

221013 VPS AR 1팀 보고서

AR 하나

기획 현황

프로토타입 기준으로 작성

1차 프로토타입 빌드 10.28일까지 구현

- 게임 및 캐릭터 이름 수정

- AR 토모다치 → AR 하나
- 캐릭터 컨셉에 맞춰 한국과 일본에서 통용될 수 있는 이름으로 설정

- 핵심 기능 기획 & 구현

캐릭터와 데이트 코스를 따라갈 수 있게 만들어 실제 데이트하는 느낌을 유저들에게 제공

- 공간 인식

- 캐릭터 구현

- 캐릭터 디자인
- 캐릭터 컨셉
- 캐릭터 출력
- 캐릭터 이동
- 캐릭터 모션

- 스토리 모드

- 데이트 코스 선택
- 데이트 포인트 지정

- 데이트 진행

- 상호작용

- 지정 장소 인식
- 지정 장소에서 상호작용
- 캐릭터와 대화에 따른 VPS 연출
- 방명록

- 숨겨진 이야기

- 프롤로그
- 호감도 1 스토리

- UI

변경된 사항

1. 기존에 데이트 코스를 유저가 짜고 네비게이션 안내를 스토리 진행에 따라 미리 설계된 코스 중 고르게 하여 네비게이션 기능 삭제
 2. 네비게이션을 사용하지 않지만 경로를 미리 설계 하여 유저의 동선 안내, 유도를 화살표 혹은 가야할 길에 색상을 열게 입히는 식으로 처리 할 예정
-

캐릭터 컨셉

- 한국으로 유학 온 일본인 여대생 컨셉
 - 캐릭터 FBX와 블렌더를 이용하여 직접 캐릭터를 제작
 - 모션(애니메이션과 표정 등) 구현 예정
 - 모션의 종류는 유니티 anim 파일로 리깅을 잡아 제작
- 15~30 종류 (걷기 3종 , 코너 3종 , 애교-상호작용 등등 9종) 기획 예정

