**1. 프로젝트 개요**

|  |  |
| --- | --- |
| 항목 | 내용 |
| **게임 제목** | 프로토콜 이스케이프 |
| **개발 목표** | 빠른 템포와 전략적 전투 시스템을 제공하여 조작은 간단하고 도전적인 액션을 선사하는 그런 몰입감을 구현하는 것이 목표입니다. 플레이어에게 강렬하고 즉각적인 피드백을 제공하여 몰입도를 높이기 위해, 빠른 전투 템포와 정밀한 조작이 필수적입니다. |
| **장르** | 3D TPS, 액션, 슈팅, 싱글 플레이 |
| **타겟 플랫폼** | PC를 우선 개발 플랫폼으로 설정하며, 향후 콘솔로의 확장 가능성도 검토합니다. |
| **타겟 사용자** | 18~35세 남성, FPS 및 액션 게임 팬 |
| **주요 경쟁작** | Warhammer 40,000: Space Marine 2, HELLDIVERS™ 2, DOOM ETERNAL. 이들 게임은 유사한 슈팅 및 액션 시스템을 제공하며, 특히 DOOM ETERNAL의 빠른 템포 전투 시스템이 참고할 가치가 있습니다. |
| **핵심 특징** | 빠른 템포 전투, 수직적 맵 디자인, 다양한 무기 시스템, 그리고 직접적으로 보이는 클리어 타임을 다른 사람과 비교 할 수 있는 시스템으로 승부욕을 자극합니다. |

**추가 설명:**

1. 플랫폼 호환성을 고려하여 PC 개발을 우선적으로 진행하며, 필요 시 콘솔 버전 확장을 위한 컨트롤러 지원도 구현할 수 있습니다. PC와 콘솔의 차이를 고려한 조작 최적화가 필요합니다.
2. 기한 안에 완성하기 위해 확장성에 초점을 둔 스테이지 형식으로 제작합니다.

**2. 게임 컨셉**

|  |  |
| --- | --- |
| 항목 | 내용 |
| 세계관 설정 | 현대적인 연구소가 무기화된 생명체 연구로 인해 파괴되었습니다. |
| 스토리 요약 | 연구소에서 통제 불능의 실험체가 탈출하면서 주인공은 연구소를 탈출하는 내용입니다. |
| 캐릭터 설정 | 주인공은 보안 책임자로서 생존과 임무 수행 능력이 뛰어난 인물입니다. 상황이 일어난 후 연구소를 탈출하기 위해 고군분투 합니다. |
| UX 목표 | 처음에는 낮지만 점진적으로 높아지는 난이도와 긴장감을 제공하는 전투와 빠른 속도감과 스타일리쉬한 경험을 목표로 합니다. 플레이어는 스테이지별로 점점 더 도전적인 상황에 맞닥뜨리게 됩니다. |

**구현 내용:**

1. 빠른 전투 경험을 위한 타임어택 기능을 넣는다.
2. 스타일리쉬하고 도전적인 경험을 위해 플레이어가 몬스터를 완벽히 처치하면 버프를 받는다.
3. 스테이지의 마지막 맵의 보스를 해치워야 스테이지의 클리어이다.
4. Cinemachine을 이용해 긴장감 넘치는 연출을 극대화하며, 타임라인 기능으로 스토리 진행을 동적으로 연출합니다.

**3. 핵심 게임 시스템**

|  |  |  |
| --- | --- | --- |
| 기능 | 설명 | Unity 구현 방법 |
| 이동 | 캐릭터의 8방향 이동 및 카메라 기반 이동 | **CharacterController**로 물리 기반 이동 구현 |
| 공격 | 기본 무기 공격 및 특수 무기 사용 | **Raycasting**을 사용해 적과의 충돌 판정 처리 |
| 대쉬 | 짧은 거리 빠른 이동 및 회피 가능 | **Velocity** 조작으로 대쉬 구현, **Cooldown** 시스템 도입 |
| 더블 점프 | 특정 상태에서 더블 점프 가능 | **Velocity** 조작을 통한 점프 구현 및 두 번째 점프 판단 코드 |
| 무기 전환 | 여러 무기를 사용하여 다양한 전투 스타일 구현 | **Inventory 시스템**을 사용해 무기 변경 시스템 구현 |
| 플레이어 상태 | 플레이어의 현재 버프 디버프 상태에 따른 여러 효과 적용 | **BitMask** 연산을 통해 빠른 상태값 전달 구현 |
| 카메라 줌인 | 플레이어가 무기를 들고 줌인 키를 누르면 화면이 줌인 | **Cinemachine**의 값을 조작해 구현 |

