



BAREN

B A R E N : T H E P R O P H E T

바렌 : 더 프라핏

프로토타입 개발 내역서

TEAM SEOLHWA

업데이트 기록			
버전	날짜	항목	이름
Ver.1.0	2024.07.24	프로토타입 개발 내역서 초안 작성	홍진선

0. Index

0. Index 1

1. 개요 3

1.1 게임 소개 3

1.2 프로토타입 개발 목적 및 범위 3

1.2.1. 개발 목적 3

1.2.2. 개발 범위 3

1.3. 기술 스펙 3

1.3.1. 사용 기술 및 툴 3

2. 플레이 매뉴얼 4

2.1. 테스트 시나리오 4

2.2. 조작 방법 5

3. 기획안 6

3.1. 기본 조작 6

3.1.1. 조작키 6

3.1.2. 이동 7

3.1.3. 카메라 제어 8

3.1.4. 공격 및 스킬 9

3.1.5. UI 및 상호작용 9

3.2. UI 11

3.2.1. 씬 플로우 11

3.2.2. 와이어 프레임 11

3.3. 레벨 12

3.3.1. 레이아웃 12

3.3.2. 레벨 컨셉 12

4. 개발 과정 및 현황 13

4.1. 프로토타입 제작 13

4.1.1. 기본 조작 13

4.1.2. 깃허브 연동 14

4.1.3. 모델 및 애니메이션 15

4.1.4. 전투 15

4.1.5. UI 17

4.1.6. 트리거 20

4.1.7. 추가 기능 20

5. 평가 및 업데이트 계획 21

5.1. 평가 21

5.2. 후속 계획 21

1. 개요

1.1 게임 소개

- ⇒ Baren: The Prophet(이하 BTP)은 액션 로그라이트 RPG 장르의 게임입니다.
- ⇒ 주요 콘텐츠는 세미 오픈월드를 탐험하고 메인 퀘스트를 진행하며 7개의 챕터를 클리어하는 것입니다.

1.2. 프로토타입 개발 목적 및 범위

1.2.1. 개발 목적

- ⇒ 게임의 메인 시스템인 ‘메모리아 시스템’을 테스트하고 게임 플레이의 재미 요소를 검증합니다.
- ⇒ 그래픽 스타일과 UI가 게임의 컨셉과 일관성이 있는 지 평가합니다.
- ⇒ 이번 프로토타입은 PC 플랫폼으로 가정하고 개발합니다.

1.2.2. 개발 범위

분류	목록
주요 기능	캐릭터 이동 및 전투, 상호작용, 메모리아 선택
레벨	블록아웃, 몬스터 배치
UI	타이틀 스크린, HUD, 주요 인게임 UI
그래픽	캐릭터 모델/애니메이션, 오브젝트(건물/Prop 등) 모델
사운드	로비/게임 BGM

1.3. 기술 스펙

1.3.1. 사용 기술 및 툴

분류	목록
게임 엔진	Unity (2022.03.27f1)
개발 언어	C#
UI 디자인	Figma
기타	Mixamo(모델/애니메이션), Git-hub(버전관리)

2. 플레이 매뉴얼

2.1. 테스트 시나리오

([시연 영상\[유튜브\]](#)) *BGM이 들어가 있습니다. 소리가 클 수 있으니 유의해주시면 감사하겠습니다.

- 1) 타이틀 스크린에서 '새 게임'으로 게임을 시작합니다. (캐릭터는 무기와 알터 에고가 없는 상태로 생성됩니다.)
- 2) 게임을 처음 시작하면 스폰되는 지점 근처에 위치한 무기 3종(검, 활, 지팡이)중 하나를 고르면 해당 무기에 맞는 알터 에고가 선택됩니다.
- 3) 무기를 고른 후 옆의 '빛 바랜 석판'과 상호 작용을 하면 메인 퀘스트가 진행됩니다.
- 4) 바로 앞의 몬스터와 전투를 합니다.
- 5) 몬스터 뒤에 위치한 '예언의 석판'과 상호 작용을 하면 메모리아를 선택할 수 있는 선택지가 등장합니다.
- 6) 세계의 메모리아 선택지 중 하나를 선택합니다. (선택이 완료되면 예언의 석판은 빛을 잃게 됩니다.)
- 7) 스킬을 사용하며 근처의 다른 몬스터와 전투를 합니다.
- 8) 이후 메인 퀘스트를 따라 마을로 이동합니다.
- 9) 마을 앞의 '예언의 석판'과 한 번 더 상호 작용을 하고 메모리아를 선택합니다.
- 10) 마을의 NPC와 대화를 하여 메인 퀘스트를 진행합니다.
- 11) 마을을 둘러본 후 게임 메뉴를 열어 로비로 돌아갑니다.
- 12) 활성화된 '데이터 로드' 버튼을 누르면 이전 위치에서 캐릭터가 스폰되고 게임을 다시 플레이합니다.

