

2020년 1학기 캡스톤디자인 설계최종보고서

BRAIMMO, 알고리즘 기반 모바일 게임 개발

- 과목명: 캡스톤디자인
- 과목 담당교수 : 조성배
- 연계 기업명 : 소프트웨어 팩토리
- 제출일: 2020년 12월 15일
- 프로젝트 팀원명단:

팀원명	학번	제1 전공 (심화전공)	제2전공	담당 업무 팀장부터 기입
홍원표	21701065	컴퓨터공학심화		팀장
임예찬	21400636	전산	전자	팀원

- 캡스톤 설계 완성도

점 수	캡스톤 작품 구현 완성도	
	설명	지도교수 평가 해당난에 체크(O)
5	상용 제품으로도 손색이 없는 수준	
4	모든 기능이 잘 작동하고 최적화도 어느 정도 됨	
3	전반적인 기능 원활히 작동 함. 성능은 기존 제품이나 논문에 비해 미흡한 편	
2	일부 기능만 구현하거나 동작 중 오류가 가끔 발생	
1	다수 기능이 미구현되었거나 또는 미작동함	

날짜: 2020-12-15

지도교수명: 조성배

(서명)

설계 결과 요약서						
설계주제 (Title)	BRAIMMO, 알고리즘 기반 모바일 게임					
주제어 (Keywords)	알고리즘 / 코딩 / AI / RPG / 코딩 교육 / BRAIMMO / 브레이모					
설계팀원명단	홍원표, 임예찬					
지도교수, 산업체 자문위원(Advisor)	조성배 교수님 소프트웨어 팩토리 김민진					
설계 기간(Period)	2020-06-17 ~ 2020-12-18					
설계문제의 정의 (Problem Statement)	<p>블록코딩은 교육목적에 과하게 치중되어 있어서 확장성 부족, 실용성 부족 기존 코딩 교육 기능성 게임들은 몰입도나 성취감이 부족하여 마켓에서 순위권 진입 불가.</p> <p>기존 롤플레이팅 장르 게임은 유저들의 직접적인 게임 개입을 권장하기 위해 제한된 AI 만 제공하는 트렌드. 이에 유저들은 멍청한 캐릭터 알고리즘에 불만을 호소.</p>					
설계요소 (Design Elements) (해당요소에 O표)	목표설정	분석 및 개념설계	상세설계	구현 및 제작	시험 및 평가	기타
	O	O	O	O	O	
제한조건 (Constraints) (해당요소에 O표)	제작비용 및 기간	환경	사회 및 윤리	안전 및 미학	산업표준	기타
	O	O	O	O		
설계결과물 (deliverables)	Unity Build 를 통해 apk, exe 파일 생성 한국저작권위원회를 통해 저작권 획득					
설계 결과의 요약 (Abstract) (100자 이내)	<p>교육과 게임이 합쳐진 '기능성 게임'의 단점인 '게임 몰입도 부족'을 보완하기 위해 알고리즘 교육에 최근 모바일 게임에서 가장 영향력 있는 카테고리인 RPG 를 결합.</p> <p>자연스러운 알고리즘 교육을 위해 프로그래밍 언어와 유저의 직접적인 교류 지향. 클러스터 매핑 알고리즘 설계를 통해 캐릭터 AI 구축.</p> <p>IoT, AI 의 실시간 대처 능력의 핵심인 Arduino Style 알고리즘을 이용하여 RPG 캐릭터의 움직임과 대처능력 고안.</p> <p>게임에서 필요한 홈 페이지, 캐릭터 페이지, 보상 페이지, 게임 선택 페이지, 인 게임 페이지 등 각종 Scene 구현.</p> <p>한 계정에 따른 모든 정보를 Json Format 을 이용하여 데이터베이스 관리.</p>					

