**캡스톤 디자인**

최종 보고서

|  |
| --- |
| VR FPS |

**2019.12.13**

17013141 봉수연

16013083 이주형

16013096 임민규

16011185 송제윤

- 목 차 -

**Ⅰ 개요 ................................................................. 4**

1. 개발배경 ............................................................................... 4

2. 목표설정 및 개발 기간............................................................................... 5

3. 기대효과 및 마케팅............................................................................... 5

**Ⅱ 산출물 체계 ................................................................. 6**

**Ⅲ 산출물 양식 ................................................................. 7**

**1. 분석 단계** ............................................................................................... 7

1.1사용자 요구사항 정의서 ............................................................................... 7

1.2 유스케이스 명세서 ............................................................................... 9

1.3 요구사항 추적표 ............................................................................... 23

**2. 설계 단계** ............................................................................... 24

2.1설계클라스 목록 ............................................................................... 25

2.2 시퀀스도 ............................................................................... 26

2.3 설계 클래스도 ............................................................................... 33

2.4 설계 클래스 정의............................................................................... 38

**3. 구현 단계** ............................................................................... 42

3.1맵 구현 ............................................................................... 42

3.2 아이템 구현 ............................................................................... 44

3.3 캐릭터 구현 ............................................................................... 48

3.3.1 도둑 캐릭터................................................................................................................................ 48

3.3.2 헌터 캐릭터................................................................................................................................ 50

3.4 interaction 구현 ............................................................................... 54

3.5 네트워크 구현 ............................................................................... 56

3.6 UI구현 ............................................................................... 59

3.7 게임 scene 변환 구현 ............................................................................... 62

**4. 시험 단계** ............................................................................... 63

4.1 사용자 지침서............................................................................... 63

4.2 운영자 지침서 ............................................................................... 65

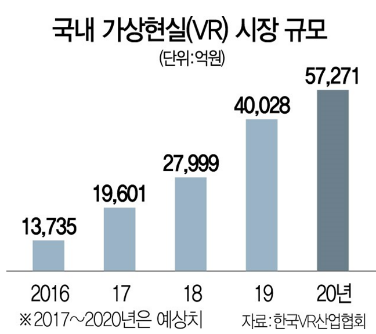
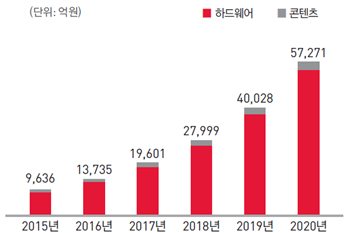
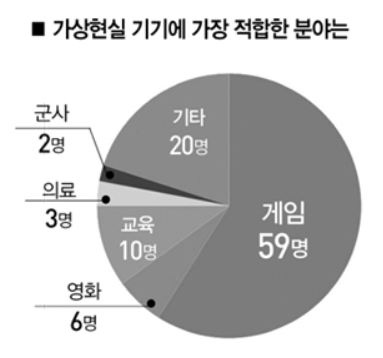
- 오큘러스 세팅 ............................................................................... 65

- 빌드 및 연동 ............................................................................... 68

4.3 시스템 시험............................................................................... 73

**Ⅰ 개요**

**1. 개발 배경**



자료 : 헤럴드경제

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjKx5uV0KrmAhVaeXAKHYBGDUwQjRx6BAgBEAQ&url=http%3A%2F%2Fwww.gametrics.com%2Fnews%2FNews02_View.aspx%3Fseqid%3D35112&psig=AOvVaw0KwGoBbHJupSXQoDBt1itm&ust=1576051666620504)다양한 VR HMD (Virtual Reality Head Mounted Display) 기기들이 앞다투어 등장했고, 그에 따라 수많은 VR 게임 컨텐츠들이 쏟아져 나왔습니다. 이 중에서 FPS(First Person Shooting) 장르는 VR기기에서 구현될 경우, 가장 크게 현실감을 느낄 수 있는 장르라고 여겨져 왔습니다. 실제로 VR HMD를 착용했을 때, 사용자는 VR HMD 내에 자신의 시야와 유사한 화면을 제공받을 수 있으므로 현실감이 매우 중요시되는 게임 장르 중 하나인, FPS 장르에 대한 잠재 수요가 매우 컸습니다.

또한 기존의 콘솔게임에서 FPS는 상당한 사랑을 받고 있는 게임 장르 중 하나입니다. 대표적으로 배틀 그라운드, 오버워치를 예를 들 수 있습니다. 이들의 유저는 작년 기준으로 전세계 이용자 수가 4억명, 4000만명을 넘어설 정도라고 합니다. 앞으로 콘솔게임이 고퀄리티 그래픽으로 VR시장이 더 발전하여 VR컨텐츠 시장 또한 활성화됨에 따라, 이러한 기존 FPS 콘솔 유저들이 자연스럽게 대거 유입 될 수 있을 것이라고 예상합니다.

따라서 저희는 게임 장르 중 제일 인기가 많은 장르 중 하나인 FPS를 저희만의 게임으로 만들어 보기로 했습니다. 평소에 사람들이 즐겨보는 영화, 드라마, 게임 등 다양한 매체에서 공통적으로 많은 사랑과 관심을 받는 장르인 공포(horror)에서 영감을 받아 공포를 컨셉으로 단순한 평면 스크린이나 모니터를 통한 감상이나 체험을 뛰어넘어 3D로 생동감 있게 공포를 즐길 수 있다면 많은 사랑을 받을 수 있겠다는 생각을 시작으로 본 프로젝트를 진행했습니다.

**2. 목표설정**



저희가 모티브로 따온 게임은 소위 ‘데바데’로 불리는 ‘DAY BY DEADLIGHT’ 게임과 모바일게임 제 ‘5인격’입니다. 이 두개의 게임은 순위권에 오를 정도로 인기가 있는 게임이라 이것을 바탕으로 FPS 요소를 추가하였습니다. 또한 어렸을 때 했던 게임처럼, 저희 게임만의 특색은 기존의 FPS게임 뿐만 아니라 아케이드적인 요소를 추가한 것에 있습니다. 슈팅 및 액션을 좋아하는 캐릭터인 헌터와 아케이드를 좋아하는 도둑을 선택하여 사용자들의 친숙한 상자잡기 아이템 습득 등을 통해 보다 폭넓은 사람들의 사랑을 받을 수 있는 게임으로 기획했습니다. 그리고 혼자만 즐기는 1인칭 슈팅게임이 아닌 실제 캐릭터가 되어 와이파이만 있다면, 어디서든 친구들과 함께 VR게임을 즐길 수 있도록 네트워크를 추가하는 것을 계획했습니다.

저희가 만들 게임의 이름은 ‘The thieves’로, 장르는 생존멀티플레이 비대칭 게임으로 pun2를 이용하여 네트워크 기반 멀티플레이를 게임으로 플레이 시간은 10-15분입니다.

소프트웨어 적인 부분은Unity 3D 엔진을 사용하여 개발하였고, HMD는 Oculus Quest를 사용합니다.

**3. 기대효과 및 마케팅**

데모영상 및 게임하는 영상 등을 유튜브, 트위치와 같은 게임 유저들이 많이 보는 플랫폼에 배포하여 인지도를 향상시키는 것을 목표로 마케팅 합니다. 실제로 오큘러스로 VR을 즐기는 사람들에게 오큘러스 스토어에 무료 게임으로 등록해 놓으므로 VR시장에서 FPS게임의 선호도가 높은 것에 대비하여 실제 FPS게임 시장점유율은 비교적 높지 않고 유료화가 많다는 점을 이용하여 ‘The Thieves’는 프리마케팅 전략을 사용합니다. 게임을 이용한 사용자들에게 기존에 있는 컨텐츠들과 다른 저희만의 차별성을 어필할 수 있습니다. 본 프로젝트 팀에 대한 인지도를 향상시키는 것을 목표로 하며, 게임은 무료로 제공하고, 게임을 위한 아이템 구입은 유료로 제공하는 등 차후에 진행하게 될 부분적 유료 컨텐츠에 대한 마케팅 기대효과를 확보할 것입니다.

**Ⅱ 산출물 체계**

|  |  |  |
| --- | --- | --- |
| 단계 | 코드 | 산출물 |
| 분석 | R1 | 사용자 요구사항 정의서 |
| R2 | 유스케이스 명세서 |
| R3 | 요구사항 추적표 |
| 설계 | D1 | 클래스 설계서 |
| 구현 | I1 | 맵 구현 |
| I2 | 아이템 구현 |
| I3 | 캐릭터 구현 |
| I4 | interaction 구현 |
| I5 | 네트워크 구현 |
| I6 | UI 구현 |
| I7 | 게임 scene 변환 구현 |
| 시험 | T1 | 사용자 지침서 |
| T2 | 운영자 지침서 |

산출물 ID의 기본적인 부여 방법은 다음과 같이 제시한다. 단, 프로젝트에

별도의 산출물 ID 부여 방법이 존재할 경우는 프로젝트에서 정하는 바에 따라 사용할 수 있다.

프로젝트 ID + 분류 ID + 산출물 코드 + 순서번호

【산출물 ID 부여】

프로젝트 ID: SS

요구사항 ID: G(게임), C(일반), S(시스템), D(개발), P(프로젝트)

분류 ID: SUB-서브 시스템, USD – 유스케이스 다이어그램, UC – 유스케이스

DCD: 시퀀스도 SCD: 설계 클래스도

인 경우 산출물 ID는 다음과 같이 부여될 수 있다.

