie wanted na 250