Summary **Date: 2020-12-15**

Course Name	Capstone Design	Year/Semester	2020 / 02
Design Title	BRAIMMO, the educational RPG based on Algorithm Coding		
Keywords	Algorithm / Programming / AI / RPG / Education / BRAIMMO		
Designers (stdudents)	Wonpyo Hong Yechan Lim		
Advisor(s)	Sungbae Jo, Software Factory Minjin Kim		
Design Problem Definition	<p>Since Block Coding is weighted heavily on education, it has limits on its scalability and practicability. New approach is on demand.</p> <p>Since current coding educational games lack senses of immersion and accomplishments, it is nearly impossible to enter market top charts.</p>		
Design Constraints	<p>The development must be finished by the end of 2020</p> <p>The design of the game character and UI must be pretty enough to have at least 60% of test-users' positive vote on game design.</p> <p>The algorithm creating page must be educational enough that at least 60% of test-users vote that they could understand the basics of programming through experiencing our product.</p>		
Design Objectives	<p>The product should deliver the product through a form of a mobile app that can be downloaded from either Appstore or Google Play.</p> <p>The product should have enough stages and items to create an attractive game environment.</p> <p>The algorithm creating technology should have a capability to be applied to other IoT devices such as drones or smart rice cookers.</p> <p>The algorithm creating technology we create should be designed with various visual aids so that it is easy enough for users to understand the algorithm without external aids.</p>		
Deliberables	<p>.apk and .exe file through Unity Build</p> <p>Copyright by Korea Copyright Commission</p>		
Extended Abstracts	<p>To meet both ends of 'serious games' and 'RPG', the combination of the two could create a new genre which could result in success of both complex AI and immersion.</p> <p>To create the algorithm creating technology so that users can understand how it functions by intuition, it will have a form of cluster map, which is a widely known way to express an idea.</p>		

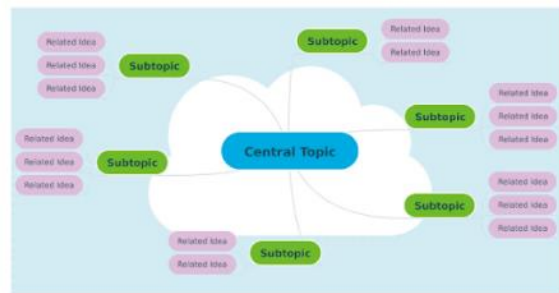
● Level Descriptor (문제수준 설명)

번호	문제의 속성	문제수준 설명	체크란
		컴퓨팅 문제란 아래 속성들 중 <u>일부</u> 또는 전부를 갖는 컴퓨팅 문제이다.	
1	상충되는 요건의 범위	상충되는 기술적, 컴퓨팅적 요건과 그 외 다른 상충적 요건을 포함한다.	
2	요구되는 분석의 깊이	명백한 해답이 없으며, 적절한 추상적인 모델을 수립하기 위해 개념적인 사고와 독창적인 분석을 요구한다.	√
3	요구되는 지식의 깊이	교과과정을 통해 습득한, 깊이 있는 컴퓨팅 지식 또는 특정 분야의 지식 그리고 확립된 이론에 근거한 분석적인 방법이 사용된다.	√
4	논점의 대상	흔히 다루지 않는 논점을 포함하고 있다.	
5	문제의 수준	전문적인 컴퓨팅에 요구되는 표준적인 방식과 일반적인 실무 절차로 해결되지 않는다.	
6	이해당사자들의 요구 수준 및 범위	다양한 이해당사자의 서로 다른 요구사항을 고려한다.	√
7	영향력	문제해결이 광범위한 분야에 중요한 영향을 미친다.	√
8	상호의존성	상호 의존하는 구성요소들 또는 많은 하위요소들로 구성되어 있다.	√
9	문제의 명확성	문제의 요구조건 또는 필요성이 모호하거나 명백하게 기술되어 있지 않다.	

0. Executive SUMMARY

초등학생의 모바일 게임 시간은 주 15.6 시간이라고 한다. BRAIMMO 는 이 16 시간을 코딩을 공부하는 시간으로 활용하고자 한다. 현재 초등학교, 중학교에서 코딩 교육이 의무화 됨으로써, 코딩 교육이 중요해지고 있다. 이에 미국 및 전세계에서는 코딩 교육을 위해 기능성 게임을 활용하는 사례가 늘어나고 있다. 예시로 마인크래프트는 교육용 게임으로 개발되어 전세계에서 수학, 과학, 코딩 수업 교재로 활용하고 있다. 또한 기능성 게임을 활용한 교육 시장도 굉장히 커지고 있다. 현재 90 억달러인 시장은 2025 년 300 억달러까지 성장할 전망이다. 그러나 현재 국내 코딩 기능성 게임은 문법 교육 위주다.