**Ⅲ. 산출물 양식**

**1. 분석단계**

**1.1 사용자요구사항 정의서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R1** | **사용자 요구사항 정의서** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 분석 | 작성일자 | 2019.10.06 | 버전 | V1.0 |

**1.1.1 기능적 요구사항**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명 | 구분 | 요구사항 설명 | 제약사항 | 해결방안 |
| C001 | 제한시간 | 유저(공통) | 제한시간인 10분을 나타내는 타이머 | x |  |
| C002 | 이동 | 유저(공통) | 휠을 이용해서 움직일 수 있는 기능 | x |  |
| C003 | 슈팅 | 유저(공통) | 버튼을 누를 경우 총을 발사하는 기능 | x |  |
| G001 | 상자 줍기 | 도둑 | 상자 쪽으로 조정해서 버튼을 이용하여 상자를 여는 기능 | 도둑만 가능 | 유저(공통) 클래스를 상속 |
| G002 | 아이템 줍기 | 도둑 | 아이템을 집고 해당 아이템 표시해주는 기능 | 도둑만 가능 | 유저(공통) 클래스를 상속 |
| G003 | 기절 | 도둑 | 술래에게 총 2방을 맞았을 경우, 기절(모든 키 제한)하게 된다. | 도둑만 가능 | 유저(공통) 클래스를 상속 |
| G004 | 도둑 잡기 | 헌터 | 술래가 버튼을 때리는 기능 | 헌터만 가능 | 유저(공통) 클래스를 상속 |
| G005 | 상자 | 아이템 | 상자를 열었을 때 뭐가 나올지 결정하는 기능과 술래에게 상자의 위치가 표시되는 기능(보물 상자 제외) | 도둑만 가능 |  |
| G006 | 독가스 | 아이템 | 상자가 열었을 때 독가스일 경우 도둑이 스턴에 걸리게 되는 기능 | 도둑만 가능 |  |
| G007 | 소생 총알 | 아이템 | 소생 총알을 사용시, 도둑이 풀려나는 기능 | 도둑만 가능 |  |
| G008 | 스턴 총알 | 아이템 | 스턴총알을 헌터에게 맞출 경우, 스턴에 걸리게 하는 기능 | 도둑만 가능 |  |
| G009 | 보물상자 | 아이템 | 보물 상자를 집었을 때 보물상자 위치가 발각되는 기능 | 도둑만 가능 |  |
| G010 | 음향효과 | 아이템 | 상황별 음향 효과 기능 | X |  |
| G011 | 피격 기능 | 유저(공통) | 총알을 맞았을 때 속성값 변화기능 | X |  |

**1.1.2 비기능적 요구사항**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 요구사항 ID | 요구사항 명 | 구분 | 요구사항 설명 | 제약사항 | 해결방안 |
| S001 | 성능 | 시스템 | 저사양 디바이스를 위한 성능 제한 | x |  |
| S002 | 용량 | 시스템 | 1GB를 넘지 않는 용량 제한 | x |  |
| S003 | 품질 | 시스템 | 3D에서 느낄 수 있는 어지러움증 개선 | x |  |
| P001 | 일정관리 | 프로젝트 | 정해진 기간안에 프로젝트가 완성 |  |  |
| D001 | H/W | 개발 | 게임을 개발하기 위한 하드웨어 요구사항 |  |  |
| D002 | S/W | 개발 | 게임을 개발하기 위한 소프트웨어 요구사항 |  |  |

**1.2 유스케이스 명세서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R2** | **유스케이스 명세서** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 분석 | 작성일자 | 2019.10.06 | 버전 | V1.0 |

**1.2.1서브시스템 목록**

|  |  |  |
| --- | --- | --- |
| 서브시스템 ID | 서브시스템명 | 서브시스템 설명 |
| SS\_SUB\_00\_01 | Grab\_box | 도둑만 상자를 잡을 수 있고 그랩 버튼 사용시 상자는 사라진다. |
| SS\_SUB\_00\_02 | Grab\_item\_01~04 | 도둑만 아이템을 잡을 수 있다. |
| SS\_SUB\_00\_03 | shot\_thief\_01 | 도둑만 쏠 수 있으며 헌터를 맞춘다. |
| SS\_SUB\_00\_04 | shot\_thief\_02 | 도둑만 쏠 수 있으며 기절 도둑을 맞춘다. |
| SS\_SUB\_00\_05 | shot\_hunter\_01 | 헌터만 쏠 수 있으며 도둑을 맞춘다. |
| SS\_SUB\_00\_06 | Trajectory\_stun | 맞은 액터 (헌터, 도둑)는 기능키 제한(5초간)된다. |
| SS\_SUB\_00\_07 | Trajectory\_recover | 목숨이 없는 도둑만 소생할 수 있다. |
| SS\_SUB\_00\_08 | hit\_hunter | 헌터만 때릴 수 있으며 도둑을 때린다. |
| SS\_SUB\_00\_09 | beat\_thief | 도둑만 발생하며 헌터가 때릴 때 목숨이 줄어든다. |
| SS\_SUB\_00\_10 | Shock | 도둑이 헌터에게 2번 맞고 모든(제한시간&시선 제외)키가 제한된다. |
| SS\_SUB\_00\_11 | shot\_possible | 도둑, 헌터 모두 사용 가능하며 방아쇠 버튼 시 탄환의 여부와 종류를 판단한다. |
| SS\_SUB\_00\_12 | bullet\_diminish | 총 사용 후 탄의 개수를 감소시킨다. |
| SS\_SUB\_00\_13 | trajectory\_appear | 탄의 종류에 따라 총구 앞에 스턴 궤적obj와 소생 obj를 생성시킨다. |
| SS\_SUB\_00\_14 | character\_move | 도둑, 헌터의 움직임 정보 |
| SS\_SUB\_00\_15 | character\_hunter | 헌터의 기본값 정보 |
| SS\_SUB\_00\_16 | character\_thief | 도둑의 기본값 정보 |
| SS\_SUB\_00\_16 | limit\_time | 제한시간 정보 |

**1.2.2 유스케이스 다이어그램(UCD)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 유스케이스 ID | 유스케이스명 | 유스케이스 설명 | 관련액터ID | 관련 SUB ID |
| SS\_USD\_00\_01 | Grab | 도둑이 상자를 열 때와 그 안의 아이템과의 관계 | Thief | SS\_SUB\_00\_01  SS\_SUB\_00\_02 |
| SS\_USD\_00\_02 | Shot | 도둑 과 헌터가 슈팅을 하였을 때의 관계 | Thief,hunter | SS\_SUB\_00\_03  SS\_SUB\_00\_04  SS\_SUB\_00\_05  SS\_SUB\_00\_11  SS\_SUB\_00\_12 |
| SS\_USD\_00\_03 | Shot\_effect | 도둑이 총을 쏠 때 어떤 종류의 총알인지의 따른 관계 | Thief,Hunter | SS\_SUB\_00\_06  SS\_SUB\_00\_07  SS\_SUB\_00\_13 |
| SS\_USD\_00\_04 | Hit | 헌터가 도둑을 때렸을 때의 관계 | Thief,Hunter | SS\_SUB\_00\_08  SS\_SUB\_00\_09  SS\_SUB\_00\_10 |
| SS\_USD\_00\_05 | Character | 도둑, 헌터의 기본값 정보 관계 | Thief,Hunter | SS\_SUB\_00\_14  SS\_SUB\_00\_15 |
| SS\_USD\_00\_06 | limit\_time\_gameover | 제한시간과 게임종료의 관계 | End\_system | SS\_SUB\_00\_16 |