1안

블라우스 혹은 얇은 니트와 A라인 스커트를 입은 캐릭터

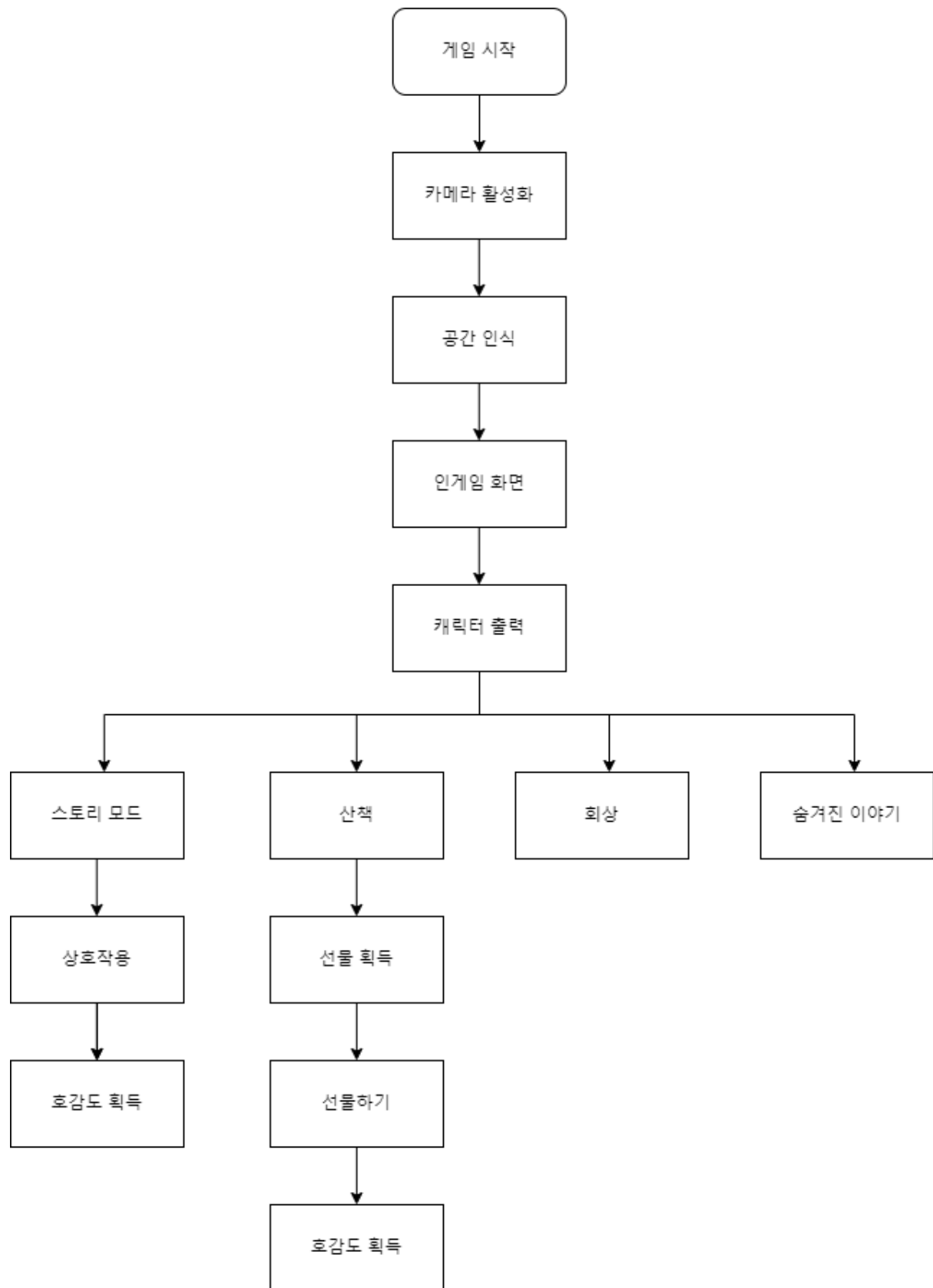


2안

레이어드 원피스와 크로스백을 착용한 캐릭터

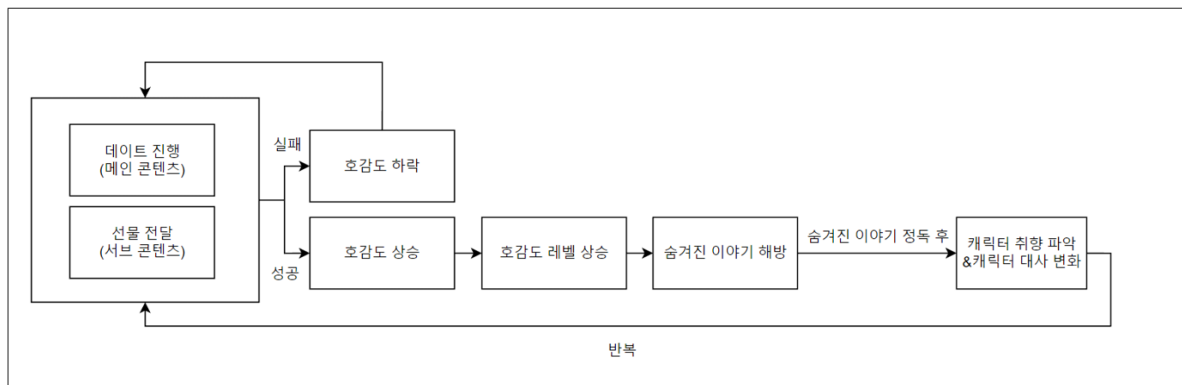


컨텐츠 플로우



스토리 모드 및 상호작용 시스템

- 기존의 콘텐츠 '데이트' 명칭을 '스토리 모드' 로 변경
- 세이브 앤 로드 기능 구현에 많은 시간이 소요될 것이라 예상
- 프로토타입 빌드에서는 맵 데이터를 불러와 미리 설정한 경로와 특정 지점을 통해 플레이 하도록 기획



기획 의도

- 연애 시뮬레이션에 맞게 플레이어가 NPC와 연애적 교감할 수 있는 주 콘텐츠가 필요함
- AR에서만 가능한 화려한 VPS 데이트 연출을 보여주며 플레이어에게 다양한 볼거리를 제공

기획 의도

- 기존에 유저가 직접 경로를 선택하고 안내하는 방식 구현이 불확실
- 변수
 - 유저가 경로 이탈 시
 - 다양한 경로에서 상호작용을 어떻게 할 지
 - Google Maps API의 단점을 어떻게 보완할지
 - 네비게이션 R&D의 어려움 보완

진행 방식

- 경로 1과 경로 2를 직접 제시하여 플레이어에게 선택
- 경로 선택 후 첫 번째 장소로 이동하라는 UI 출력
- 해당 장소로 이동 후 데이트 시작
- 캐릭터가 다음 장소로 가이드
- 다음 장소에 도착 후 그 다음 장소로 유도하는 UI 출력
- 도착 후 반복

→ 유저는 정해진 경로를 따라 가야함

데이트 상호작용 유형

취향 맞추기

"숨겨진 이야기" 등 여러 콘텐츠에서 파악한 캐릭터의 취향에 맞는 선택지를 골라야 함

성공 조건

- 선택지의 Type이 NPC 의 세부 취향 중 Like와 일치해야 함
- 선택지의 리소스 Type&캐릭터 세부 취향=성공 여부(성공=True/실패=False)

코드 (캐릭터 취향 / 선택된 아이템)

- Case1. Like&Like=True
- Case2. Like&Hate=False
- Case3. Hate&Hate=False

진행 방식

- 건물 인식 UI 출력
 - 건물 Date Type 판단
 - 관련 NPC 애니메이션 및 대사 출력
 - 건물에 입력된 선택지 GUI 출력
 - 플레이어 선택 후 해당 선택지의 Type 판단
 - 해당 이벤트의 성공 조건 판단
 - 성공 or 실패에 따른 호감도 수치 상승or하락 및 관련 애니메이션 출력
-