문법 교육 게임은 사실 재미가 없어 교육 효과가 떨어진다. 학습은 몰입을 바탕으로 시작되고, 몰입은 재미가 있어야 가능하기 때문이다. 우리는 BRAIMMO 의 재미와 몰입성을 위해 어떤 게임이 가장 많이 플레이 되는지 조사하였다. 그 결과 롤플레이팅, 즉 RPG 가 가장 인기 있다는 것을 알 수 있었는데, 그래서 우리는 BRAIMMO 를 RPG 장르로 개발해 검증된 재미를 확보했다. 캐릭터도 똑똑해지고, 사용자도 자연스럽게 알고리즘을 배우는 게임, BRAIMMO 는 캐릭터의 알고리즘을 직접 코딩하고, 코딩한 대로 캐릭터가 적을 무찌르며 스테이지를 클리어 해나간다. 우리는 직관적으로 알고리즘을 보여주기 위해, 클러스터 매핑을 개발했다.



<전 연령층에게 익숙한 마인드 맵 형식>



<브레이모 게임에 사용되는 코드수정창>

클러스터 매핑은 레이블 - 액션 - 조건의 구조로 구성 되어있다. 사용자는 레이블 밑에 캐릭터가 실행할 액션을 연결하고, 액션 밑에는 액션이 발동되는 조건을 연결한다, 게임은 매 프레임마다 이 알고리즘을 체크하면서 조건이 만족되는지 확인하고 액션을 실행하는데, 이 과정에서 코딩의 기본인 반복문, 조건문, 논리연산을 배울 수 있다.

레이블

이동 / 공격 / 아이템으로
카테고리화



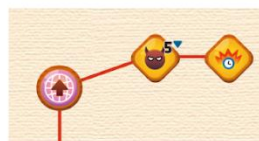
액션

각 레이블에 해당하는 액션



조건

액션이 게임 내에서
실행 될 수 있는 모든 조건



<AND 조건>



<OR 조건>

우리 린캔버스를 소개한다.

Braimmo 의 주 타겟층은 코딩을 배우고 있는 초등학생, 중학생들과 이들의 학부모, 그리고 기존 모바일 게임 유저다. 우리는 현실을 안 해도 게임하는데 문제없는 게임을 지향한다. 그래서 아이템 구매, 캐릭터 스킨 같이 게임 플레이에서 지장이 없는 부분을 유료화해서 수익을 얻을 예정입니다. 손익분기점으로 한달 만명의 유저를 목표로 하고 있다. 우선 앱스토어에 런칭하여 모바일 게임 유저를 확보하고, 인지도가 쌓이면 웹 기반의 교육용 콘텐츠를 제작해서 학원과 학교 같은 교육 기관으로 타겟을 확장할 예정이다.

Lean Canvas

Problem	Solution	Unique Value Proposition	Unfair Advantage	Customer Segments
국내 기능성 게임의 오락성 부재	AI 알고리즘을 직관적으로 보여주는 기술 개발	게임에 몰입하면서 자연스럽게 알고리즘 학습	현재 게임과 교육 트렌드를 잘 반영할 수 있는 대학생으로 모인 개발진 보유	코딩 의무 교육 정책으로 알고리즘 학습이 필요한 초등학교
게임의 몰입도가 낮아 학습률이 떨어짐	게임의 재미와 몰입도를 위해 RPG 장르 도입	알고리즘이 정교해짐에 따라 캐릭터AI가 똑똑해짐	프로토타입이 아닌 이미 개발된 버전 1.1 보유	중학생 자녀에게 코딩 및 알고리즘 교육을 권하고 싶어하는 학부모
현재 알고리즘을 집중 학습할 수 있는 게임 부재	Key Metrics 게임 유저수 아이템 수익률		Channels 유튜브 소셜미디어 (페이스북, 인스타그램)	기존 모바일 게임 유저들
Existing Alternatives				
Sprite Box, lightbot, 스크래치, 엔트리				
Cost Structure	Revenue Streams			
광고 비용 - 총 50,000,000원 / 제품 홍보 영상 제작, 온라인 마케팅 대행	캐릭터 생성 제한 수 1 증가 = \$9.99			
제작 비용 - 총 42,000,000원 / 웹 개발 및 디자이너 인건비, 서버 구축	아이템 랜덤박스 (무작위 등급의 아이템 5개) = \$9.99			
출원 비용 - 총 5,000,000원 / 국내 특허 및 PCT 특허	캐릭터에 착용 가능한 각종 스타일, 스킨 판매			

