**컴공이면 게임해야조 테스트 명세서**

**기능 시험 명세서 (Functional Test Cases)**

1. **기능 명세서 1: 게임 실행**
   * **목적**: 플레이어가 성공적으로 게임에 접속할 수 있는지 확인
   * **사전 조건**: 플레이어가 게임에 접속
   * **입력 데이터**: 플레이어의 연결 요청
   * **예상 결과**: 플레이어가 성공적으로 세선에 연결되고 게임이 시작됨.
2. **기능 명세서 2: 이동 처리**
   * **목적**: 체스 말의 이동 규칙이 정상적으로 적용되는지 확인
   * **사전 조건**: 게임 진행 중
   * **입력 데이터**: 플레이어의 이동 요청 Ex) Rook (0,0) to (0,4)
   * **예상 결과**: 룩이 (0,4)로 정상적으로 이동하고 보드가 업데이트 됨.
3. **기능 명세서 3: 전투 처리**
   * **목적**: 전투 이벤트가 제대로 처리되는지 확인.
   * **사전 조건**: 게임 진행 중.
   * **입력 데이터**: VR 플레이어가 상대 기물과 전투 페이즈에 돌입함.
   * **예상 결과**: 전투 이벤트가 발생하며 전투 결과가 서버로 전송됨.
4. **기능 명세서 4: 순서 처리**
   * **목적**: WEB 플레이어(흑/백)와 VR 플레이어(흑/백)의 순서가 규칙대로 진행되는지 확인
   * **사전 조건**: 게임 진행 중.
   * **입력 데이터**: 플레이어의 턴을 5번 이상 진행
   * **예상 결과**: WEB White -> VR White -> WEB Black -> VR Black -> WEB White 순서로 턴이 진행됨.
5. **기능 명세서 5: 실시간 메시징**
   * **목적**: 체스판 상태와 플레이어 간의 실시간 메시지가 정상적으로 전송되는지 확인.
   * **사전 조건**: 모든 플레이어가 게임에 참가한 상태.
   * **입력 데이터**: 한 플레이어가 체스 말을 이동한 후 상대방에게 메시지 전송.
   * **예상 결과**: 상대방이 즉시 업데이트된 체스판을 확인하고, 전송된 메시지를 받을 수 있음.
6. **기능 명세서 6: 종료 및 재실행**
   * **목적**: 게임이 재설정되는지 확인
   * **사전 조건**: 게임 진행 중
   * **입력 데이터**: 플레이어의 게임 초기화 요청 또는 게임 종료
   * **예상 결과**: 모든 플레이어와 서버의 상태가 초기 상태로 돌아가고 재시작 가능.

**비기능 시험 명세서 (Non-Functional Test Cases)**

1. **비기능 명세서 1: 보안 테스트**

* **목적**: 체스 게임 시스템이 적절한 보안 조치를 통해 외부 공격을 차단하는지 확인.
* **사전 조건**: 클라이언트가 웹소켓을 통해 서버에 연결.
* **입력 데이터**: 비정상적인 웹소켓 연결 요청.
* **예상 결과**: 서버가 비정상적인 요청을 차단하고, 클라이언트의 비정상적인 접근이 방지됨.

1. **비기능 명세서 2: 사용성 테스트**

* **목적**: 사용자가 직관적으로 게임을 플레이할 수 있는지 확인.
* **사전 조건**: 웹 클라이언트가 체스 게임에 접속.
* **입력 데이터**: 사용자가 게임 UI를 통해 체스 말을 이동하는 과정.
* **예상 결과**: 사용자가 직관적으로 UI를 이용해 게임을 진행할 수 있으며, 모든 기능이 쉽게 접근 가능함.

1. **비기능 명세서 3: 가용성 테스트**

* **목적**: 시스템이 다운타임 없이 안정적으로 작동하는지 확인.
* **사전 조건**: 서버가 정상 작동 중.
* **입력 데이터**: 체스 게임을 24시간 연속 실행.
* **예상 결과**: 시스템이 중단 없이 24시간 동안 정상적으로 동작하며, 모든 클라이언트 요청이 처리됨.

1. **비기능 명세서 4: 확장성**

* **목적**: 시스템이 실시간 VR 모드에서 여러 캐릭터 간 상호작용을 성능 저하 없이 처리할 수 있는지 확인.
* **사전 조건**: 게임이 VR 모드에서 실행되고, Player와 Enemy (Khaimera 또는 Rampage)가 AI Perception과 Trigger Box를 사용하여 상호작용 중.
* **입력 데이터**: 여러 인스턴스의 적과 플레이어가 만들어지고 VR 환경에서 이동**.**
* **예상 결과**: 시스템은 상호작용(적이 플레이어를 추적, 애니메이션, 플레이어 입력 처리)을 성능 저하 없이 안정적으로 처리하고, 캐릭터 간 상호작용이 부드럽게 전환됨**.**

1. **비기능 명세서 5: 유지보수성**

* **목적**: 하나의 블루프린트를 변경해도 프로젝트의 다른 부분에 영향을 미치지 않도록 모듈화된 구조를 유지하는지 확인.
* **사전 조건**: PatrolController\_BP의 AI 행동을 변경하거나 새로운 Perception 유형 추가.
* **입력 데이터**: PatrolController\_BP의 AI 행동을 변경하거나 새로운 Perception 유형 추가.
* **예상 결과**: 다른 클래스를 재구성할 필요 없이 변경 사항이 반영되며, 시스템은 정상적으로 작동을 유지. 특정 블루프린트를 테스트해도 다른 블루프린트에 영향을 미치지 않음.

1. **비기능 명세서: 안정성(VR 상호작용)**

* 목적: Unreal Engine 환경에서 VR 상호작용이 매끄럽고 반응이 빠른지 확인.
* 사전조건: Player와 Enemy가 활성화된 상태에서 VR 환경에서 게임이 실행됨.
* 입력 데이터: 플레이어 이동, 적 감지 및 전투 상호작용을 시뮬레이션. 또한 Trigger Box 오버랩과 환경 상호작용을 테스트.
* 예상 결과: VR 움직임이 사용자 입력에 반응하고, 적이 플레이어 행동에 적절하게 반응하며 (추적, 공격, 멈춤), 이러한 상호작용 중에 게임플레이가 끊기지 않음.