1

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_01 | UCD 명 | Grab |
| 관련 서브시스템 ID | SS\_SUB\_00\_01  SS\_SUB\_00\_02 | 관련 서브시스템명 | Grab\_box  Grab\_item\_01~04 |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-01 | 유스케이스명 | 아이템 줍기 |
| **1. 주요 액터**  - 도둑, 시스템  **2. 이해관계자와 관심사항**  - 그랩 버튼을 눌렀을 때, 상자 obj가 사라져야 한다.  - 상자 obj가 사라지면서 item obj가 생성되어야 한다.  - 독가스가 생성될 시 도둑이 스턴에 걸려야 한다.  - 생성된 obj에 그랩 버튼을 누르면, 아이템이 주워진 걸로 처리된다.  **3. 전제조건**  - 상자 obj나 아이템 obj에 그랩 버튼을 누르면 이벤트가 발생해야 된다.  **4. 종료조건**  - 아이템에 obj에 그랩 버튼을 눌렀을 때 이벤트가 발생하고, 발생했다는 게  시스템에 전달되어야 한다  **5. 기본 시나리오**  1) 도둑이 지나가다가 상자를 발견한다.  2) 상자 근처로 가서 그랩 버튼을 사용한다.  3) 이벤트가 발생하면서, 상자가 사라진다.  4) 상자가 사라진 자리에 아이템 obj가 생성된다.  5) 독가스의 경우 도둑은 5초간 스턴에 걸린다.  6) 다른 아이템의 경우에는 도둑은 아이템 근처로 가서 그랩을 할 수 있다.  7) 그랩 버튼을 누르면, 눌렀다는게 시스템에 전송되고 아이템을 주운걸로 처리한다.  8) 아이템이 사라진다.  **6. 대안 시나리오**  1) 헌터가 상자에 그랩 버튼을 사용할 시에는 아무런 이벤트가 발생하지 않는  다.  2) 도둑이 나온 아이템에 그랩 버튼을 안 사용할 수도 있다. 7. 구현 시 고려사항 - Obj에 그랩 버튼의 작동유무  - 시스템에서 변경된 정보가 obj에 제대로 재현되는가  - 독가스가 나올 경우, 즉발적으로 스턴에 걸리는가 8. 발생 빈도 - 게임내에 배치된 상자의 개수 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_02 | UCD 명 | shot |
| 관련 서브시스템 ID | SS\_SUB\_00\_03  SS\_SUB\_00\_04  SS\_SUB\_00\_05  SS\_SUB\_00\_11  SS\_SUB\_00\_12 | 관련 서브시스템명 | shot\_thief\_01  shot\_thief\_02  shot\_hunter\_01  shot\_possible  bullet\_diminish |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-02 | 유스케이스명 | 총 맞은 후 효과 |
| **1. 주요 엑터**  - 도둑, 헌터  **2. 이해관계자와 관심사항**  - 슈팅하는 버튼을 누르면 발생 여부에 따라 이벤트 진행  - 총알이 발사되면서 궤적도 obj 생성.  - 총알이 발사되면서 애니메이션 효과 생성  - 총알이 발사되면서 슈팅 effect 생성  - 총알이 발사되면서 해당 액터의 총알 개수 감소  **3. 전제조건**  - 버튼을 누르면 이벤트가 발생해야 된다.  **4. 종료조건**  - 총을 쏜 후 총알의 개수가 줄어든 것이 헌터나 도둑의 정보에 반영되어야 한다.  **5. 기본 시나리오**  1) 헌터나 도둑이 상대를 향해 겨냥하고 버튼을 누른다.  2) 발생여부에 따라, 총알이 없으면 아무런 이벤트가 발생하지 않는다.  3) 총알이 있을 경우, 총알이 발사되면서 애니메이션 효과 발생  4) 궤적도 obj가 생성된다.  5) 슈팅 effect 생성된다.  6) 해당 액터의 총알 개수가 감소한다.  **6. 대안시나리오**  **7. 구현 시 고려사항**  - 버튼의 작동유무  - 총을 쏘고 난 후에 총알에 개수 같은 정보가 바로 헌터나 도둑에게  적용되는가.  **8. 발생 빈도**  - 총을 쏠 때마다 발생 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_03 | UCD 명 | Shot\_effect |
| 관련 서브시스템 ID | SS\_SUB\_00\_06  SS\_SUB\_00\_07  SS\_SUB\_00\_13 | 관련 서브시스템명 | Trajectory\_stun  Trajectory\_recover  Trajectory\_appear |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-03 | 유스케이스명 | 총 맞은 후 효과 |
| **1. 주요 액터**  - 궤적도 obj, 총에 맞는 obj  **2. 이해관계자와 관심사항**  - 궤적도 obj랑 총에 맞은 obj 충돌 여부 확인.  - 스턴 총알에 충돌했을 경우 해당 obj 5초간 스턴.  - 소생 총알과 충돌했을 경우 해당 obj의 기능 제한 해제.  **3. 전제조건**  - Obj와 obj의 충돌함수가 구현되어 있어야한다.  **4. 종료조건**  - 총알에 맞은 후 나타나는 효과가 총알에 맞은 obj에 구현되면 종료.  **5. 기본 시나리오**  1) 궤적도 obj가 플레이어 obj에 충돌함.  2) 발사된 총알이 무슨 총알인지 여부를 따짐.  3) 스턴 총알일 경우에는 해당 obj 기능을 5초간 제한.  4) 소생 총알일 경우 해당 obj 목숨 값을 확인.  5) 목숨 값이 0일때는 맞은 obj에 기능키제한 해제  **6.대안시나리오**  - 발사된 궤적도 obj가 플레이어 obj와 충돌하지 않을 경우에는 아무런  일도 발생하지 않는다  - 소생 총알을 맞았을 때 기절 상태가 아닐 경우에는 아무런 효과가 일어나지 않는다.  **7. 구현 시 고려사항**  - 궤적도 obj의 길이가 어느 정도까지 구현해야 할지 고려해야 된다.  - 총을 쏘고 나서 생겨난 궤적도 obj가 몇 초뒤에 사라질지도 고려해야 된다.  **8. 발생 빈도**  - 총을 쏠 때마다 발생 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_04 | UCD 명 | Hit |
| 관련 서브시스템 ID | SS\_SUB\_00\_08  SS\_SUB\_00\_09  SS\_SUB\_00\_10 | 관련 서브시스템명 | hit\_hunter  beat\_thief  Shock |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-04 | 유스케이스명 | 때리기 |
| **1. 주요 액터**  - 헌터, 도둑  **2. 이해관계자와 관심사항**  - 헌터가 도둑 근처에 있을 때, 그랩 버튼을 누르면 이벤트 발생.  - 이벤트 발생으로 헌터 hit 애니메이션  - 도둑 obj랑 충돌했는지 여부.  - 맞았을 경우 도둑 obj 체력 감소.  - 도둑의 목숨이 0이되면 도둑 obj의 모든 기능키 제한.  **3. 전제조건**  - Obj와 obj의 충돌함수가 구현되어 있어야한다.  **4. 종료조건**  - 도둑이 맞은 다음, 감소된 목숨 값이 도둑 obj에 전달되면 종료.  **5. 기본 시나리오**  1) 헌터가 도둑을 쫓다가 때리면 닿을 거리가 되면 그랩 버튼을 누른다.  2) 때리는 애니메이션이 구현된다.  3) 헌터 딜레이가 발생한다.  4) 만약 때리는 모션과 도둑 obj가 충돌했을 경우 맞은 도둑 obj 체려감소.  5) 목숨 값이 0일때는 맞은 도둑 obj의 기능 제한  **6. 대안 시나리오**  - 헌터가 도둑을 못 맞추면 아무런 일도 일어나지 않는다.  **7. 구현 시 고려사항**  - 헌터 hit 애니메이션을 구현할 때 히트박스의 영역을 고려해봐야 된다.  **8. 발생 빈도**  - 헌터가 hit 버튼을 누를 때마다 발생. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_05 | UCD 명 | character |
| 관련 서브시스템 ID | SS\_SUB\_00\_14  SS\_SUB\_00\_15  SS\_SUB\_00\_16 | 관련 서브시스템명 | character\_move  character\_hunter  character\_thief |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-05 | 유스케이스명 | 캐릭터 |
| **1. 주요 액터**  - 헌터, 도둑  **2. 이해관계자와 관심사항**  - 각 obj는 공통적으로 총알에 대한 정보를 가지고 있어야한다.  - 도둑과 헌터의 이동속도가 달라야한다.  - 맞았을 경우 도둑 obj 목숨이 감소해야한다.  - 도둑의 목숨이 0이되면 도둑 obj의 모든 기능키 제한.  **3. 전제조건**  - 도둑과 헌터에 해당하는 obj가 개별적으로 구현되어 있어야한다.  **4. 종료조건**  - 게임이 종료되면 각 obj는 기능을 종료한다.  **5. 기본 시나리오**  1) 헌터가 도둑을 쫓다가 때리면 닿을 거리가 되면 그랩 버튼을 누른다.  2) 도둑이 맞았을 경우 도둑 obj내에 목숨 값이 감소한다.  3) 기절이 되었을 경우 도둑 obj 내에 이동속도는 0이 된다.  4) 총을 사용시 도둑, 헌터 obj내에 총알에 대한 정보가 변경된다.  5) 도둑 obj의 목숨 값이 0일경우 소생 총알을 통하여 값이 1 증가한다.  **6. 대안 시나리오**  **7. 구현 시 고려사항**  - 도둑 obj의 목숨 값이 0이 아닐 때에 소생 총알을 통한 목숨 값 증가를 제한한다.  **8. 발생 빈도**  - 총을 쏘거나 hit가 발생하면 일어난다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_USD\_00\_06 | UCD 명 | limit\_time\_gameover |
| 관련 서브시스템 ID | SS\_SUB\_00\_16 | 관련 서브시스템명 | limit\_time |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스ID | SS\_UC\_00-06 | 유스케이스명 | 시간제한 |
| **1. 주요 액터**  - 헌터, 도둑  **2. 이해관계자와 관심사항**  - 제한시간이 0이 되면 도둑과 헌터의 obj내에 이동속도가 0이된다.  - 모든 동작을 제한하고 나면 end\_system에 돌입하게 된다.  **3. 전제조건**  - 게임이 종료되기 위하여 제한시간을 이용한 종료함수가 필요하다.  **4. 종료조건**  - 제한시간이 0이 되면 종료  **5. 기본 시나리오**  1) 도둑들의 목숨 값이 0이 되기전까지 제한시간은 0으로 수렴한다.  2) 제한시간이 0이되는 순간 플레이어 obj들의 이동속도는 모두 0이된다.  3) 게임 end\_system을 호출하여 게임을 정상적으로 종료한다.  **6. 대안 시나리오**  **7. 구현 시 고려사항**  - 제한시간이 남아있어도 도둑들의 모든 obj의 목숨 값이 0 이되면 종료되어야한다.  **8. 발생 빈도**  - 게임에 정해진 시간이 되었을 때 1번 발생 | | | |

**1.2.3 액터 목록**

|  |  |  |  |
| --- | --- | --- | --- |
| 액터 ID | 액터명 | 액터유형 | 액터설명 |
| End\_system | 게임종료시스템 | 보조 | 게임이 조건에 달하면, 게임을 종료시킨다. |
| Time | 시계 | 보조 | 10분으로 시작하여 0분시  End\_system 를 부른다. |
| Thief | 도둑 플레이어 | 주요 | 게임내에서 도둑을 플레이하는 사람  (목숨 값, 스피드, 총알 유무, 총알 종류, 보물 보유) |
| Hunter | 헌터 플레이어 | 주요 | 게임내에서 헌터를 플레이하는 사람  (스피드, 탄 개수) |
| Trajectory  \_stun\_obj | 스턴총알의 궤적 | 주요 | Shot\_effect를 주는 object |
| Trajectory  \_recovery\_obj | 소생총알의 궤적 | 주요 | Shot\_effect를 주는 object |

**1.2.4 통합 유스케이스 다이어그램**

|  |
| --- |
| **텍스트, 지도이(가) 표시된 사진  자동 생성된 설명** |

**1.3 요구사항 추적도**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R3** | **요구사항 추적도** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 분석 | 작성일자 | 2019.10.06 | 버전 | V1.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **분석단계** | | | |
| **사용자 요구사항 명세서** | | **유스케이스 모형 기술서** | |
| **요구사항 ID** | **요구사항 명** | **유스케이스 ID** | **유스케이스 명** |
| G001, G002 | 상자열기, 아이템 줍기 | SS\_USD\_00\_01 | Grab |
| C003 | 슈팅 | SS\_USD\_00\_02 | Shot |
| G007, G008, G011 | 소생, 스턴, 피격기능 | SS\_USD\_00\_03 | Shot\_effect |
| G003, G004 | 기절, 도둑 잡기 | SS\_USD\_00\_04 | Hit |
| C002 | 이동 | SS\_USD\_00\_05 | character |
| C001 | 제한시간 | SS\_USD\_00\_06 | limit\_time\_gameover |