(Business Model 참고)

Business Model

Key Partner	Key Activities	Value Proposition	Customer Segment	Customer Relationships
교육부 국내 / 해외 교육 시설	코딩 교육을 위한 논리적 사고 배양 무수한 게임 상황을 위한 꾸준한 액션, 조건 노드 업데이트	코딩 이해력 강화 코딩 진입장벽 낮춤 정보과학적 사고력 향상	초중고생 학부모 교육기관 모바일 게임 유저	Get -> Free to Play Get -> Free to Download Keep -> Engaging Game Grow -> 새로운 스토리 콘텐츠 Grow -> 새로운 조건 노드
Key Resources		Channels		
App Developers Graphic Designers		App Store Google Play 학교 / 교육 기관		
Revenue Streams			Cost Structure	
캐릭터 생성 제한 수 증가 - \$9.99 per Slot 아이템 매출 - \$9.99, \$29.99 등 아이템 패키지 판매 스킨 매출 - 캐릭터에 착용 가능한 각종 스타일, 액세서리 판매			광고 비용 - 교육 기관 방문 마케팅 및 각종 광고비 ₩2,700,000 제작 비용 - 서버 구축 및 유지 (근 2년) ₩3,120,000 출원 비용 - PCT 특허 ₩9,460,000	

1. PROJECT OVERVIEW (3~4 페이지 분량으로 작성)

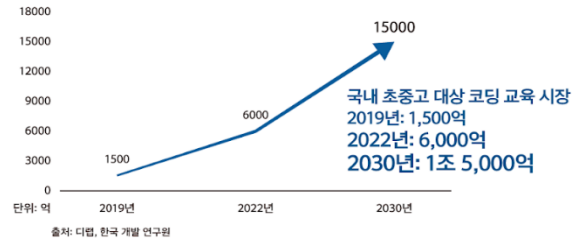
1.1 Introduction

사업개요 국내 코딩 교육 열풍

초중등학교 소프트웨어 교육필수화		
학교	시기	교육시간
중학교	2018년 실시	3년간 총 34시간 이상
초등학교	2019년 실시	5~6학년 대상 총 17시간 이상

- 국내 코딩 교육 의무화에 따라 코딩 공교육, 사교육 증가
- 초중고생을 타겟으로 하는 코딩 교육 소프트웨어 시장 증가

국내 초중고 대상 코딩 교육 시장 규모 전망



초중고 과정에서 알고리즘 및 코딩 교육 의무화 & 코딩 교육 시장의 규모 증가

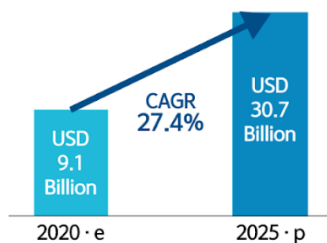
코딩 교육에 초중고 학생들이 즐길 수 있는 게임을 융합시킨 '알고리즘 / 코딩교육컨텐츠'

현재 IT 기술의 발전으로 인해 맞이한 4차 산업 시대에는 각 산업 분야의 경계선이 불분명해지고 융합되는 형태로 변화되고 있다. 이에 따라 교육부는 IT 기술에 익숙한 미래 인재를 양성하기 위해 초중고 과정에서 알고리즘 및 코딩 교육을 의무화하였고, 사교육 시장에서도 그 비율이 증가하고 있다. 우리 팀, LnP는 이런 흐름에 발 맞춰 코딩 교육에 초중고 학생들이 즐길 수 있는 게임을 융합한 '코딩교육게임'이라는 창업 아이템을 통해 전 연령층이 코딩에 쉽고 재밌게 접근할 수 있도록 진입장벽을 낮출 것이다.