2.2. 조작 방법

PC (키보드&마우스)

ESC

F1

F2

F3

F4

F5

F6

F7

F8

F9

F10

F11

F12

`

1

2

3

4

5

6

7

8

9

0

-

=

BACK SPACE

TAB

Q

W

E

R

T

Y

U

I

O

P

[

]

₩

Caps Lock

A

S

D

F

G

H

J

K

L

;

'

Enter

L Shift

Z

X

C

V

B

N

M

,

.

/

R Shift

Ctrl

Ⓜ

Alt

SPACE BAR

한/영

Fn

한자

Ctrl

W

A

S

D

달리기

R

체멸 스킬

P

캐릭터 정보

Space

점프

ESC

게임 메뉴

F

상호 작용

L Shift

대쉬

M

월드맵

Q


타겟 고정


L Ctrl

걷기


I

인벤토리






일반 공격



고유 스킬



카메라 회전

콘솔 (PS 컨트롤러)

L2

L1

R2

R1

체멸 스킬

게임 메뉴

고유 스킬

점프

일반 공격 / 열기

대쉬 / 닫기

상호 작용

달리기

카메라 회전

걷기 / UI 선택



3. 기획안

3.1. 기본 조작

3.1.1. 조작키

⇒ 프로토타입은 PC 기준으로 개발하기 때문에 조작키도 PC 기준으로 설정합니다.

조작		Key
이동	전방 이동	W
	후방 이동	S
	좌측 이동	A
	우측 이동	D
	걷기	L Ctrl
	웅크리기	C
	점프	Space
	대쉬	L Shift
카메라 제어	카메라 회전	마우스 회전
	타겟 고정	Q
공격	일반 공격	마우스 좌클릭
스킬	고유 스킬	마우스 우클릭
	체멸 스킬	R
	탈 것 소환	V
UI	게임 메뉴	ESC
	캐릭터 정보	P
	퀘스트	J
	월드맵	M
	인벤토리	I
	스킬	K
상호작용	인 게임 상호작용	F
	UI 선택	마우스 좌클릭
	메모리아 선택	1~3

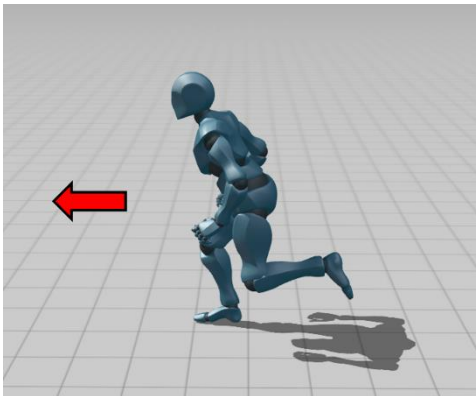
3.1.2. 이동

- ⇒ 해당 표의 Factor Value는 기본 수치를 나타냅니다.
- ⇒ 메모리아 선택, 캐릭터 스탯에 따라 수치는 변화합니다.

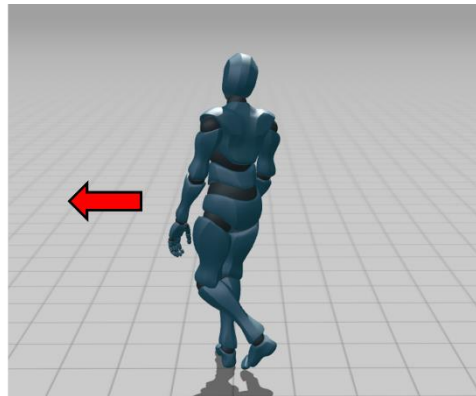
조작	Factor Type	Factor Value
달리기	이동 속도	4m/s
타겟 고정 시 후방/좌우측 이동	이동 속도	2m/s
걷기	이동 속도	1m/s
웅크리기	이동 속도	1m/s
점프	점프 높이	0.7m
대쉬	대쉬 거리	5m

1) 달리기 (기본 이동)

- ⇒ 카메라가 보는 방향을 기준으로 전후좌우 네 방향을 결정합니다.
- ⇒ 이동 방향으로 캐릭터가 회전합니다.
- ⇒ 타겟 고정 시 캐릭터가 대상을 바라본 상태로 뒷걸음(후방), 게걸음(좌우측)으로 이동합니다.