**2. 설계단계**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D1** | **클래스 설계서** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 설계 | 작성일자 | 2019.10.06 | 버전 | V1.0 |

**2.1 설계 클래스 목록**

|  |  |  |
| --- | --- | --- |
| **설계 클래스 ID** | **설계 클래스명** | **관련 유스케이스ID** |
| **SS\_DCD\_00\_01** | **도둑 기절 정보** | SS\_USD\_00\_02  SS\_USD\_00\_04 |
| **SS\_DCD\_00\_02** | **히트 정보** | SS\_USD\_00\_04 |
| **SS\_DCD\_00\_03** | **독가스 상자 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_04** | **보물 상자 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_05** | **상자 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_06** | **상자 줍기 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_07** | **소생 총알 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_08** | **스턴 총알 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_09** | **아이템 줍기 정보** | SS\_USD\_00\_01 |
| **SS\_DCD\_00\_10** | **이동 정보** | SS\_USD\_00\_05 |
| **SS\_DCD\_00\_11** | **제한시간 정보** | SS\_USD\_00\_06 |
| **SS\_DCD\_00\_12** | **총알 슈팅 정보** | SS\_USD\_00\_02  SS\_USD\_00\_03 |
| **SS\_DCD\_00\_13** | **캐릭터 정보** | SS\_USD\_00\_05 |

**2.2 시퀀스도**

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_01** | **시퀀스도명** | **도둑 기절 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_02, SS\_USD\_00\_04** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_02** | **시퀀스도명** | **히트 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_04** | | |
| **주요 액터** | **hunter, thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_03** | **시퀀스도명** | **독가스 상자 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_04** | **시퀀스도명** | **보물 상자 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief, hunter** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_05** | **시퀀스도명** | **상자 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_06** | **시퀀스도명** | **상자 줍기 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_07** | **시퀀스도명** | **소생 총알 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_08** | **시퀀스도명** | **스턴 총알 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief, hunter,**  **Trajectory\_stun\_obj** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_09** | **시퀀스도명** | **아이템 줍기 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01** | | |
| **주요 액터** | **thief** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_10** | **시퀀스도명** | **이동 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_05** | | |
| **주요 액터** | **thief, hunter** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_11** | **시퀀스도명** | **제한시간 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_06** | | |
| **주요 액터** | **thief,hunter,End\_system** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_12** | **시퀀스도명** | **총알 슈팅 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_02, SS\_USD\_00\_03** | | |
| **주요 액터** | **thief, hunter** | **주요 클래스** |  |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **시퀀스도 ID** | **SS\_DCD\_00\_13** | **시퀀스도명** | **캐릭터 정보** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_05** | | |
| **주요 액터** | **thief, hunter** | **주요 클래스** |  |
|  | | | |

**2.3 설계 클래스도**

|  |  |  |  |
| --- | --- | --- | --- |
| **설계 클래스도 ID** | **SS\_SCD\_00\_01** | **설계 클래스도명** | **도둑 클래스** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_07~10, SS\_USD\_00\_12, SS\_USD\_00\_13,** | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **설계 클래스도 ID** | **SS\_SCD\_00\_02** | **설계 클래스도명** | **헌터 클래스** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_01~02, SS\_USD\_00\_08, SS\_USD\_00\_10, SS\_USD\_00\_12** | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **설계 클래스도 ID** | **SS\_SCD\_00\_03** | **설계 클래스도명** | **총알 클래스** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_05~09** | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **설계 클래스도 ID** | **SS\_SCD\_00\_04** | **설계 클래스도명** | **씬전환 클래스** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_11** | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **설계 클래스도 ID** | **SS\_SCD\_00\_05** | **설계 클래스도명** | **상자 클래스** |
| **관련 유스케이스 ID** | **SS\_USD\_00\_03~09** | | |
|  | | | |

**2.4 설계 클래스 정의**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 id | SS\_DCD\_001\_01\_01 | | 클래스명 | 도둑 클래스 |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| Thief\_move | public |  | N/A | 도둑의 기본정보와 변화 값 저장 |
| Thief\_Collider | public |  | N/A | 도둑의 접촉시 정보 변경 |
| Thief\_Fire | private |  | N/A | 도둑의 총구에서 총알 생성 |
| Thief\_UI | public |  | N/A | 도둑의 총알과 목숨개수를 UI표시 |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Move\_update() | public | Movespeed  Turnspeed  Stun  Recover  Life | public | 도둑의 기본 정보값 저장 |
| Hit() | public | heart | public | 도둑의 목숨을 1줄인다. |
| Stun() | public | Movespeed  tunspeed | public | 도둑의 움직임을 5초 정지시킨다. |
| Recoverplus() | public | recover | public | 소생총알 개수를 1증가시킨다. |
| Stunplus() | public | stun | public | 스턴총알의 개수를 1증가시킨다. |
| Thief\_Ui\_update | public | Life  Stun  recover | public | 도둑의 총알 개수와 목숨개수를 캠버스에 표시한다. |
| Fire\_update | public | Stun  recover | public | 도둑의 총알 개수를 1줄이고 총구앞에 총알을 생성시킨다. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 id | SS\_DCD\_001\_01\_02 | | 클래스명 | 헌터 클래스 |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| Hunter\_move | public |  | N/A | 헌터의 기본정보와 변화 값 저장 |
| Hunter\_Collider | public |  | N/A | 헌터의 접촉시 정보 변경 |
| Hunter\_Fire | private |  | N/A | 헌터의 총구에서 총알 생성 |
| Hunter\_UI | public |  | N/A | 헌터의 총알과 목숨개수를 UI표시 |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Move\_update() | public | Movespeed  Turnspeed  Stun  Recover  Life | public | 헌터의 기본 정보값 저장 |
| Hit\_stun() | public | Movespeed  Turnspeed | public | 헌터가 도둑을 때릴시 헌터의 움직임을 5초 정지시킨다. |
| Stun() | public | Movespeed  tunspeed | public | 헌터의 움직임을 5초 정지시킨다. |
| Hunter\_Ui\_update | public | Stun  recover | public | 헌터의 총알 개수와 목숨개수를 캠버스에 표시한다. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 id | SS\_DCD\_001\_01\_03 | | 클래스명 | 총알 클래스 |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| Recover\_bullet | public |  | N/A | 소생총알을 도둑이 접속시 도둑소생 개수 1증가 |
| Stun\_bullet | public |  | N/A | 스턴총알을 도둑이 접속시 도둑스턴 개수 1증가 |
| disappear | private |  | N/A | 총알이 도둑과의 접속시 사라지고 효과 생성 |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Recover\_update() | public | Recover | public | 도둑의 소생총알 개수 증가 |
| Stun\_update() | public | Stun | public | 도둑의 스턴총알 개수 증가. |
| Disappear\_update() | public | R\_particle  S\_particle | public | 총알이 물체에 닿을 시 총알을 없애고 효과를 0.5초 생성후 삭제. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 id | SS\_DCD\_001\_01\_04 | | 클래스명 | 씬전환 클래스 |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| FirstScene | private |  | N/A | 첫화면 생성 |
| VideoScene | private |  | N/A | 비디오 화면 생성 |
| MainScene | private |  | N/A | 메인 화면 생성 |
| EndScene | private |  | N/A | 종료 화면 생성 |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| SceneChange() | private |  | void | 해당 씬 전환 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 클래스 id | SS\_DCD\_001\_01\_05 | | 클래스명 | 박스 클래스 |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| Box | private |  | N/A | 빈 상자 |
| Recover\_Box | private |  | N/A | 소생 총알 상자 |
| Stun\_Box | private |  | N/A | 스턴 총알 상자 |
| PoisonGas\_Box | private |  | N/A | 독가스 상자 |
| Treasure\_Box | private |  | N/A | 보물 상자 |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Disappear\_recover() | private | Recover | void | 상자를 없애고 소생총알 생성 |
| Disappear\_stun() | private | Stun | void | 상자를 없애고 스턴총알 생성 |
| Thief\_stun | private | Thief | void | 독가스 상자를 없애고 독가스를 3초생성과 동시에 도둑의 이동값 제어 |

**3.구현단계**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I1** | **맵 구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 구현 | 작성일자 | 2019.12.09 | 버전 | v1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 충돌감지 및 성능 최적화 | | |
| 관련 유스케이스 ID | | - | | |
| 구현 설명 | | 게임에 필요한 Map에 필요한 setting값을 넣어줍니다. | | |
| 프로그램 파일 | | | Inspector & setting  [Add Component] | 비고 |
| ID | 파일명(Assets) | |
| Map | Mega Fantasy Props Pack | | Mesh Renderer  Mesh Collider  Box collider  Rigidbody |  |
| Mapsky | UrbanNightSky | | Skybox |  |
| MapAudio | HorrorElements | | Audio Source |  |

2. 구현 모습

|  |
| --- |
|  |
| [그림 3.1] 맵 충돌 감지 값 적용 |
| 캐릭터들이 한정적으로 돌아다닐 맵의 사이즈를 결정하고 컴포넌트를 추가하여 각각의 성과 성벽, 집과 여러 아이템에 collider(충돌체)값과 rigidbody(물리력을 영향받도록 하는 컴포넌트)값에 mass(질량)과 gravity(중력)을 필요한 부분에 개별적으로 추가합니다. |

|  |
| --- |
|  |
| [그림 3.2] light bake 및 성능 최적화 |
| Directional Light만 mode를 Realtime두고 나머지 Point Light 및 Spot Light는 bake모드로 soft shadow를 만들어 주도록 한다. 이때 캐릭터를 제외한 모든 물체들은 Static으로 두어 bake할 수 있도록 한다. Realtime된 조명은 매 프레임마다 조명 연산이 되어 화면을 업데이트 하기 때문에 연산 부하가 크고 간접 광원을 사용하지 않아 그림자가 검은색으로 표현된다. 따라서 성능 최적화를 위해 대부분의 조명은 bake한다. bake된 조명은 런타임시 따로 연산을 하지 않아 표면의 조명과 그림자를 생성한다. |

|  |
| --- |
|  |
| [그림 3.3] 최종적 Map의 이미지 |
| 왼쪽그림은 전체 Map의 그림이고 오른쪽의 그림은 Map안에서의 분위기를 보여준다  Map 전체에 Audio Source가 추가되어 반복재생으로 무서운 BGM이 계속해서 들려온다. |