1. **플레이어 시스템**

|  |  |  |
| --- | --- | --- |
| 기능 | 설명 | Unity 구현 방법 |
| 플레이어 추적 | 적이 플레이어를 인식하고 추적 | **NavMesh** 및 **AI 경로 탐색**을 통해 경로 추적 구현 |
| 공격 | 적이 일정 거리 내에 있을 경우 공격 | **Animator** 및 **Trigger**를 사용해 공격 애니메이션 구현 |
| 회피 및 방어 | 적이 일정 확률로 플레이어의 공격을 회피 | **Raycasting**을 사용하여 플레이어의 공격방향을 예상하고 특정 확률로 AI가 회피하도록 구현 |

1. **적 AI 시스템**
2. **게임 내부 시스템**

|  |  |  |
| --- | --- | --- |
| 시스템 | 설명 | Unity 구현 방법 |
| 체력 및 방어구 | 플레이어와 적의 체력, 방어구 시스템 | UI 체력 바, 방어구 시스템 연동 |
| 탄약 관리 | 플레이어와 적의 탄약 제한 및 보급 시스템 | **Inventory** 시스템과 연동하여 탄약 소모 및 충전 구현 |
| 데미지 처리 | 각 무기별 데미지 계산 및 적 사망 처리 | **Collider**와 **Raycast**로 데미지 충돌 판정 |

1. **레벨 디자인 및 진행 시스템**

|  |  |  |
| --- | --- | --- |
| **기능** | **설명** | **Unity 구현 방법** |
| **레벨 구성** | **던전형식으로 여러 방이 연결된 구조** | **ProBuilder를 사용해 레벨 디자인 및 NavMesh 설정** |
| **스테이지 진행** | **적을 처치할 때마다 다음 방으로 진행 가능한 문이 열림** | **Trigger 시스템으로 문 열림과 스테이지 클리어 조건 설정** |
| **클리어 조건** | **마지막 방의 보스를 클리어하면 한 스테이지를 클리어한다.** | **Trigger 시스템으로 마지막 문 열림과 게임 클리어 조건 설정** |

**추가 설명**:

1. 구현해야할 기능들 및 방법은 추후에 추가되거나 변경될 수 있습니다.

**4. 기술적 요구사항**

**4-1. 플랫폼 요구사항**

|  |  |  |
| --- | --- | --- |
| 플랫폼 요구사항 | 최소사양 | 권장사양 |
| 운영체제 | Windows 10, 64-bit /  Windows 11, 64-bit | Windows 10, 64-bit /  Windows 11, 64-bit |
| 프로세서 | Intel Core i3 | Intel Core i5 |
| 메모리 | 4 GB RAM | 8 GB RAM |
| 저장공간 | 1 GB 사용 가능 공간 | 1 GB 사용 가능 공간 |
| 그래픽 | - | Intel HD  Graphics 520 |

**4-2. 엔진 요구사항**

|  |  |
| --- | --- |
| 게임엔진 | 유니티 UNITY |
| 버전 | 2022.3.45f1 |
| 프로그래밍 언어 | C# |
| 프로그래밍 툴 | Visual Studio Code |
| 데이터 공유 방식 | GitHub |
| 비고 | - |

**4-3. 데이터베이스 요구사항**

|  |  |
| --- | --- |
| 데이터베이스 | SQLite |
| 데이터베이스 버전 | 3.46.1 (최신버전) |
| 비고 | - |

**4-4. 디자인 요구사항**

|  |  |
| --- | --- |
| **종류** | **사운드, UI , 텍스쳐 등** |
| **검색 방법** | **Unity Asset Store** |
| **데이터 공유 방식** | **GitHub** |
| **비고** | [**https://assetstore.unity.com/**](https://assetstore.unity.com/) |