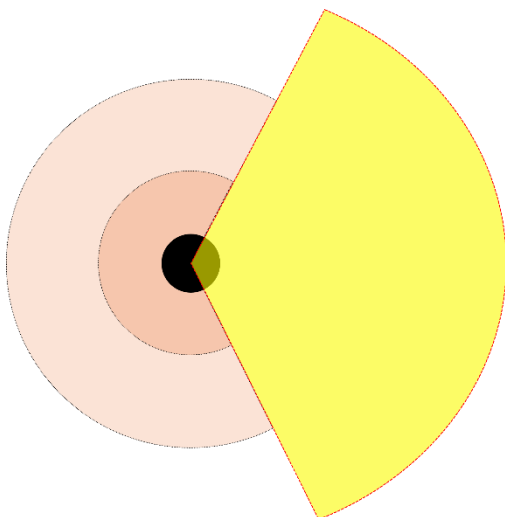


<기본 좌측 이동>





<타겟 고정 좌측 이동>

2) 걷기 및 웅크리기



- ⇒ 몬스터의 발소리 감지 범위는 달리기/걷기/웅크리기 마다 다릅니다.
(단, 몬스터의 시야 감지 범위는 무시하지 않습니다.)
- ⇒ 걷기의 발소리 감지 범위는 달리기의 50%로 적용됩니다.
- ⇒ 웅크리기의 발소리 감지 범위는 달리기의 0%로 적용됩니다.

-  달리기의 발소리 감지 범위
-  걷기의 발소리 감지 범위

3) 점프

⇒ 일부 지형에 올라가기 위해 점프를 사용합니다.

4) 대쉬

⇒ 대쉬는 무적 기능이 있어 적의 공격을 회피하기 위한 수단으로 사용합니다.

⇒ 메모리아로 대쉬를 강화할 수 있습니다.

3.1.3. 카메라 제어

⇒ BTP의 기본 해상도는 1920x1080(FHD)입니다.

⇒ 기본 카메라는 키보드 이동(wasd)을 기준으로 하는 3인칭 시점의 백뷰입니다.



⇒ 카메라가 바라보는 방향이 캐릭터의 정면 방향이 됩니다.

⇒ 마우스 이동으로 카메라를 제어합니다. (단, 카메라 줌인/줌아웃 기능은 없습니다.)

⇒ 유저가 컨트롤 할 수 있는 카메라의 상하 최대 회전 각도에 제한을 둡니다.

⇒ 게임 플레이 시 마우스는 화면 중앙에 고정되고 커서는 사라집니다.

⇒ 일부 UI가 켜지면 마우스가 등장합니다.

⇒ 화면 정중앙에서 가장 가까운 몬스터로 타겟 고정(락온)을 할 수 있습니다. 이때 카메라는 타겟에게 고정됩니다.

3.1.4. 공격 및 스킬

3.1.5. UI 및 상호작용

3.2. UI

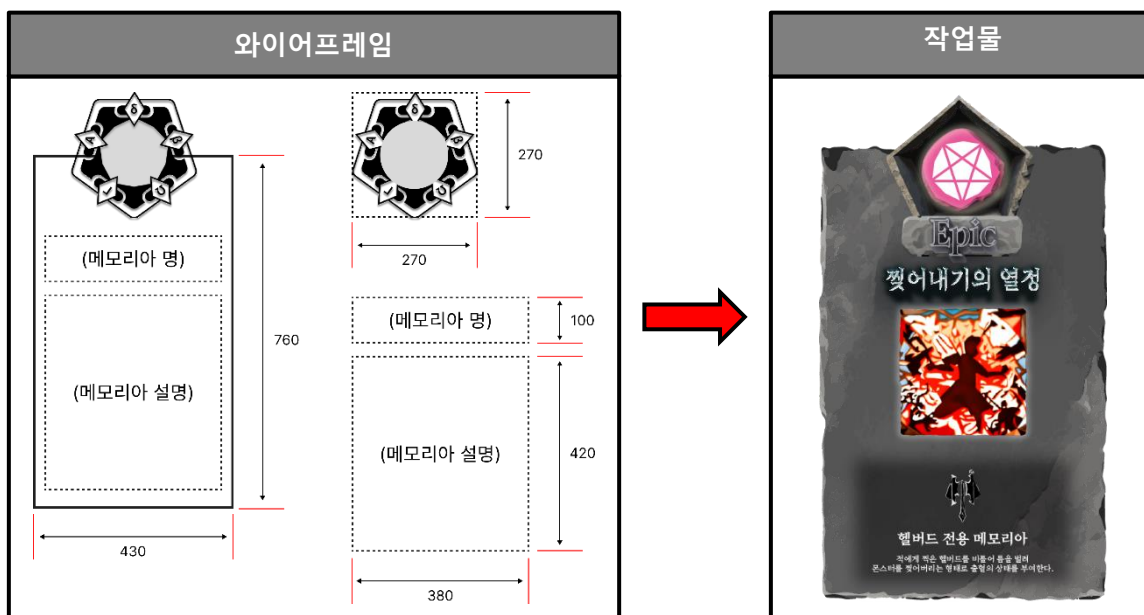
3.2.1. 씬 플로우

[BTP_씬플로우.pdf](#)