3. 고려사항

|  |
| --- |
| 그림1.1에서의 고려사항은 collider를 전부 설정해 주면 바닥과 문에도 충돌 값이 발생하여 진입하지 못하게 되기 때문에 바닥과 문이 있는 부분에는 is trigger를 설정해주어 충돌 값을 발생하지 않도록 설정해주거나 collider를 추가하지않아야 하므로 개별적으로 설정해주었다. 그림1.2에서의 고려사항은 성능 저하를 방지하기 위해 실시간 라이트 매핑 옵션을 비활성화 해야 하며 Lightmap Resolution(유닛 당 텍셀 값으로 라이트의 품질과 베이크시간이 비례한다)과 lIghtmap Size(라이트맵의 크기)를 하드웨어의 성능에 맞게 적절한 값을 찾아야했다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I2** | **아이템 구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 구현 | 작성일자 | 2019.12.09 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 상자 이벤트 발생 및 아이템 습득 & 총알 구현 | | |
| 관련 유스케이스 ID | | SS\_UC\_00\_01  SS\_USD\_00\_02 | | |
| 구현 설명 | | 도둑이 상자에 다가가 특정 키를 누르면 상자가 열리고 해당아이템이 나타난다. 2가지의 bullet아이템은 특정키를 눌러 습득이 가능하며 독가스 상자의 경우는 상자가 열림과 동시에 5초간 기능키가 제한된다.  헌터 혹은 도둑이 쏜 총알은 직선 방향으로 직진하며 다른 오브젝트를 맞을 경우 effect를 뿜으며 사라진다. | | |
| 서브시스템명 | 서브시스템 ID | 프로그램 파일 | | 비고 |
| ID | 파일명(Assets) |
| Grab\_box | SS\_SUB\_00\_01 | Box[prefab] | Treasure chest closed | Audio effect |
| Grab\_item | SS\_SUB\_00\_02 | stunBullet[prefab]  recoverBullet[prefab]  TreasureBox[prefab]  Unity Particle Pack 5.x | Ammunition pack (demo)  ChestFree |  |
| trajectory\_appear | SS\_SUB\_00\_13 | S\_bullet(prefabs) | Bullet(script) |  |

2. 프로그램 코드 및 알고리즘

|  |
| --- |
|  |
| [그림 3.4] BoxDisapper script |

BoxDisapper script를 상자에 넣어두고 도둑의 총에 ‘thief\_gun’ tag명을 붙여 준다.

이때, 도둑의 총이 상자에 다가가 F 키를 누르거나, 컨트롤러의 X,Y,A,B 버튼을 눌러준다면,

컴포넌트에 추가된audio가 실행되고 상자gameobject가 사라지게 된다.

그리고Instantiate함수를 이용하여 Gameobiect 컴포넌트에 추가한 해당 아이템이 나타나게 된다.

|  |
| --- |
|  |
| [그림 3.5] poison\_box script |

poison\_box script를 독가스상자에 넣어두고 도둑의 총에 ‘thief\_gun’ tag명을 붙여 준다.

이때, 도둑의 총이 상자에 다가가 F 키를 누르거나, 오큘러스 컨트롤러의 X,Y,A,B 버튼을 눌러준다면, 컴포넌트에 추가된audio가 실행되고 상자gameobject가 사라지게 된다.

그리고Instantiate함수를 이용하여 5초동안 gas event가 발생하게 되고 도둑의 thief\_move에 있는 gasstun함수가 발생하게 되어 5초간 기능키제한이 된다.

|  |
| --- |
|  |
|  |
| [그림 3.6] 총알 관련 [stun\_b script & recover\_b script] |

스턴 총알에 stun\_b script를, 회복 총알에 recover\_b script를 넣어주고 도둑의 총에 ‘thief\_gun’ tag명을 붙여 준다. 이때, 도둑의 총이 총알에 다가가 F 키를 누르거나, 컨트롤러의 X,Y,A,B 버튼을 눌러준다면, 해당 총알의 gameobject가 사라지게 된다. 그리고 도둑의 thief\_move에 있는 각 총알에 맞는 stunplus() recoverplus()함수가 해당 총알이 증가하게 된다.

|  |
| --- |
|  |
| [그림 3.7] 총알 생성 코드 [Hunter\_fire(script), Thief\_fire(script) ] |

Hunter\_fire(script)와 Thief\_fire(script)에서 firepos에 s\_bullet(prefabs)혹은 r\_bullet(prefabs)를 생성시킨다. 현재 총알은 bullet(script)를 가지고 있다.

|  |
| --- |
|  |
| [그림 3.8] 총알 직진과 이펙트 발생 코드 [Bullet(script)] |

Bullet(script)를 가지고 있는 총알은 생성과 즉시 vector3(총구가 가리키는 방향)으로 직진한다. 그후 어떠한 오브젝트를 만날 시 총알에 맞는 effect(prefabs)를 충돌위치에 0.5f만큼 생성 후 삭제한다. 그 후, 총알 오브젝트 또한 삭제한다.

**3. 구현 모습**

|  |
| --- |
|  |
| [그림 3.9] BOX에서 아이템이 생성된 모습 |

|  |
| --- |
|  |
| [그림 3.10] 독가스 상자 구현 |
| 독가스 상자의 경우 따로 총알과 보물이 들어있는 상자와는 별개로 만들어진 상자로, 열자 마자 effect효과와 함께 5초간 스턴이 발생하게 된다. |

|  |
| --- |
|  |
| [그림 3.11] 총알 생성 및 총알 발사 구현 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I3** | **캐릭터 구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 구현 | 작성일자 | 2019.12.21 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 도둑의 이동 및 슈팅 | | |
| 관련 유스케이스 ID | | SS\_USD\_00\_02  SS\_USD\_00\_05  SS\_USD\_00\_03 | | |
| 구현 설명 | | 도둑 캐릭터 기본정보 &  도둑이 총을 쏠 때의 작동구현 | | |
| 서브시스템명 | 서브시스템 ID | 프로그램 파일 | | 비고 |
| ID | 파일명(Assets) |
| character\_move | SS\_SUB\_00\_14 | Thief(prefabs) | Thief\_move(script) |  |
| trajectory\_appear | SS\_SUB\_00\_13 | Thief(prefabs) | Thief\_Fire(script) |  |
| bullet\_diminish | SS\_SUB\_00\_12 | Thief(prefabs) | Thief\_Fire(script) |  |

**1.1도둑 구현**

**1.2 프로그램 코드 및 알고리즘**

|  |
| --- |
|  |
| [그림 3.12]Thief의 기본 정보 코드 [Thief\_move(Script)] |

도둑의 정보:

1. moveSpeed : 도둑의 이동속도 변수 (5f)
2. turnSpeed : 도둑의 회전속도 변수 (350f)
3. stun : 도둑의 스턴총알 개수 변수 (5)
4. recover : 도둑의 회복총알 개수 변수 (0)
5. heart :도둑의 목숨의 개수 변수 (5)

도둑의 기본 최초 값은 start()함수에 정의 하고 update()함수에서는 게임 유저의 움직임 정보 값을 실시간으로 입력 받아 h,v변수에 기입 후 움직임을 제어한다.

|  |
| --- |
|  |
| [그림 3.13] 오른쪽 방아쇠를 누를 시 총알 생성 [Thief\_fire(Script)] |
|  |
| [그림 3.14]도둑의 기본정보 값 중 총알개수 줄이기 [Thief\_move(Script)] |

위의 스크립트는 도둑의 총을 쏘는 것을 구현한 것으로 최초 도둑의 정보가 담긴 thief\_move(script)의 heart와 stun의 개수를 불러온다. 그후 update()함수에서는 heart의 개수와 stun의 개수가 0보다 클 경우 오른쪽 오큘러스 퀘스트의 방아쇠버튼의 입력을 받아 총구 앞 firepos포지션에 s\_bullet(prefab)를 위치시킨다. 그후 thief\_move(script)의 stunsub함수를 불러온다.

Thief\_fire(script)로서 호출된 thief\_move(script)에 있는 stunsub함수는 도둑의 기본값 중 stun의 개수를 줄인다.

|  |
| --- |
|  |
| [그림 3.15] 도둑 애니메이션 파라미터 |

최초 상태는 IDLE상태[정지 모션]이며 오큘러스 조이스틱을 움직여 움직임을 제어시 RUN상태[뛰는 모션]로 전환 된다. 조이스틱움직임을 멈출 시 IDLE상태로 전환된다. 그후 방아쇠 버튼을 누를 시 SHOT[총 쏘는 모션]으로 전환 후 다시 이전 상태로 전환 된다. 총알을 맞거나 헌터에게 피격 당했을 경우 DAMAGE[맞는 모션]으로 전환 된다.

**1.3 고려사항**

도둑 이용자의 테스트를 위하여 stun총알의 개수를 최초 5개를 주었다.