기능성 게임(Gamification) ?

게임을 교육, 스포츠, 의료 등 다양한 분야와 접목하여 오락 및 여타 목적을 추구하는 게임 콘텐츠

Attractive Opportunities in the Gamification Market



출처: Secondary Research, Expert Interviews, and MarketsandMarkets Analysis 출처: Analysis of Gamification Models in Education Using MDA Framework

교육이 Gamification에서 차지하는 비율



Gamification 시장의 꾸준한 증가세

국내, 전세계 모바일 시장이 급격하게 성장하고 있습니다. 또한 2025년까지 전세계 교육 gamification 시장이 90억달러에서 300억달러까지 성장할 전망입니다.

기능성 게임은 놀이와 즐거움이 주된 목적이 아닌 교육을 주된 목적으로 삼는 게임이다. 모바일 게임 시장이 현재 게임 이용자에게 가장 큰 비중을 차지하는 가운데, 여러 가지 교육의 목적성을 탑재한 기능성 게임들이 출시되고 있지만 실제 기능성 게임의 플레이스토어나

앱스토어의 점유율은 매우 낮다. 이에 카이스트에서 2007 년부터 기능성 게임을 연구해온 우탁 박사는 기능성 게임들은 너무 많은 교육 요소를 넣는 과정에서 게임의 중요한 요소인 몰입성이 사라진다고 말하였다. 이에 LnP 는 현재 모바일 앱 스토어의 동향을 관찰하여, Top-Selling 게임들의 장르를 파악하고 게임의 몰입성을 살리면서도 알고리즘을 학습하고 있다는 사실조차 모르게 배울 수 있는 모바일 게임을 개발하는 데에 목표를 두고 있다.

현재 초등학교에서 사용되는 코딩 교육 프로그램은 엔트리, Code.org, 스크래치, 앱인벤터 등이 있습니다. 이 프로그램들은 공통적으로 블록 코딩에 기반을 두고 있습니다. 하지만 초등학교 코딩선도교사들의 인터뷰 통계 결과, 블록 코딩은 교육목적에 과하게 치중되어 있어서 정의된 블록 외에는 확장성이 부족하고 실용적 측면에서 현재 실무에서 사용되는 프로그램 언어와는 동떨어져 있다고 밝혀졌습니다. 무엇보다 교육용 프로그램이라 딱딱하고 재미가 없습니다.

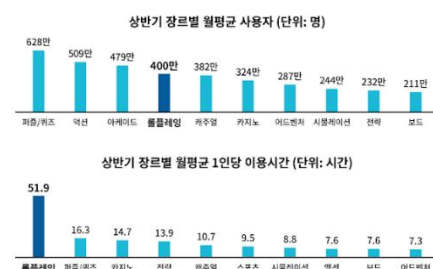
코딩 교육 게임화 사례
대표적인 게임을 활용한 프로그래밍 블록코딩



2020 년 구글 플레이스토어의 교육관련 인기 게임 앱 200 가지를 조사한 결과 알고리즘 코딩에 관련된 앱이 하나도 없었다. 그래서 조사 모집단을 특정하여 코딩 교육 목적을 가진 모바일 앱 중 대표적인 게임들을 조사해봤다. 대표적으로 SpriteBox 는 10 만 이상 다운로드에 평점 4.3 을 가진 게임이지만, 재미가 없다는 평이 태반이었고, 실제 플레이 한 결과 2D 캐릭터를 원하는 지점에 움직이기 위하여 블록을 생성하는 플 레이 방식이었다. 이는 사실상 알고리즘이라고 부르는

어렵다. 알고리즘 시 티는 3D 기반이지만 y 축이 없는 2D 형식으로, 펭귄을 임의의 장소로 이동하는 방식의 게임이다. 알고리즘 시티 또한 10 만 이상 다운로드에 평점이 4.3 이지만, 실제 플레이 한 결과 알고리즘의 핵심 중 하나인 IF 및 operator 형식이 없었고, 캐릭터 모션과 퍼즐 게임이 너무 간단하여 몰입도가 떨어지고 성취감이 부족하였다.