3.2.2. 와이어 프레임

⇒ 일부 UI는 요소별로 와이어프레임 작성 후 이미지를 제작하여 사용했습니다.
(모든 UI를 와이어프레임 작성 후 제작하려 했지만, 시간 관계상 일부만 와이어프레임 작업을 했습니다.)

⇒ 예시) 메모리아 선택지



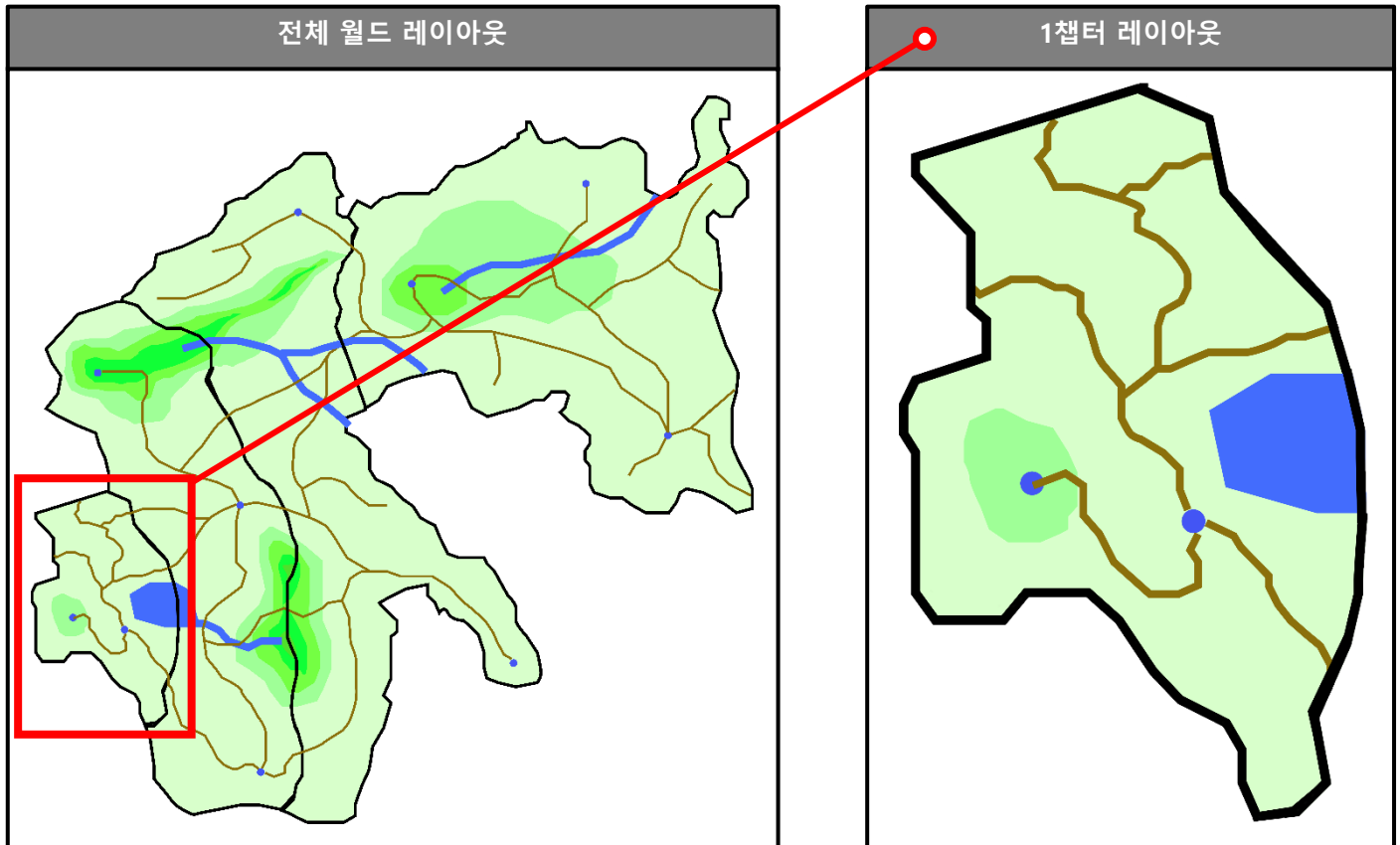
최종 결과물



3.3. 레벨

3.3.1. 레이아웃

⇒ 전체 레이아웃 중 프로토타입 구현 범위는 1챕터입니다.