**2.1헌터 구현**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | **헌터의 이동 및 슈팅** | | |
| 관련 유스케이스 ID | | SS\_USD\_00\_02  SS\_USD\_00\_05  SS\_USD\_00\_03 | | |
| 구현 설명 | | 헌터 기본값 정보  헌터의 총을 쏠 때의 작동구현 | | |
| 서브시스템명 | 서브시스템 ID | 프로그램 파일 | | 비고 |
| ID | 파일명(Assets) |
| character\_move | SS\_SUB\_00\_14 | hunter(prefabs) | hunter\_move(script) |  |
| trajectory\_appear | SS\_SUB\_00\_13 | Hunter(prefabs) | Hunter\_Fire(script) |  |
| bullet\_diminish | SS\_SUB\_00\_12 | hunter(prefabs) | Hunter\_Fire(script) |  |

**2.2 프로그램 코드 및 알고리즘**

|  |
| --- |
|  |
| [그림 3.16] Hunter의 기본 정보 코드 [Hunter\_move(script)] |

헌터의 정보 :

1. moveSpeed : 헌터의 이동속도 변수 (6.5f)
2. turnSpeed : 헌터의 회전속도 변수 (350f)
3. stun : 헌터의 스턴총알 개수 변수 (300)

|  |
| --- |
|  |
| [그림 3.17] 오른쪽 방아쇠를 누를 시 총알 생성 [Hunter\_Fire(script)] |
|  |
| [그림 3.18] 헌터의 기본정보 값 중 총알개수 줄이기 [Hunter\_move(script)] |

위의 스크립트는 헌터의 총을 쏘는 것을 구현한 것으로 최초 헌터의 정보가 담긴 hunter\_move(script)의 heart와 stun의 개수를 불러온다. 그후 update()함수에서는 stun의 개수가 0보다 클 경우 오른쪽 오큘러스 퀘스트의 방아쇠버튼의 입력을 받아 총구 앞 firepos포지션에 s\_bullet(prefab)를 위치시킨다. 그후 hunter\_move(script)의 stunsub함수를 불러온다.

hunter\_fire(script)로서 호출된 hunter\_move(script)에 있는 stunsub함수는 헌터의 기본값 중 stun의 개수를 줄인다.

|  |
| --- |
|  |
| [그림 3.19] 헌터 애니메이션 파라메터 |

최초 상태는 IDLE상태[정지 모션]이며 오큘러스 조이스틱을 움직여 움직임을 제어시 RUN상태[뛰는 모션]로 전환 된다. 조이스틱움직임을 멈출 시 IDLE상태로 전환된다. 그후 방아쇠 버튼을 누를 시 SHOT[총 쏘는 모션]으로 전환 후 다시 이전 상태로 전환 된다. 총알을 맞을 시 DAMAGE[맞는 모션]으로 전환 된다.

**2.3 구현 모습**

|  |
| --- |
|  |
| [그림 3.20] (왼쪽) 도둑의 총을 맞고 헌터가 잠시 기절한 모습 (오른쪽) 헌터의 총을 맞고 도둑이 멈춘 모습 |

**2.4 고려사항**

Demo상 총알의 이상여부를 판단하기 위해 헌터의 기본 정보 스턴 총알은 300개로 되어있다

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I4** | **Interaction구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 |  | | |
| 단계명 | | 구현 | 작성일자 |  | 버전 | V1.0 |

**1. 시스템목록**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 관련 유스케이스 ID | | SS\_USD\_00\_02  SS\_USD\_00\_04 | | |
| 구현 설명 | | 스턴 총알을 맞을 때 헌터와 도둑의 제한  회복 총알을 맞을 때 도둑의 life개수 증가  헌터가 도둑을 때렸을 때의 상황 | | |
| 서브시스템명 | 서브시스템 ID | 프로그램 파일 | | 비고 |
| ID | 파일명(Assets) |
| Trajectory\_stun | SS\_SUB\_00\_06 | Thief(prefabs) | thief\_collider(script) |  |
| Trajectory\_stun | SS\_SUB\_00\_06 | Hunter(prefabs) | Hunter \_collider(script) |  |
| Trajectory\_stun | SS\_SUB\_00\_06 | Thief(prefabs) | thief\_collider(script) |  |
| hit\_hunter | SS\_SUB\_00\_08 | Hunter(prefabs) | Hunter \_hit(script) |  |
| beat\_thief | SS\_SUB\_00\_09 | Thief(prefabs) | Hunter \_hit(script) |  |

**2. 프로그램 코드 및 알고리즘**

|  |
| --- |
|  |
| [그림 3.21] Thief\_collider(Script) |
|  |
| [그림 3.22] Hunter\_collider(script) |

도둑은 헌터가 쏜 스턴 총알(tag: s\_bullet)을 맞을 시 thief\_move의 stun()함수를 불러온다.

Stun()함수의 기능은 맞은 도둑의 movespeed와 turnspeed의 값을 5초간 0으로 만든다.

도둑은 도둑이 쏜 소생 총알(tag : r\_bullet)을 맞을 시 thief\_move의 recoverhit()함수를 불러온다. recoverhit()함수의 기능은 맞은 도둑의 heart의 개수를 1증가시킨다.

헌터은 도둑가 쏜 스턴 총알(tag : s\_bullet)을 맞을 시 hunter\_move의 stun()함수를 불러온다. Stun()함수의 기능은 맞은 헌터의 movespeed와 turnspeed의 값을 5초간 0으로 만든다

|  |
| --- |
|  |
| [그림 3.23] Hunter\_hit(script) |

헌터가 가지고 있는 오브젝트hunter\_gun(tag: hunter\_gun)이 thief오브젝트(tag: thief)에 충돌 시 hunter\_move의 hit\_stun()함수를 불러온다. Hit\_stun()은 hunter의 속도를 5초간 멈추게 한다.

또한 thief\_move의 hit()함수 또한 불러온다. Hit()함수는 도둑의 목숨 (heart)를 1낮춘다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I5** | **네트워크 구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 구현 | 작성일자 | 2019.12.21 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 네트워크 | | |
| 구현 설명 | | 멀티플레이를 지원하기 위한 네트워크 설정 | | |
| 프로그램 파일 | | | Inspector & setting  [Add Component] | 비고 |
| ID | 파일명(Assets) | |
| Network\_Photon | Photon Pun2 | |  |  |

**1. 프로그램 파일**

**2. 프로그램 코드 및 알고리즘**

|  |
| --- |
|  |
| [그림 3.24] 네트워크 초기설정 및 마스터서버 접속 |

photonNetwork에서 제공하는 메소드를 활용하여 초기값을 setting한다.

MasterServer에 접속한 뒤 JoinRandomRoom함수를 통하여 생성되어 있는 방에 접속한다.

|  |
| --- |
|  |
| [그림 3.25] 방접속 예외처리 및 접속구현 |

방접속에 실패했을 경우 새로운방을 생성하고 RoomOptions을 통하여 초기 방 설정한다

방에 접속을 성공하면 Createcheracter을 통하여 자신의 캐릭터를 맵에 생성하여준다.

|  |
| --- |
|  |
| [그림 3.26] 방접속 예외처리 및 접속구현 |

방 접속에 실패했을 경우 새로운 방을 생성하고 RoomOptions을 통하여 초기 방 설정한다.

방에 접속을 성공하면 Createcheracter을 통하여 자신의 캐릭터를 맵에 생성하여 준다.

|  |
| --- |
|  |
| [그림 3.27] 서버에 자신의 캐릭터를 생성 |

PointGroup에 있는 좌표값을 통하여 랜덤한 위치에서 캐릭터들이 생성되도록 한다.

생성된 캐릭터를 카메라가 추적하고 총을 사격하기 위하여 값들을 Setting하여 준다.

**3. 구현 모습**

|  |
| --- |
|  |
| [그림 3.28] 네트워크 구현을 위한 Asset |
| Photon사에서 제공하고 있는 PUN2 서버를 통하여 동시접속이 20명까지 가능한 무료서버를 사용하였다.  본 Asset에서 제공하는 다양한 Scipt를 통하여 서버에 접속하고 클라우드 서버를 통한 게임진행을 구현하였다. |

|  |
| --- |
|  |
| [그림 3.29] Object에 네트워크 관련 Component추가 |
| Photon View, Photon Transform View, Photon Animator View의 스크립트를 통하여 온라인에 접속한 모든 유저들이 같은 환경의 게임을 즐길 수 있도록 구현하였다. |

|  |
| --- |
|  |
| [그림 3.30] 서버를 통하여 유저들이 접속한 상태 |
| 클라우드 서버를 통하여 두명의 클라이언트가 접속하여 게임내부에 진입한 모습이다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I6** | **UI구현** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 구현 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 플레이시, 유저들 앞에 보이게 되는 화면UI 모습 | | |
| 구현 설명 | | 플레이 유저가 원활하게 게임을 진행하는데 있어 보게 될 UI 화면으로 현재 남은 목숨, 남은 총알 개수 등을 파악할 수 있다. | | |
| 서브시스템명 | 서브시스템 ID | 프로그램 파일 | | 비고 |
| ID | Inspector & setting  [Add Component] |
| SS\_SUB\_00\_15 | Hunter\_infoo |  | Text(TMP)bullet |  |
| SS\_SUB\_00\_16 | Thief\_infoo |  | Text(TMP)bullet  Text(TMP)life |  |

**1. 시스템 목록**

**2. 프로그램 코드 및 알고리즘**

**2.1헌터관점 UI**

|  |
| --- |
|  |
| [그림 3.31] Thief\_heart\_info Script |

TextMeshPro를 나타내주는 TextMeshProUGUI resourceText 선언해준다.

임의의 int형 변수인 life를 선언해줘서 life가 Tag가 Thief인 게임 오브젝트의 스크립트 thief.move에서의 변수 heart를 받게 한다. Void Update()에 life가 계속 thief.move의 heart값을 업데이트 하게한 후, TextMeshPro의 텍스트가 텍스트 LIFE :랑 도둑의 목숨 값을 계속 표기 해준다.

|  |
| --- |
|  |
| [그림 3.32] Thief\_bullet \_info Script |

TextMeshPro를 나타 내는 TextMeshProUGUI resourceText 선언해준다.