방명록

건물 벽이나 오브젝트에 이름 혹은 낙서 남기기

성공 조건

건물 Mesh에 적용

진행 방식

- 건물 인식 UI 출력
 - 건물 Date Type 판단
 - 관련 캐릭터 애니메이션 및 대사 출력
 - Touch & Draw할 수 있는 화이트보드 UI 출력
 - Touch & Draw 진행
 - 해당 글씨 에셋화
 - 건물 벽 상세 위치 지정 안내 UI 출력
 - 해당 에셋 건물 Mesh에 적용
-

특별 VPS 이벤트

Date Type에 따른 성공 조건을 만족 시 100% 확률로 등장하는 특별 VPS 연출 이벤트

진행 방식

- 특별 VPS 이벤트 안내 UI 출력
- 해당 위치에 있는 VPS 연출 Type 판단
- 관련 NPC 애니메이션 및 대사 출력
- VPS 연출 재생
- 10초 재생 후 나가기 UI 출력

개발 현황

Type	Context
R&D	Google AR Anchor
R&D	안드로이드 데이터 세이브 로드
R&D	Google AR 샘플 앱 탐색 및 코드 분석
R&D	샘플 앱 ◦ Depth Lab ◦ 광화담
R&D	GPS + Firebase
R&D	ARCore Depth Lab
R&D	ARCore Depth Lab 아바타 관련 프로젝트
R&D	ARCore 콜라이더 관련 기능 테스트
R&D	Depth Lab 오브젝트 콜라이더 관련 스크립트 적용
R&D	AR Depth Occlusion ◦ 오브젝트가 물체 뒤에 위치 했을때 가려지는 경계 면을 매끈하게 처리하는 방법 연구 - 매트리얼을 응용해 Plane의 경계면은 매끄럽게 처리 가능
R&D	GoogleMap API 건물 렌더링 기능 + Occlusion 기능 결합 ◦ 기존 건물 렌더링에 Depth Occlusion R&D시 알게 된 매트리얼 기능을 활용. - 건물 렌더링은 투명하게 바뀌었으나 명일 외부 테스트 필요함.
R&D	오브젝트 상호작용 ◦ 화면 터치 레이캐스트 이슈 있음 - 간혹 터치 인식이 되지 않는 현상 발생. - Plane상 오브젝트를 터치로 제어하는 기능은 조금 더 개선해봐야 할 것으로 판단
R&D	UI 상호작용 ◦ 상호작용 구현 완료 되는대로 금방 적용 가능할 것으로 판단
R&D	• ARCore Depth 활용, AR 공간 내 캐릭터 삽입 및 애니메이션 작동 확인 ◦ 버튼을 터치 시 화면 중앙의 Plane 위로 포인터 위치 및 캐릭터 이동 - 해당 이동 방법 기준 진행도 85% (다른 이동 방법으로 구현 시 20%) - 기획서 상의 볼륨을 짐작하기 어려운 관계로 보류 결정
R&D	• 구글맵스 API를 이용한 건물 렌더링 기법 탐구 ◦ 회전 및 오차범위를 보정하기 위한 지속적인 테스트 요망

Type	Context
R&D	<ul style="list-style-type: none"> • GPS + Firebase 데이터 저장 및 불러오기 ◦ GPS 위도와 경도, 오브젝트 프리팹을 DB에 저장 ◦ 저장된 데이터를 불러와 정상적으로 배치되는 지 확인할 예정 ◦ 추가적인 Markless AR R&D 예정
R&D	<ul style="list-style-type: none"> • 스마트폰터치 레이캐스트 테스트 ◦ 구조 파악은 일정 부분 완료, 본 프로젝트에 도입하기 위한 모듈 ◦ 프리팹 화를 염두에 두고 새로운 프로젝트 생성, 추가 테스트 예정 ◦ 오브젝트 터치 시 유동적인 이벤트를 발생시킬 수 있도록 기능 구현 예정 ◦ 오브젝트 상호작용 완성시 UI와 오브젝트 상호작용, 오브젝트 이동, 파괴, 회전 등 물리적인 부분을 추가적으로 테스트 할 예정 ◦ 텍스트UI 활용. 프리팹 화 하여 모바일 환경에서 디버깅을 쉽게 할 수 있도록 모듈 개발 예정
R&D	<ul style="list-style-type: none"> • UI 상호작용 동작 테스트 완료 시 ◦ 인벤토리, 아이템 획득, 아이템 사용에 관한 설계방안을 개발팀 내부적인 회의를 통해 결정할 필요성 느낌
R&D	바닥 면 Plane 인식, AR 카메라 높이 지정
R&D	<ul style="list-style-type: none"> • 벽면과 바닥을 각각 다른 Plane으로 인식 R&D ◦ 가능하다면 실내에서도 안정 정인 Occlusion 가능 예상 ◦ 매트리얼을 활용. 건물 렌더링을 투명하게 바꾸는 것 성공
R&D	그림자 처리
Task	코드 컨벤션 문서 작성
Task	안드로이드 빌드 기준 InputManager 스크립트 모듈 선 제작 완료



- 가상의 벽면을 설치하고 오브젝트가 벽면을 인식하여 해당 벽면 뒤로 이동했을 때 안 보이도록 구현 완료
- 회전 구현은 아직 미 구현