3.3.2. 레벨 컨셉

- ⇒ 게임의 배경과 레벨 컨셉은 다크 판타지에 어울리게 어둡고 탁한 느낌을 줍니다.
- ⇒ 햇빛이 약하기 때문에 시든 나무들과 밝은 등불들이 레벨을 구성하는 중요한 요소가 됩니다.



<도로 예상 이미지>



<밝은 사당 예시 이미지>

4. 개발 과정 및 현황

4.1. 프로토타입 제작

4.1.1. 기본 조작

1) 이동 ([이동이 구현된 영상 링크 \[유튜브\]](#))

- ⇒ 키 입력에 따른 캐릭터 이동을 위해 '캐릭터 컨트롤러'를 사용합니다.
- ⇒ W/A/S/D로 이동합니다.
- ⇒ 캐릭터의 진행 방향이 바뀌면 캐릭터가 회전합니다.

2) 카메라 조작

- ⇒ 마우스 이동으로 카메라를 회전시킵니다.

⇒ 피드백 [1]

문제점	마우스가 게임 화면을 벗어나면서 카메라 회전의 조작감이 불편합니다.
해결 방안	Unity의 스크립팅 API중 하나인 CursorLockMode 사용 (Cursor.lockState = CursorLockMode.Locked)
수정 후	게임 플레이 시 마우스 커서가 사라지고 마우스는 화면 중앙에 고정

⇒ 피드백 [2]

문제점	UI창이 열려도 마우스 커서가 없고 마우스 이동에 따라 카메라가 회전
해결 방안	마우스 커서를 사용해야 하는 UI 창이 열리면 Cursor의 LockMode를 None로 변경
수정 후	일부 UI 창이 열리면 마우스 커서가 등장하고 마우스 이동에 따른 카메라 회전 중단

3) 점프 ([점프가 구현된 영상 링크\[유튜브\]](#))

- ⇒ 지형/Prop에 올라가거나 적의 공격을 회피하는 수단 중 하나로 점프를 사용합니다.
- ⇒ 캐릭터 컨트롤러의 스크립팅 API중 하나인 isGrounded로 지면에 있는 지 확인합니다.
- ⇒ Space 로 점프를 합니다.

⇒ 피드백 [1]

문제점	isGrounded의 정밀도가 낮아 지면 확인이 제대로 되지 않아 지면에 있음에도 점프 불가능
해결 방안	레이캐스트를 활용
수정 후	캐릭터 컨트롤러가 현재 충돌한 오브젝트의 레이어가 Ground라면 점프 가능

4) 대쉬

- ⇒ 대쉬는 적의 공격을 회피하기 위해 사용합니다.
- ⇒ 캐릭터가 보고 있는 방향으로 대쉬를 합니다.
- ⇒ Shift 키로 대쉬를 합니다.

5) 일반 공격 ([대쉬와 일반 공격이 구현된 영상 링크\[유튜브\]](#))

- ⇒ 캐릭터가 보고 있는 방향으로 일반 공격을 합니다.
- ⇒ 마우스 좌클릭으로 일반 공격을 합니다.

4.1.2. 깃허브 연동

- ⇒ 레벨 블록아웃 제작을 맡은 팀원과의 원활한 협업을 위해 깃허브로 버전 관리를 했습니다.
- ⇒ 깃허브 링크: https://github.com/hjs0913/BTP_Ver.1.0

⇒ 피드백 [1]

문제점	블록아웃 제작 중 용량이 큰 에셋을 사용하게 되어 깃허브에 업로드 불가능
해결 방안	깃허브의 LFS를 사용
수정 후	LFS로도 업로드 불가능

⇒ 피드백 [2]