임의의 int형 변수인 bullet을 선언해줘서 bullet이 Tag가 Thief인 게임 오브젝트의 스크립트 thief.move에서의 스턴 알의 개수를 뜻하는 stun을 받게 한다. Void Update()에 bullet이 계속 thief.move의 stun값을 업데이트 한 후, TextMeshPro의 텍스트가 텍스트 BULLET:랑 도둑의 스턴총알 개수를 속 표시 해준다.

**2.2 헌터 관점 UI**

|  |
| --- |
|  |
| [그림 3.33] Thief\_bullet \_info Script |

TextMeshPro를 나타내는 TextMeshProUGUI resourceText 선언해준다.

임의의 int형 변수인 bullet을 선언해줘서 bullet이 Tag가 Player인 게임 오브젝트의 스크립트 hunter.move에서의 스턴 총알의 개수를 뜻하는 stun을 받게 한다. Void Update()에 bullet이 계속 hunter.move의 stun값을 업데이트 한 후, TextMeshPro의 텍스트가 텍스트 BULLET:랑 헌터의 스턴 알 개수를 계속 표기 해준다.

**3. 구현 모습**

|  |
| --- |
|  |
| [그림 3.34] 남은 목숨과 총알 개수를 나타내는 UI |
|  |
| [그림 3.35] 총알의 조준점과 현재 방향을 알려주는 UI |

**4. 고려사항**

|  |
| --- |
| 콘솔 화면에서 보여지는 것과 VR 화면에서 보여지는 것이 다르기 때문에 텍스트들의 위치 조정을 빌드해서 확인해야한다.  Canvas 내 Text를 만든 후, OVRCameraRig에 Text를 맞출 경우 해당Canvas-Text의 글씨 해상도가 깨졌었습니다. 이를 해결 하기위해, Text Mesh Pro를 이용해야 합니다. 예전버전은 Asset Store에서 따로 다운을 받아야 하지만, 지금은 Unity는 기본적으로 구현되도록 되어있어, 성공적으로 변수가 실시간으로 표기되는 걸 확인이 가능했습니다. 콘솔 창으로 보면서 변수 조정하는 건 어느 정도 한계가 있기 때문에, 위치를 조정할 때마다 Oculus Quest에 빌드 후 테스트해야 합니다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I7** | **게임 Scene 변환 구현** | | | | | |
| 시스템명 | |  | 서브시스템명 |  | | |
| 단계명 | | 구현 | 작성일자 |  | 버전 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | | 게임 시작부터 종료까지의 Scene 변환 | | |
| 관련 유스케이스 ID | | - | | |
| 구현 설명 | | 게임 서버에 접속되고, 영상을 확인 후, 게임 화면으로 전환된다. | | |
| 프로그램 파일 | | | Inspector & setting  [Add Component] | 비고 |
| ID | 파일명(Assets) | |
| FirstScene | Photon | | Canvas, Button |  |
| SecndScene | Video | | Video Player, Canvas, Button |  |
| DemoScene | - | | - | 본 게임 화면 |
| EndScene | Video | | Video Player |  |

**1. 프로그램 파일**

**2. 프로그램 코드 및 알고리즘**

|  |
| --- |
|  |
| [그림 3.36] changescene script |
| using UnityEngine.SceneManagement;를 추가하면 해당 scene으로 이동가능한 SceneManager.LoadScene 기능을 사용할 수 있다. |

**3. 구현 모습**

|  |
| --- |
|  |
| [그림 3.37] Scene 전환 순서  FirstScene[게임 로비]🡪 SecondScene[스토리 영상]🡪 DemoScene[게임 창]🡪 EndScene[게임 종료] |
| 버튼 컴포넌트에 On Click()이 존재한다. 이곳에 위에서 쓴 스크립트가 들어간 오브젝트를 추가하여 해당 버튼을 클릭하면 원하는 Scene으로 변경이 가능한 메소드가 실행된다. |

**4.1 시험단계**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T1** | **사용자 지침서** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.21 | 버전 | V1.0 |

**1. 사용 지침**

**1.1 설치 방법**

다른 설치는 필요없이 빌드 된 오큘러스 퀘스트를 착용하여 게임에 접속한다.

**1.2 시작 및 종료 방법**

[그림 7.1 Scene 전환 순서를 참고]HMD착용 후 게임 시작화면에 있는 Start 버튼을 클릭하면 게임 영상으로 넘어간다. 영상 시청한 후, 게임 화면으로 화면이 넘어간다.

**2. 기능별 사용법**

**2.1 기능 이름 & 설명**

|  |
| --- |
|  |
| [그림 4.1] 도둑의 조작법 |

|  |
| --- |
|  |
| [그림 4.2] 헌터의 조작법 |

**2.2 주의 사항**

-왼쪽의X,Y버튼과 오른쪽의 A,B버튼은 같은 역할을 한다.(양손 가능)

-헌터가 도둑을 때려 목숨을 앗아갈 때, 때린 헌터는 3초간 정지 상태가 된다.

**3. 캐릭터 별 게임 방식**

|  |
| --- |
|  |
| [그림 4.3] 도둑 관점 |

|  |
| --- |
|  |
| [그림 4.4] 헌터 관점 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T2** | **운영자 지침서** | | | | | |
| 시스템명 | | Thiefs | 서브시스템명 | 도둑들 | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

**1. 시스템 설치**

**1.1 시스템 구성도**

|  |
| --- |
|  |
| [그림 4.5] 시스템 구성도 |
| Client들은 VR기기를 착용하여 게임서버에 접속함으로써 상호 작용 멀티 네트워크로 게임을 즐길 수 있습니다. 게임서버 안에서 플레이어들 다른 유저들의 실시간 행동을 볼 수 있으며, 게임을 Unity 3d 엔진에서 또한 실시간으로 볼 수 있습니다. |

**1.2 하드웨어(HMD)**

|  |
| --- |
|  |
| [그림 4.6] 오큘러스 퀘스트 |
| 하드웨어는 Oculus quest를 사용하였으며, 무선이어서 굉장히 몸이 자유롭습니다.  리프트에서 가장 욕먹던 자체 헤드 셋이 없어지고, 개인 이어폰을 사용 가능하며,  HMD에 달려있는 4개의 트래커가 바로 컨트롤러를 트래킹 하는 방식이라서 공간에 굉장히 자유롭습니다. (작은곳부터 넓은 곳까지)  또한, 트래킹의 범위 제한이 없고, 카메라로 외부를 볼 수 있어, HMD의 고질적 문제인 시야가 모두 가려지기 때문에 위험성 문제가 있었는데, 퀘스트는 HMD를 끼고도 전방을 볼 수 있는 기능을 가지고 있기 때문에 안전성을 확보하였습니다. |

**1.3 소프트웨어**

|  |
| --- |
|  |
| [그림 4.7] Unity hub/ Unity 3D 엔진 |
| 3D 및 2D [비디오 게임](https://ko.wikipedia.org/wiki/%EB%B9%84%EB%94%94%EC%98%A4_%EA%B2%8C%EC%9E%84)의 개발 환경을 제공하는 [게임 엔진](https://ko.wikipedia.org/wiki/%EA%B2%8C%EC%9E%84_%EC%97%94%EC%A7%84)이자, 3D 애니메이션과 건축 시각화, [가상현실](https://ko.wikipedia.org/wiki/%EA%B0%80%EC%83%81%ED%98%84%EC%8B%A4)(VR) 등 인터랙티브 콘텐츠 제작을 위한 통합 저작 도구로 게임 개발에 사용하는 스크립트 언어는 [C#](https://ko.wikipedia.org/wiki/C_%EC%83%A4%ED%94%84_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%EC%96%B8%EC%96%B4))와 [자바스크립트](https://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8)(UnityScript라는 이름으로)를 지원합니다. 스크립트 작성은 유니티와 함께 설치되는 [비주얼 스튜디오](https://ko.wikipedia.org/wiki/%EB%B9%84%EC%A3%BC%EC%96%BC_%EC%8A%A4%ED%8A%9C%EB%94%94%EC%98%A4)(맥OS의 경우, [모노디벨로프](https://ko.wikipedia.org/w/index.php?title=%EB%AA%A8%EB%85%B8%EB%94%94%EB%B2%A8%EB%A1%9C%ED%94%84&action=edit&redlink=1))를 이용하며, 다른 편집기와 연동하는 것도 가능합니다. |

**1.4 네트워크 구성**

|  |
| --- |
|  |
| [그림 4.8] 네트워크 구성도 |
| Photon cloud 서버로, 사용자들의 상호 교류 데이터를 기록해 실시간으로 사용자 데이터를 전송합니다. 한 게임서버에 4명이 접속하여 PLAY가 가능하며 독립적인 게임서버들을 관리하는 마스터서버의 경우는 100명정도가 접속가능 하도록 지원됩니다.  **마스터** **서버** 마스터 서버는 서버를 관리하는 위치이며, 자체적으로 여러 게임 서버와 연결하여 각 서버의 상태를 기록하고 감독하며 주요 데이터를 점검하고 업데이트합니다. 자주 사용하는 클라이언트는 마스터 서버와 연결한 후 게임 서버의 연결 데이터를 취득해야 하며 나아가 실제 게임 서버 플랫폼과 연결해야 합니다. 일부 마스터 서버는 서로 다른 클라이언트를 다른 게임 서버로 분배해 동적이거나 맞춤화 된 로드 밸런싱을 조정합니다.  **네임** **서버(Name Server)** 네임 서버 또한 관리 서버로, 자체적으로 여러 지리적 위치별로 분포한 세계 각지의 서로 다른 마스터 서버에 연결하기 때문에 해당 서버로 연결만 되면 조건에 따라 마스터 서버에서 원하는 데이터를 얻을 수 있어 후속 연결에 유리합니다. 최고 품질로 동적 연결한 마스터 서버는 연결하고자 하는 마스터 서버에 문제가 발생한 경우에도 게임 서버와 연결할 수 있습니다.  **게임** **서버** 독립된 서버로, 사용자들의 상호 교류 데이터를 기록해 실시간으로 사용자 데이터를 전송합니다. 이전 단계에서 부하 분산을 위해 사용자별로 서로 다른 게임 서버로 분산시켰기 때문에 각 서버를 독립적으로 운용할 수 있습니다. 물론, 일부 게임 세계 설계에서 각 게임 서버별로 데이터를 전송할 수 있게 하려면 게임 서버에서 별개로 설계합니다. |