상반기 장르별 월평균 모바일 게임 사용 현황



코딩을 재밌게
배울 수 있다?

BRAIMMO: AI 코딩 교육 콘텐츠

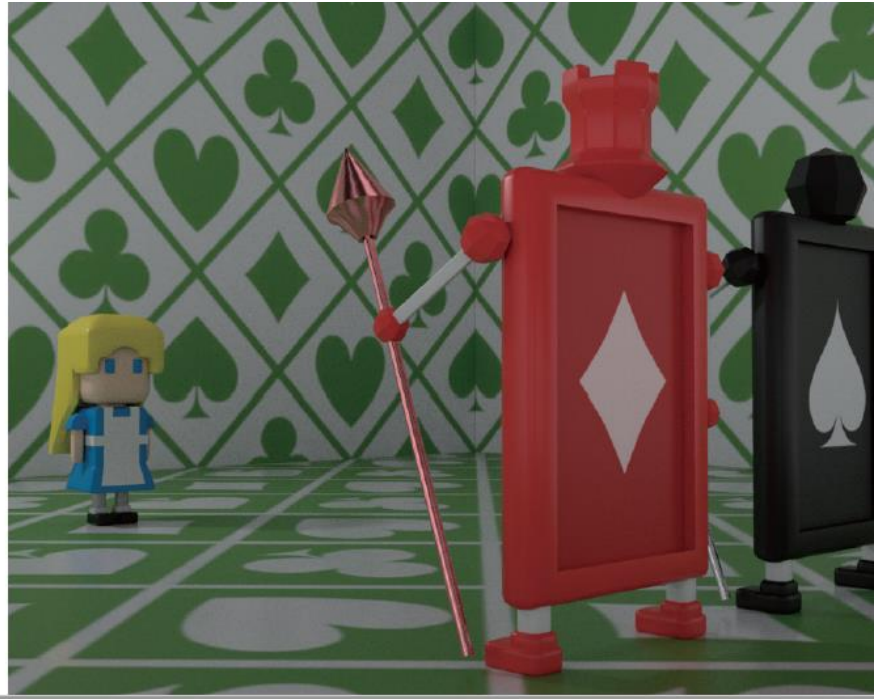
코딩에 익숙하지 않은 사람들도
재밌게 코딩 할 수 있는
AI 코딩 교육 gamification 어플리케이션

나만의 캐릭터 AI를 직접 프로그래밍하고
PVP, PVE 등의 대전을 통해
캐릭터를 성장시키는 게임

CONCEPT

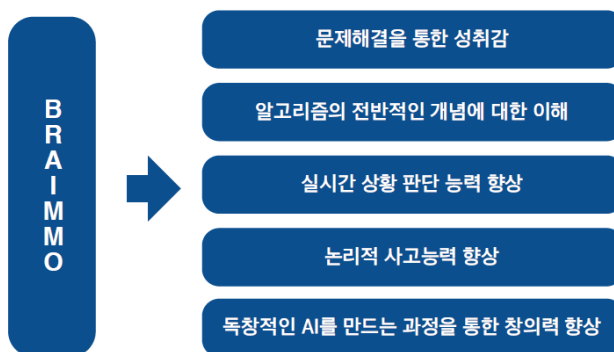
1. 전 연령층에 익숙한 RPG요소와
코딩교육을 접목시킨다.
2. 책의 스토리와 주인공들을 게임의 세계관과
캐릭터로 삼는다.

→ 코딩의 진입장벽을 낮추고 흥미를 높인다



나만의 캐릭터 AI를 직접 프로그래밍 하는 콘텐츠, BRAIMMO는 코딩의 진입장벽을 낮추기 위해 모두가 익숙한 장르인 롤플레이팅과 마인드맵 형식의 알고리즘을 접목시킨 AI 코딩 교육 어플리케이션이다.

교육적인 요소



1.2 Project Goal :

- 코딩 및 알고리즘 사고에 익숙지 않은 유저가 게임을 하며 자연스럽게 알고리즘적인 사고를 배운다.
- 브레이모를 통해 돈이 없어 코딩 사교육을 받지 못하는 학생들이 사교육을 받지 않고도 알고리즘적인 사고를 배울 수 있다.

