

Environment Query System, EQS

주요 개념 및 구성 요소

1. 쿼리(Queries):

- 쿼리는 AI가 환경에서 정보를 수집하고 분석하는 과정입니다. EQS 쿼리는 여러 테스트와 필터를 통해 특정 조건을 만족하는 위치나 객체를 찾습니다.

2. 생성기(Generators):

- 생성기는 쿼리의 출발점으로, 테스트할 위치나 객체의 세트를 만듭니다. 예를 들어, 격자(grid) 생성기는 일정한 간격으로 배치된 점들을 생성합니다.

3. 테스트(Tests):

- 테스트는 생성된 점이나 객체가 특정 기준을 충족하는지 평가합니다. 예를 들어, 거리 테스트는 특정 지점에서 얼마나 먼지, 가시성 테스트는 특정 지점이 보이는지 등을 평가합니다.

4. 필터(Filters):

- 필터는 테스트를 통과한 결과를 추가로 좁히는 역할을 합니다. 예를 들어, 가장 가까운 지점을 선택하거나 특정 조건을 만족하는 지점만 남길 수 있습니다.

5. 컨텍스트(Context):

- EQS 쿼리는 다양한 컨텍스트 정보를 사용하여 상황에 맞는 결정을 내립니다. 예를 들어, AI 캐릭터의 현재 위치나 목표 위치를 컨텍스트로 사용하여 쿼리를 수행할 수 있습니다.

사용 사례

1. 적 탐지 및 추적:

- AI가 적을 탐지하고 추적하는 데 EQS를 사용하여 가장 좋은 위치를 찾습니다. 예를 들어, 적이 잘 보이는 위치나 적에게 접근하기 쉬운 위치를 선택할 수 있습니다.

2. 은신처 찾기:

- AI가 공격을 피하기 위해 숨을 수 있는 장소를 찾는 데 EQS를 사용할 수 있습니다. 이 경우, 가시성 테스트와 거리 테스트를 통해 적에게 보이지 않는 위치를 찾습니다.

3. 최적 경로 탐색:

- AI가 목적지로 가는 최적의 경로를 찾기 위해 EQS를 사용할 수 있습니다. 여러 위치의 가시성, 장애물 유무 등을 평가하여 안전하고 효율적인 경로를 선택합니다.

EQS 사용 시 이점

1. 성능 최적화

효율적인 쿼리 처리

EQS는 복잡한 환경 쿼리를 효율적으로 처리할 수 있도록 설계되었습니다. 이는 AI가 환경에서 최적의 위치를 찾는 데 필요한 계산을 최소화합니다.

- **병렬 처리:** EQS는 병렬 처리를 지원하여 여러 쿼리를 동시에 실행할 수 있습니다. 이는 특히 많은 수의 AI 캐릭터가 있는 상황에서 성능을 크게 향상시킵니다.
- **적응형 샘플링:** EQS는 필요에 따라 샘플링 밀도를 조정할 수 있습니다. 예를 들어, 필요할 때는 더 많은 샘플을 생성하고, 그렇지 않은 경우 샘플 수를 줄여 성능을 최적화합니다.

테스트와 필터의 최적화

EQS의 구조는 다양한 테스트와 필터를 적용하여 필요한 데이터를 효율적으로 선별할 수 있게 합니다. 이는 AI가 필요한 데이터만 처리함으로써 불필요한 연산을 줄여줍니다.

- **테스트 병합:** 여러 테스트를 한 번에 수행하여 중복 계산을 줄입니다.
- **효율적인 필터링:** 중요도가 높은 데이터를 우선적으로 처리하여 최적의 결과를 빠르게 도출합니다.

2. 개발 효율성

빠른 프로토타이핑

EQS를 사용하면 복잡한 AI 행동을 신속하게 프로토타이핑하고 테스트할 수 있습니다.

- **비주얼 스크립팅:** 블루프린트를 사용하여 시각적으로 쿼리를 생성하고 수정할 수 있어, 프로그래밍 경험이 적은 개발자도 쉽게 사용할 수 있습니다.
- **빠른 피드백 루프:** 즉각적인 피드백을 통해 쿼리의 결과를 실시간으로 확인하고 조정할 수 있습니다.

재사용성과 모듈화

EQS 쿼리는 다양한 AI 캐릭터와 시나리오에 재사용할 수 있습니다. 이는 코드의 재사용성을 높이고, 개발 시간을 줄이는데 도움이 됩니다.

- **모듈화된 쿼리:** 다양한 시나리오에 맞게 모듈화된 쿼리를 생성하고, 필요에 따라 쉽게 수정할 수 있습니다.
- **공유 가능성:** 동일한 쿼리를 여러 AI 캐릭터가 공유할 수 있어, 일관된 행동 패턴을 유지하면서도 개발 시간을 절약할 수 있습니다.

3. AI 복잡성과 유연성

정밀한 환경 분석

EQS는 AI가 환경을 더 정밀하게 분석하고 이해할 수 있도록 도와줍니다. 이는 AI의 행동을 보다 자연스럽게 전략적으로 만듭니다.

- **다양한 테스트 조합:** 다양한 테스트를 결합하여 복잡한 조건을 설정할 수 있습니다. 예를 들어, 거리, 가시성, 커버리지 등을 동시에 고려할 수 있습니다.
- **동적 환경 대응:** 환경 변화에 실시간으로 대응하는 동적 쿼리를 통해 AI의 반응성을 높일 수 있습니다.

고급 전술 결정

EQS를 통해 AI가 더 복잡한 전술 결정을 내릴 수 있습니다. 이는 게임 플레이어의 깊이를 더하고, 플레이어에게 도전적인 경험을 제공합니다.

- **전술적 위치 선정:** 보스 몬스터가 플레이어의 위치와 환경을 분석하여 최적의 전투 위치를 선택할 수 있습니다.
- **상황별 행동 선택:** 상황에 따라 다양한 행동을 선택할 수 있어, 예측 불가능한 AI 행동을 구현할 수 있습니다.


수치적 사례

- **성능 향상:** EQS를 사용하면 전통적인 방법보다 쿼리 처리 속도가 30% 이상 향상될 수 있습니다.
- **개발 시간 단축:** 프로토타이핑과 테스트 시간을 50% 이상 단축시킬 수 있습니다.
- **AI 복잡성 증가:** EQS를 사용한 AI는 전통적인 방법에 비해 2배 이상의 복잡한 행동 패턴을 구현할 수 있습니다.

결론

Unreal Engine의 EQS를 사용하면 성능 최적화, 개발 효율성 향상, AI의 복잡성과 유연성 증대를 통해 수치적으로 많은 이점을 얻을 수 있습니다. 이를 통해 보다 정교하고 반응성 높은 AI를 구현할 수 있으며, 개발 과정에서의 시간과 비용을 절감할 수 있습니다.


언리얼 엔진의 인바이런먼트 쿼리 테스트 폰 | 언리얼 엔진 5.3 문서 | Epic Developer Community
EQS 테스트 폰을 사용하여 EQS 쿼리가 무엇을 하고 있는지 디버그 및 표시하는 방법을 설명합니다.

 https://dev.epicgames.com/documentation/ko-kr/unreal-engine/environment-query-testing-pawn-in-unreal-engine?application_version=5.3



Environment Querying System (EQS) : Basic Tutorial

> 들어가기 앞서 > 접기 Environment Querying System : Basic Tutorial 원문 : T06: ...

 <https://blog.naver.com/PostView.naver?blogId=devdeepblue&logNo=220282035752>