**5. 스케줄 및 리소스**

1. **주간 일정표**

|  |  |  |
| --- | --- | --- |
| **스케줄** | **강의 주제** | **일정** |
| **1주차** | **Term 개용 설명** | **4인조 편성** |
| **2주차** | **프로젝트 주제발표** | **주제 정하기, ppt 만들기** |
| **3주차** | **기초 시스템 분석** | **기초 시스템 분석서 작성,**  **기초적인 기능들 구현(프로토타입 제작)** |
| **4주차** | **시스템 기능 정의** | **기초적인 기능들 구현**  **(프로토타입 제작)** |
| **5주차** | **Screen 레이아웃** | **기초적인 기능들 구현**  **(프로토타입 제작)** |
| **6주차** | **데이터베이스 설계** | **데이터베이스 연동 및**  **디버깅, 기능테스트(1차테스트)** |
| **7주차** | **시스템 상세설계** | **디버깅 및 ppt제작** |
| **8주차** | **중간점검 (평가)** | **시연 테스트** |

|  |  |  |
| --- | --- | --- |
| **스케줄** | **강의 주제** | **일정** |
| **9주차** | **중간 점검 수정사항 발표** | **자료 수정 및 발표,**  **게임 방향성 회의** |
| **10주차** | **프로젝트 개발** | **플레이어 관련 기능 구현** |
| **11주차** | **프로젝트 개발** | **적 AI관련 기능 완전구현,**  **맵 디자인 및 완전구현,**  **성능 테스트(2차 테스트)** |
| **12주차** | **프로젝트 개발** | **플레이어 및 적**  **애니메이션 제작 및 적용** |
| **13주차** | **프로젝트 개발** | **사운드,UI 등 유저 인터페이스 완전구현, 최종테스트** |
| **14주차** | **최종점검,프로젝트 시연** | **최종점검 및 버그수정** |
| **15주차** | **최종점검,프로젝트 시연** | **최종점검 및 버그수정** |

**6. 위험 요소 및 해결 방안**

**6-1. 팀원 간 작업 충돌**

|  |  |
| --- | --- |
| **설명** | * **팀원들의 작업물이 통합되면서 발생하는 버그나 충돌** |
| **해결 방안** | * **기획단계에서 변수나 함수의 이름 또는 타입 등 충돌이 예상되는 부분을 정확히 명시합니다.** * **GitHub를 사용해 버전을 통합하고 백업합니다.** |
| **비고** | **-** |

**6-2. 일정 지연**

|  |  |
| --- | --- |
| **설명** | * **예상보다 더 많은 작업 시간이 소요되거나 기능구현 실패로 인한 일정 지연된 경우** |
| **해결 방안** | * **각 주마다 회의를 통해 각 작업과정을 확인하고 조율합니다.** |
| **비고** | **-** |

**6-3. 성능 저하**

|  |  |
| --- | --- |
| **설명** | * **게임이 메모리나 그래픽을 과도하게 사용하여 컴퓨터가 느려지거나 프레임드랍이 발생** |
| **해결 방안** | * **코드적 최적화 문제를 개선하여 메모리 누수를 최소화하고 부족하다면 해상도를 조정합니다.** |
| **비고** | **-** |

**추가 설명:**

1. **유니티 엔진을 사용하므로 GitHub 레지스트리를 이용해 하나의 프로젝트에서 각자 맡은 부분과 역할을 작업하고 제작과 합치는 과정을 동시에 진행합니다.**
2. **팀 프로젝트이기 때문에 개개인의 사정에 의해 맡은 부분을 못하거나 미완한 경우 매주 회의를 통해서 부족한 부분에 대한 역할분배를 다시하고 게임의 방향성을 재확인하며 진행합니다.**
3. **게임속 최적화에 대해 최대한 생각하고 목표로 가지고있는 성능 수준을 지키며 서로가 제작한 부분에 대해 피드백하며 개선해 나갈 예정입니다.**