1.3. Key Project Stakeholders

- 주 고객층은 초등학교 고학년,중학생
- 알고리즘적 사고에 익숙하지 않은 모든 사람

1.4. Project Requirements and Constraints :

1.4.1. Requirements

- 교육과 게임이 합쳐진 '기능성 게임'의 단점인 '게임 몰입도 부족'을 보완하기 위해 알고리즘 교육에 최근 모바일 게임에서 가장 영향력 있는 카테고리인 RPG 를 결합.
- 자연스러운 알고리즘 교육을 위해 프로그래밍 언어와 유저의 직접적인 교류 지양.
- IoT, AI 의 실시간 대처 능력의 핵심인 Arduino Style 알고리즘을 이용하여 RPG 캐릭터의 움직임과 대처능력 고안.

1.4.2 Constraints

제한조건	내용
제작비용 및 기간	2019-12-30 ~ 2020-12-15 1년
사회, 문화, 윤리	수익 창출을 위해선 과금이 필요한데 어떻게 이를 적정선에서 조절할 수 있을 것인가?
안전, 보건, 환경	게임 페이지, 캐릭터 구현
산업표준	해당 없음
기타 (미학 작 요인 등)	Unity, Json Database, Firebase Realtime Database, Git

1.5 Deliverables (최종산출물) :

- MS WORD 보고서 file
- 보고서 hardcopy
- PDF 경진대회 발표 file
- PDF 경진대회 영어 발표 file
- 경진대회 발표 hardcopy
- 경진대회 발표 영어 hardcopy
- 데모 동영상
- 홍보 동영상
- APK 어플리케이션
- MS WORD 심사위원 feedback file
- Design assets

2. PROJECT BACKGROUND

2.1 Literature Review

논문명: 기능성게임 성과분석 및 활성화 방안 연구 (2018.3)

KOCCA17-53

- 발행처: KOCCA(한국콘텐츠진흥원)
- 핵심 아이디어 분석

기능성게임의 요소에 대해 논의하려면 디지털 게임에 대한 파악이 필요하다. 디지털 게임의 요소 가운데 가장 중요한 것이 게임성이다.

여러가지 교육의 목적성을 탑재한 기능성 게임들이 출시되고 있지만 실제 기능성 게임을 이용하는 이용자수는 상당히 소수이다. 기능성 게임들은 너무 많은 교육 요소를 넣는 과정에서 게임의 중요한 요소인 몰입성이 사라지고 있다.

- 활용할 아이디어

핵심 아이디어 분석을 통해 기능성 게임으로서 성공하기 위해서는 몰입도가 키워드라는 것을 알았다. 몰입도를 높일 방법을 찾기 위해 기존 게임 시장에 대하여 조사 한 결과, 19 년 상반기 모바일 게임월 평균 이용시간은 롤플레이게임이 51.9 시간으로 제일 높았고 이는 롤플레이이라는 장르가 유저가 몰입하기 좋은 게임 장르임을 입증한 다고 볼 수 있다. 따라서 현재 모바일 게임 시장에 서 롤플레이게임이 강세임을 알 수 있었고, 롤플레이 장르를 기반으로 게임을 만들기로 정하였다.

- 보완 발전

게임 내 캐릭터 AI 를 유저가 직접 생성 및 수정할 수 있는 설계 시스템을 게임에 적용하여 다른 게임에서 는 찾아볼 수 없는 알고리즘의 교육적 가치도 포함하도록 하여 기능성 게임의 본래 가치를 잃지 않도록 하였다. 이를 통해 유저들은 몰입도 있게 게임을 즐기면서 무의식적으로 알고리즘이 돌아가는 원리를 체득할 수 있게 하였다.