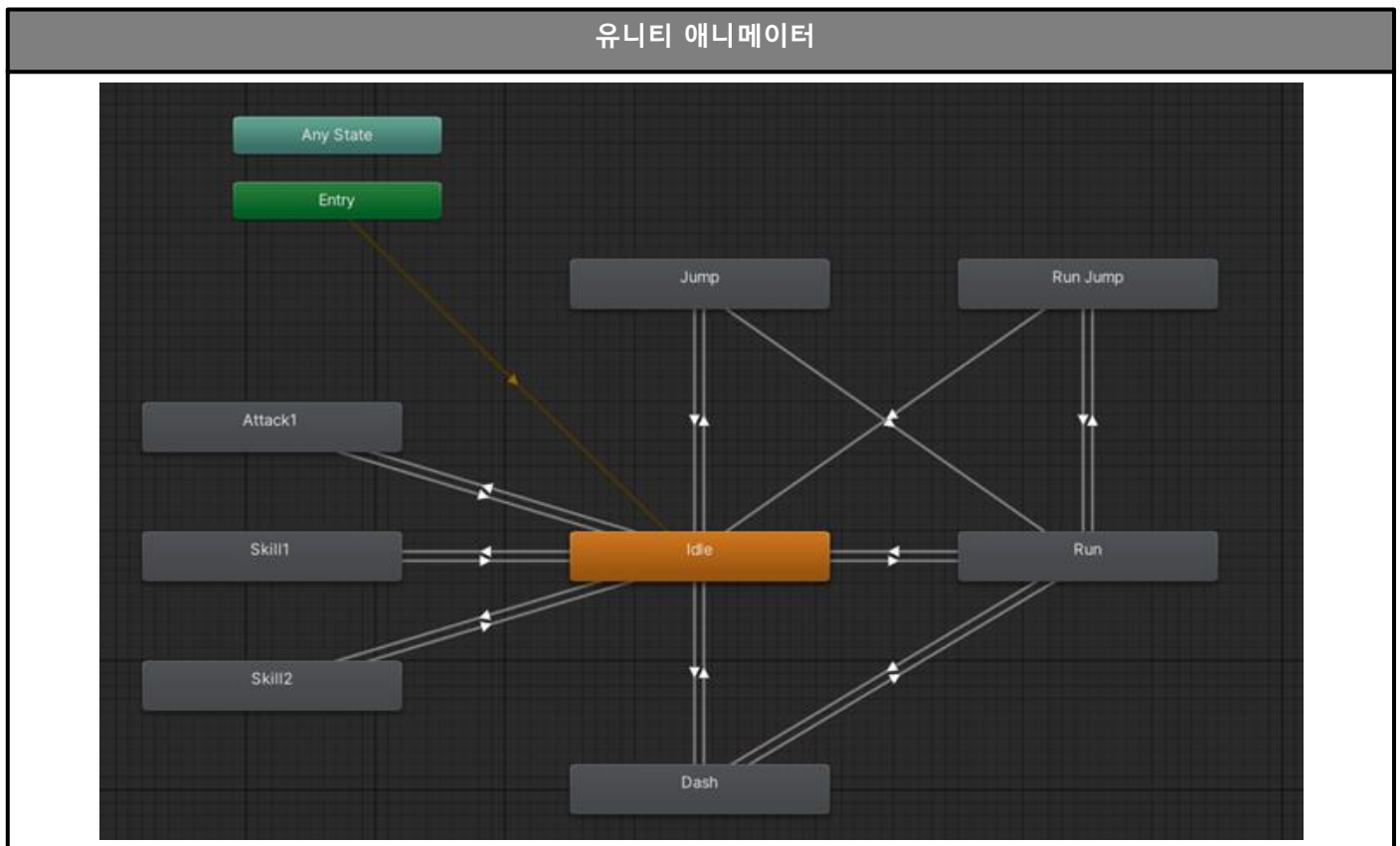
문제점	깃허브 레포지토리의 허용된 용량을 초과하는 에셋 사용
해결 방안	1. 용량 조절 - 사용하지 않은 모델/텍스처를 삭제하여 용량 조절 - 모델의 텍스처 크기를 줄여서 사용
수정 후	깃허브에 업로드 완료

⇒ 피드백 [3]

문제점	branch를 main에 merge하는 과정에서 유니티 파일의 fileID 중복 오류로 conflict 발생
해결 방안	1. conflict이 발생한 부분 하나 하나 찾아가며 수정 -> fileID가 뭘 의미하는 지 상세히 확인할 수 가 없어서 제대로 수정이 불가능 2. 블록 아웃 작업한 씬을 패키지 파일로 추출한 후 패키지 импорт
수정 후	2번 방안으로 수정 후 main에 merge 완료

4.1.3. 모델 및 애니메이션

- ⇒ 모델과 애니메이션은 Mixamo와 Unity Asset Store에서 게임의 컨셉과 비슷한 에셋을 찾아 사용합니다.
- ⇒ 애니메이션 전환을 위한 트랜지션을 설정합니다.