**2. 사용 지침**

**2.1.1 설치 방법**

|  |
| --- |
|  |
| [그림 4.9] Oculus 앱 |
| Oculus Quest를 컴퓨터에 연결해서 사용하려면 Oculus가 배포하는 Oculus앱을 필수적으로 설치야 된다. |

|  |
| --- |
|  |
| [그림 4.10] Oculus Integration |
| 유니티에서 Oculus Quest를 사용해서 개발을 하려면 Oculus가 Asset Store에서 배포하는 Oculus Integration을 필수적으로 설치야 된다. |
| 2  3  1 |
| [그림 4.11] build setting |
| Oculus는 모바일 플랫폼이기 때문에, Asset을 Import한 후에 빌드 창에 들어가서 기존의 플랫폼이던 PC에서 안드로이드 기반으로 플랫폼 스위치를 해줘야 된다.  그리고 플랫폼 스위치를 한 후에, PLAYER SEETING으로 들어가서 XR SETTINGS로 들어가서 Virtual Relity Supported 체크한 후에, Virtual Reality SDKs에서 오큘러스로 추가를 해준다.  Xr Setting창을 최소화 한후에, Other settings를 눌러준다.  Graphics APIs에 Vulkan이 있을 경우에는, 빌드를 했을 때 Vulkan이 있다고 오류가 뜨면서 빌드가 안된다. 고로 Auto Graphics APIs를 체크해제 해주고, Vulkan을 삭제시켜준다.  VR은 안드로이드 4.4버전 이상에서만 작동이 되니 API Level 19 Kikat으로 요구한다**.**  그러므로 Minimum API Level을 API Level 19로 설정해준다.  오큘러스 퀘스트는 핸드폰과의 연동을 통해 빌드 파일을 넣어 줄 수 있습니다.  오큘러스와 연동된 핸드폰으로 오큘러스 앱에 들어가서 개발자 모드를 켜준다.  그 후, 오큘러스를 컴퓨터에 연결하고 빌드 해주면 된다. |

**2.1.2 고려사항**

|  |
| --- |
| 처음 빌드할 때는 시간이 오래 걸린다. 그리고 프로젝트를 압축해서 다른 컴퓨터로 옮길 때는 플랫폼이 PC, MAC & Linux Standalone으로 초기화되기 때문에 다시 플랫폼을 스위치 해주는 것이 좋다. 또한 콘솔과 oculus로 플랫폼을 스위치 할 때 시간이 오래 걸리므로, 매번 확인하기보다는, 콘솔에서 충분히 확인 후 최종으로 oculus로 확인하는 것이 시간을 줄일 수 있다. |

**2.2 시작 및 종료 방법**

|  |
| --- |
|  |
| [그림 4.12] Scenes in Build |
| First Scene[스타팅 화면], Second Scene[스토리 영상 화면], Demo Scene  [게임화면], End Scene[게임 엔딩 화면]들을 순서대로 빌드하며 원하는 Scene의 순서대로 확인 가능하다. 추가할 Scene은 build하여 추가를 하거나, 원치 않는 Scene은 체크 해제해준다. 그리고 Run Device를 연결된 oculus로 지정해준 후, Texture compression을 ETC로 바꿔준다. |
|  |
| [그림 4.13] Oculus 내 처음과 끝 이미지 |
| 중간에 게임이 멈춘다면 다시 시작 버튼을 클릭하고, 게임을 종료할 때는 홈 버튼을 눌러서 종료 할 수 있고 사진 찍기로 영상을 찍거나 할 수 있다. |

**2.3 오류 메시지 및 원인**

|  |
| --- |
|  |
| [그림 4.14] Unable to build gradle project 오류 |
| Unity에 설치된 JDK혹은 SDK의 버전이 충돌을 일으킬 때 혹은 Oculus 기기에 이미 APK 파일이 있을 경우, 빌드 실패 메시지와 함께 오류가 발생합니다.  따라서 해결방법은 JDK, SDK의 버전을 다운그레이드 해주거나, Oculus Quest 내부에 있는 APK를 삭제하고 다시 빌드 해줍니다 |

|  |
| --- |
| 화면 깨짐 현상오류 |
| 오큘러스 퀘스트에 있는 APK들을 지워주지 않고 계속 빌드를 하면, 퀘스트로 화면을 보았을 때, 메모리 부족으로 인하여 화면이 점점 깨지면서 아무것도 못 알아볼 때가 있다. 따라서 빌드하기 전에 APK파일들을 지워주어 메모리 부족 현상으로 인한 오류가 발생하지 않도록 해야 한다. |

**2.4 프레임체크**

|  |
| --- |
|  |
| [그림 4.15 ]프레임 체크 된 모습 |
|  |
| [그림 4.16] Framecheck script |
| Void Awake()에서는 화면에 표시할 위치, 폰트사이즈, 폰트, 색깔 등 포맷을 정해줍니다.  IEnumerator worstReset()은 15초마다 최저프레임을 초기화해줍니다.  Void Update()에서는 프레임 간격 사이의 시간을 계산해줍니다.  Void OnGUI에서는 프레임과 반응속도와 최저 프레임을 계산해서 계속 화면상에 표기합니다. |

**2.5 Apk파일 설치하기**

|  |
| --- |
|  |
| [그림 4.17] 파일 열기 |
|  |
| [그림 4.18] 파일 선택화면 |
|  |
| <그림 4.19> 설치 진행화면 |
| SideQuest라는 프로그램을 이용하여 Unity를 통하여 Build가 완료된 Apk파일을 Oculus Quest 디바이스에 설치하여 이용할 수 있다. 게임 재실행 시 Re-Build 과정이 필요 없다.  <그림 4.17>에 있는 SideQuest의 메인화면에 빨간색 표시를 누르면 <그림4.18>의 파일 선택화면이 나온다. 설치할 Apk 파일을 선택하면 <그림4.19>과 같이 설치 진행 상황 및 완료를 확인할 수 있다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T3** | **시스템시험 결과서** | | | | | |
| 시스템명 | | 플레이어 피격 애니메이션 | 서브시스템명 |  | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | 기능 | | | |
| 관련 요구사항 ID | C003, G008, G011 | | | |
| 시험시나리오 명 | | 시험절차 | 만족기준 | 시험결과 |
| 도둑 헌터 피격 애니메이션 | | -헌터인 플레이어가 도둑을 향해 stun총알을 쏜다.  -도둑인 플레이어가 헌터를 향해 stun 총알을 쏜다. | ●응답 시간  총을 맞고 나서 1초 안에 캐릭터가 모션을 취하는지 여부  ●작동 여부  총을 맞고나서 애니메이션이 정상적으로 작동하는지 여부 | 헌터는 총알을 맞았을 때, 넘어지는 모션이 제대로 작동했다.  도둑은 총알을 맞았을 때, 넘어지는 모션이 제대로 작동하지 않았다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T3** | **시스템시험 결과서** | | | | | |
| 시스템명 | | 헌터 Hit 애니메이션 | 서브시스템명 |  | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 시험 유형 | 기능 | | | |
| 관련 요구사항 ID | G004 | | | |
| 시험시나리오 명 | | 시험절차 | 만족기준 | 시험결과 |
| 헌터 Hit 애니메이션 | | -헌터인 플레이어가 도둑인 플레이어를 향해 컨트롤러를 휘두른다. | ●작동 여부  플레이어가 컨트롤을 휘둘렀을 때 헌터의 팔도 같이 움직이는지 여부.  헌터가 휘두르는 칼이 다른 물체들과 충돌을 하는지 여부  ●응답 시간  플레이어가 컨트롤을 휘둘렀을 때 헌터의 팔이 1초안에 동작을 따라하는지 여부 | 헌터가 휘두르는 칼이 다른 물체들과 잘 충돌을 했다.  플레이어가 컨트롤러를 움직이자 딜레이 없이 헌터의 팔도 잘 움직여졌다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T3** | **시스템시험 결과서** | | | | | |
| 시스템명 | | Idle&Run | 서브시스템명 |  | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 시험 유형 | 기능 | | |
| 관련 요구사항 ID | C002 | | |
| 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 |
| Test\_Idel&Run | -플레이어가 가만히 있다가 움직인다. | ●작동 여부  플레이어가 컨트롤러를 사용해서 캐릭터를 움직였을 때 캐릭터가 달리는 모션을 취한다.  ●응답 시간  플레이어가 컨트롤을 이용해 캐릭터를 움직였을 때 캐릭터 클론이 1초내에 움직이는지 여부 | 멈춰있던 캐릭터를 움직이자 달리는 모션으로 움직였다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **T3** | **시스템시험 결과서** | | | | | |
| 시스템명 | | 아이템 획득 기능 정상 작동 여부 | 서브시스템명 |  | | |
| 단계명 | | 시험 | 작성일자 | 2019.12.22 | 버전 | V1.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 시험 유형 | 기능 | | |
| 관련 요구사항 ID | G001, G005 | | |
| 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 |
| 상자 기능 정상 작동 여부 | -도둑인 플레이어가 상자에 접근해서 기능버튼을 누른다.  -상자가 열었을 때 상자가 사라지고 아이템이 정상적으로 생성되는지 확인한다. | ●작동 여부  도둑 플레이어가 상자에 접근해서 버튼을 누르면 상자가 열리는지 여부.  상자가 열렸을 때 상자가 사라지고 아이템이 생성되는지 여부  ●응답 시간  플레이어가 컨트롤을 이용해 상자를 열었을 때 상자가 1초내에 사라지는 여부. | 상자 근처로 가서 버튼을 누르자 상자가 정상적으로 사라지면서 아이템이 생성되었다. |