**7. 테스트 계획**

|  |  |  |
| --- | --- | --- |
| **테스트 종류** | **설명** | **테스트 기준** |
| **기능 테스트** | **모든 게임 기능이 의도대로 동작하는지 확인** | **버튼, UI, 전투, 이동 등의 작동 여부 확인** |
| **성능 테스트** | **게임이 원활하게 실행되는지, FPS가 일정하게 유지되는지 확인** | **60 FPS 이상 유지, 메모리 누수 여부 점검** |
| **프로그램 호환성 테스트** | **다양한 환경 및 플랫폼에서 게임이 실행되는지 테스트** | **최소 사양 및 권장 사양에서의 실행 여부 확인** |
| **사용자 피드백 테스트** | **실제 사용자의 피드백을 통해 UX 및 게임성 개선** | **UX 만족도, 난이도 적정성 등 사용자 반응 측정** |

**추가 설명:**

1. **성능 테스트에서는 PC 최소 사양(i5, 8GB RAM, GTX 1060)과 권장 사양(i7, 16GB RAM, RTX 3060)로 잡고 이를 기반으로 프레임 레이트와 CPU, GPU, RAM 사용량을 점검하여 최적화를 진행합니다.**
2. **기능 테스트에서는 기획된 기능들이 원하는 방향으로 작동하는지에 대해 테스트를 하고 결과를 기록 및 분석하여 게임 내 모든 기능이 제대로 작동하지는 확인하고 개선점을 찾습니다.**
3. **프로그램 호환성 테스트는 게임이 각자 팀원들의 컴퓨터를 통해서 다양한 하드웨어 및 소프트웨어 환경에서 제대로 작동하는지를 확인하는 과정으로, 사용자의 경험을 최적화하고, 다양한 플랫폼에서 게임이 기대한 대로 동작하게 할 수 있게 합니다.**
4. **사용자 피드백 테스트는 테스트에서 실시간 피드백을 받아 UX와 난이도를 조정하는 등 게임의 안정성과 품질을 보장하는 과정입니다.**
5. **모든 테스트 결과를 기록하고 저장하는 시스템을 구현합니다.**

**테스트 계획 및 일정:**

1. **지속 테스트: 지속적으로 매주 회의마다 소규모 테스트 및 평가**
2. **1차 테스트: 기능 테스트 및 초기 성능 테스트 (중간고사 전 약 6주차)**
3. **2차 테스트: 게임 중반부 기능 및 성능 최적화 테스트 (중간고사 이후 약 9주차)**
4. **최종 테스트: 프로젝트 시연 전 호환성 및 사용자 피드백 반영 (프로젝트 종료 2주 전)**

**8. 유지보수 계획**

**8-1. 유지보수 항목**

|  |  |  |
| --- | --- | --- |
| **유지보수 항목** | **설명** | **대응 방안** |
| **기능 추가** | **추가적인 게임 모드나 스테이지 추가** | **업데이트 계획 수립 후 정기적인 업데이트 진행** |
| **버그 수정** | **사용자 피드백 및 발견된 버그 해결** | **버그 추적 시스템을 통해 발견 즉시 패치** |
| **최적화** | **성능 개선 및 다양한 기기에서의 호환성 향상** | **GPU, CPU 사용량 최적화 및 메모리 사용량 개선** |

|  |  |
| --- | --- |
| **주기** | **항목** |
| **1주일** | **주요 버그 수정 및 작은 기능 업데이트** |
| **1개월** | **큰 기능 추가 및 확장팩 업데이트** |
| **추후 결정** | **새로운 모드, 스테이지 추가 및 성능 최적화** |

**8-2. 유지보수 주기**

**추가 설명:**

1. **기능 추가는 출시 후 새로운 무기, 캐릭터 스킨, 스테이지를 업데이트할 계획입니다. 이를 통해 유저들의 흥미를 유지하고 재방문율을 높일 수 있습니다.**
2. **버그 수정은 사용자로부터 받은 피드백을 바탕으로, 버그를 추적하고 이를 빠르게 수정하여 게임의 완성도를 높이는 것이 목표입니다.**
3. **최적화는 다양한 하드웨어 환경에서 안정적으로 게임을 실행할 수 있도록 메모리 관리와 그래픽 최적화를 중점으로 진행합니다.**
4. **주기적인 업데이트를 계획하고, 각 패치에 대한 GitHub를 통해 버전 관리를 철저히 진행하여 업데이트의 안정성을 확보합니다.**