4.1.4. 전투

1) 플레이어

- ⇒ 플레이어가 공격을 하면 공격 애니메이션이 재생됩니다.
- ⇒ 공격 애니메이션의 재생 시간 중 일부 구간에 충돌 처리가 가능한 박스 콜라이더를 활성화합니다.

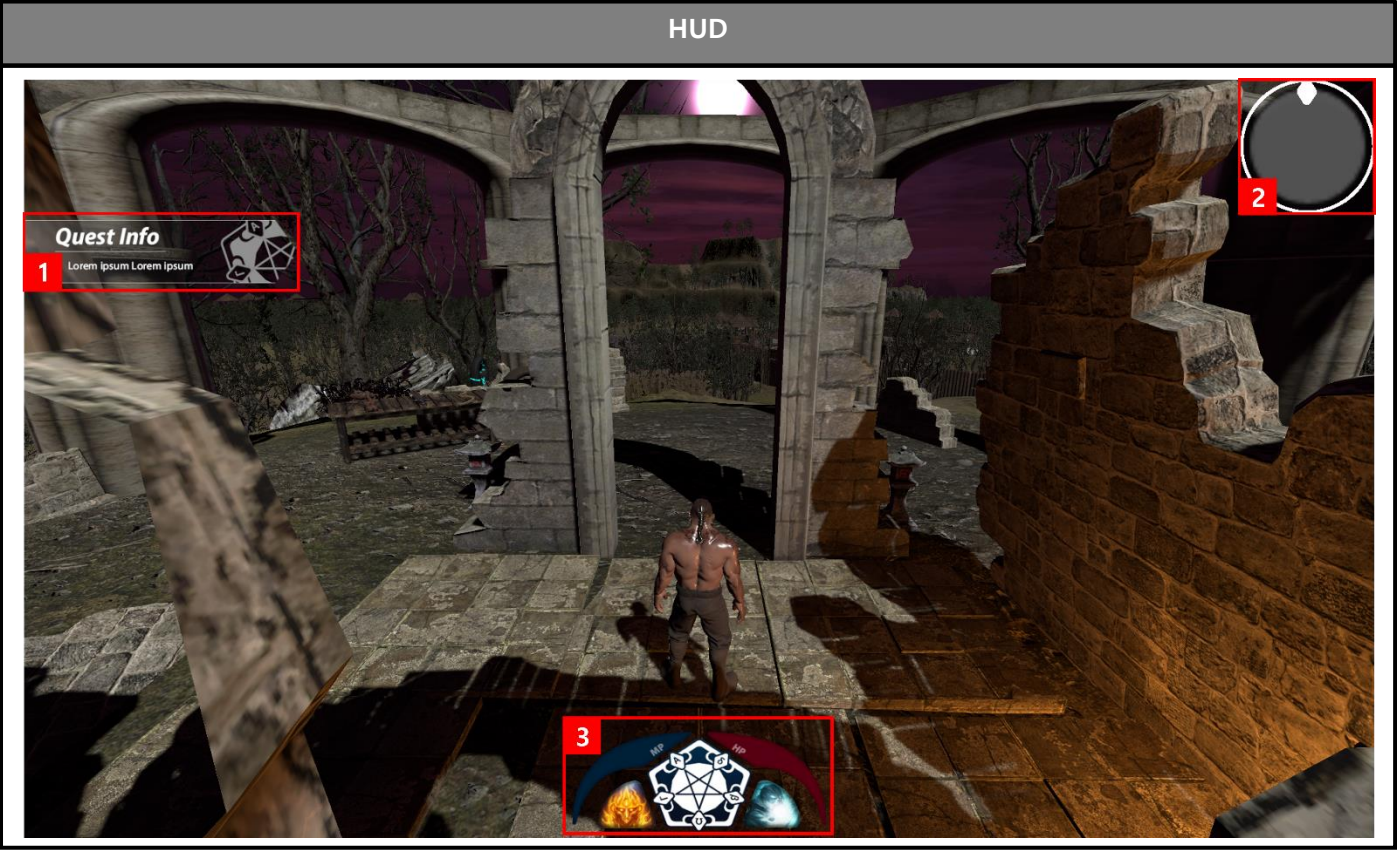


2) 몬스터

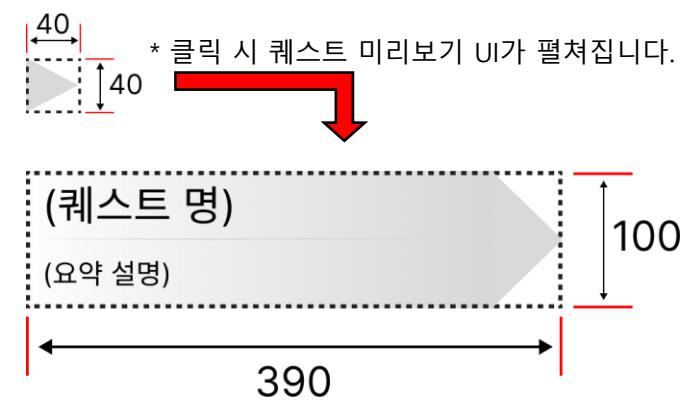
- ⇒ 몬스터는 IDLE 동작을 취하다가 체력이 0이 되면 사망합니다.
(몬스터의 AI는 해당 프로토타입 버전에서는 구현하지 않았습니다.)