**9. 법적 및 윤리적 고려사항**

|  |  |  |
| --- | --- | --- |
| **항목** | **설명** | **대응 방안** |
| **저작권 문제** | **게임 내 사용된 모든 리소스에 대한 저작권 확보 필요** | **사운드, 텍스처, 모델링 리소스에 대해 정당한 상용 라이선스 확인후 사용** |
| **개인정보 보호** | **개인정보를 수집하거나 저장하지 않음** | **개인정보 처리방침을 명시하여 사용자에게 투명하게 공개** |
| **사회적 영향** | **폭력적 콘텐츠에 대한 적절한 연령 제한** | **15세 이하 등급으로 분류될 수 있도록 게임 내 폭력적 요소 조정** |

**추가 설명:**

1. **저작권 문제: 게임에서 사용되는 모든 그래픽, 사운드 등에 대해 상용 라이선스를 확보하고, 타인의 저작물을 무단으로 사용하는 일이 없도록 주의합니다. 상업적 사용이 허가된 무료에셋 및 오픈소스를 사용 및 참고하여 제작할 예정입니다.**
2. **개인정보 보호: 게임이 싱글 플레이 기반이므로 사용자의 개인정보를 수집하지 않지만, 추후 온라인 기능 추가 시 개인정보 처리 방침을 명확하게 작성하고 이를 사용자에게 공지할 계획입니다.**
3. **사회적 영향: 게임 내 폭력적인 표현이나 잔인한 장면은 15세 이하 등급의 기준을 준수하도록 하고, 이러한 요소가 사회에 미치는 영향을 고려하여 연령 제한을 명확히 합니다.**
4. **상용 라이선스를 받은 사운드와 텍스처를 사용하고, 유료 에셋 사용 시에도 사용 범위를 철저히 확인하여 사용합니다.**

**10. 결론 및 추후 방향성**

**결론**

**이 프로젝트는 3D 액션 슈팅 게임으로, 빠른 템포와 전략적인 전투가 주요 특징입니다. 전투 중 다양한 무기 전환과 복잡한 맵 탐험을 통해 플레이어에게 끊임없는 도전과 재미를 제공합니다. 또한, 클리어 타임을 비교함으로써 플레이어들에게 경쟁심과 도전욕구를 자극시킵니다. 그리고 지속적인 업데이트와 유지보수 계획을 통해 장기적으로 플레이어의 흥미를 유지하며, 게임의 퀄리티를 높일 수 있는 기반을 마련했습니다.**

**추가 사항**

|  |  |  |
| --- | --- | --- |
| **DLC** | **새로운 스테이지 & 몬스터** | **새로운 무기 & 전투 메커니즘** |
| **지속적인 패치** | **버그 수정** | **밸런스 패치** |
| **멀티플레이** | **PVP 모드** | **PVE 협동 모드** |

1. **추가 확장팩 개발: 초기 출시 후 더 많은 스테이지와 몬스터, 그리고 새로운 전투 메커니즘을 추가하는 확장팩을 계획하여 플레이어들이 지속적으로 게임에 참여할 수 있도록 유도합니다.**
2. **커뮤니티 피드백 반영: 플레이어들로부터 받은 피드백을 적극 반영하여 게임성 개선 및 신규 콘텐츠 추가를 고려해야 합니다. 이를 통해 사용자 만족도를 높이고 커뮤니티 형성에 기여할 수 있습니다.**
3. **온라인 기능 도입 검토: 장기적으로 온라인 협동 모드나 PVP 모드를 추가하여 멀티플레이어로 확장할 가능성을 검토할 수 있습니다. 이는 게임의 수명 연장 및 플레이어 수 증가에 긍정적인 영향을 미칠 수 있습니다.**

**<텀 게임즈 스튜디오>**

**- 신윤섭 202003396**

**- 홍성진 202103653**

**- 임서현 202301458**

**- 김민성 202301991**