4.1.5. UI

1) HUD



No	UI	설명
1	퀘스트 미리보기	퀘스트 창에서 등록한 퀘스트의 정보를 미리 볼 수 있습니다.
2	미니맵	
3	HP/MP	
	간극/고유 스킬 게이지	
	스킬 아이콘	



2) 메모리아 선택 창

3) 상호작용

4) 캐릭터 정보 창

1
LEVEL

Baren

ALTER EGO | VALIUS
WEAPON | HALBERD
EXP | 77.5%

MENU

STATUS

MEMORIA

INVENTORY

MAP

QUEST

· Status

체력	312	명중	120
물리 공격	20	회피	34
물리 방어	28	치명타율	32
마법 공격	0	가드율	10
마법 방어	0	행동 속도	25

· Active Skill

· 대쉬 ·

· 액티브 ·

· 체벌 ·

2
AVATAR

1
LEVEL

Baren

ALTER EGO | VALIUS
WEAPON | HALBERD
EXP | 77.5%

MENU

STATUS

MEMORIA

INVENTORY

MAP

QUEST

Halberd

Info

발리우스에 해당되는 헬버드는 뛰어난 힘과 헬버드의 큰 크기의 금속 날을 이용해 적을 찢고 벤다. 일정 확률로 적에게 출혈피해를 입힐 수 있으며, 근접전에서 유리한 포지션이다.

2
AVATAR

5) 퀘스트 창

6) 월드맵

7) 게임 메뉴

4.1.6. 트리거

1) 무기 선택

⇒ 캐릭터 생성 후 무기를 선택하면 해당 계열의 알터 예고가 정해집니다.

2) 예언의 석판

⇒ 플레이어 캐릭터가 '예언의 석판' 근처에 가면 상호 작용 UI가 뜨고 상호 작용이 가능해집니다.

⇒ 상호작용을 하면 메모리아를 선택하는 UI창이 뜨게 됩니다.

⇒ 피드백 [1]

문제점	메모리아를 선택한 다음에도 '예언의 석판'이 상호 작용 가능한 것처럼 보임
해결 방안	메모리아를 선택하면 상호 작용한 석판의 머터리얼을 변경
수정 후	해당 오브젝트 비활성화

4.1.7. 추가 기능

1) 이전 위치 리스폰

⇒ 게임 메뉴 '로비로 가기' 후 '데이터 로드'시 이전 위치에서 스폰됩니다.

⇒ 원래는 데이터 테이블과 연결해야 하지만, 해당 프로토타입 버전에서는 씬이 전환되어도 사라지지 않는 'DontDestroyOnLoad' 오브젝트를 임시로 활용합니다.

5. 평가 및 업데이트 계획

5.1. 평가

프로토타입을 제작할 시간이 부족하였기 때문에 게임 시연에 필요한 기능을 우선 구현했습니다. 이로 인해 초기에 생각한 모든 기능을 넣지 못했습니다.

5.2. 후속 계획

1) 카메라 처리

⇒ 벽에 닿으면 카메라가 자동으로 캐릭터에 가까워지는 기능을 추가합니다.

2) 락온(대상 고정)

⇒ Q 키를 눌러 화면의 중앙에 가장 가까운 대상에게 락온을 합니다.

⇒ 락온을 하면 화면이 대상에게 고정됩니다.

⇒ 이때 캐릭터가 좌/우로 움직이면 정면을 바라보며 옆으로 걷는 모션이 나옵니다.

3) 탈 것/라이딩 스킬

⇒ V 키를 눌러 라이딩 스킬을 사용합니다.

⇒ 탈 것이 등록되어 있어야 사용 가능합니다.

⇒ 넓은 세미 오픈월드의 맵을 탐험하기 위해서는 라이딩 스킬이 필요합니다.

4) 몬스터 AI

⇒ 전투를 경험할 수 있게 몬스터 AI를 추가합니다.

5) 퀘스트

⇒ 퀘스트가 진행되도록 퀘스트 시스템을 추가합니다.