2.2 <표. 최신 정보 및 선행연구 분석>

기존 정보 조사 내용	문헌명 및 저자	문헌의 핵심 내용	본 과제와의 공통점	본 과제와의 차이점
	<p>한국등록특허 10-2002-0081733</p> <p>안재필, 이윤준, 홍동철</p>	<p>상기 게임캐릭터 또는 캐릭터화된 객체는 절대적 시공간 속에서 사용자의 의지와 관계없이 게임의 알고리즘에 대해 반응하거나 객체의 의지대로 게임이 진행되며, 사멸로 인해 게임이 종료되면 게임캐릭터 또는 캐릭터화된 객체의 유일한 특성에 의해 게임을 진행하던 캐릭터로의 반환이 불가능하여 새로운 게임캐릭터 또는 캐릭터화된 객체를 통해 새로운 게임을 진행하는 것을 특징으로 하는 절대적 시공간의 게임캐릭터를 통한 게임방법이다.</p>	<p>캐릭터화된 객체가 유저의 개입 없이 특정 알고리즘을 통해 절대적 시공간 안에서 구현이 가능하다는 점에서 본 발명과 일부 유사함.</p>	<p>게임의 시작과 끝에 대한 조건이 RPG 월드 상의 위치, 시간 알고리즘에 기반한다는 점에 있어서 본 발명과 다름.</p>
	<p>한국특허등록 10-2013-0066758</p> <p>(2013.06.11)</p> <p>(주)위메이드 엔터테인먼트</p>	<p>본 발명의 일 측면에 따른 온라인 게임에 등장하는 대상체를 자동으로 타겟팅하는 방법은, 사용자가 제어할 수 있는 캐릭터로부터의 전방 거리</p>	<p>게임에 등장하는 대상체를 타겟팅을 할 때, 사용자의 캐릭터로부터 대상체와의 거리와 게임에 등장하는 대상체의 종류를 기준</p>	<p>본 발명은 게임 내에서 타겟팅을 하는 대상의 조건은 게임을 시작하기 전에 이용자가 직접 설정할 수 있다.</p>

		<p>및 상기 캐릭터의 위치를 중심으로 한 소정의 좌우 각도를 포함하는 타겟팅(targeting) 범위를 설정하는 단계.</p> <p>상기 설정된 타겟팅 범위 내에 포함된 적어도 하나 이상의 대상체(target)들을 선택하는 단계 및 상기 선택된 대상체들과 인접한 위치에 상기 대상체의 형상과 구별되는 모양을 화면에 표시하는 단계를 포함하고, 상기 대상체는 온라인 게임에 등장하는 객체(object)들 중 상기 캐릭터가 공격할 수 있는 객체를 의미한다.</p>	<p>으로 타겟팅을 한다. 캐릭터의 위치를 중심으로 한 좌우 각도는 항상 360도이다.</p>	<p>게임을 실행하게 되면 이용자는 어떠한 조작도 하지 않고 게임이 실행되기 전에 설정된 조건들에 의해 게임 내 사용자의 캐릭터가 움직이고 타겟을 정하게 된다.</p>
	<p>한국특허등록 10-2013-0039515</p> <p>(2013.04.10)</p> <p>(주)위메이드 엔터테인먼트</p>	<p>모바일 게임에서의 스킬 동작 수행방법은 사용자의 캐릭터가 피격 대상 캐릭터에게 수행할 특정 스킬 동작을 설정하고 상기 스킬 동작에 대응하는 피격조건을 설정하는 단계, 상기 스킬 동작의 피격조건이 충족되었는지를 판단하는 단계, 상기 피격조건이 충족되면</p>	<p>본 발명은 모바일 게임에서 사용자가 본인의 캐릭터 AI를 직접 만들어서 게임이 실행되고 나면 사용자가 개입을 하지 않아도 직접 만든 AI에 따라 캐릭터가 움직이도록 하는 것을 목적으로 한다.</p>	<p>사용자의 캐릭터가 수행할 특정 액션을 선택하고 해당 액션에 대응하는 조건을 설정하는 단계, 조건이 충족되었는지를 판단하는 단계, 조건이 충족되면 자동으로 액션을 하는 단계를 포함한다.</p>

		상기 스킬 동작에 대응되는 아이콘을 표시하는 단계 및 상기 스킬 동작의 수행을 지시하는 신호를 수신하면, 상기 스킬 동작을 수행하고, 추가적인 보상을 제공하도록 제어하는 단계를 포함한다.		
활용 결과	<p>인용문헌 1의 장점을 가져와 게임이 시작된 후 유저의 캐릭터화된 객체(들)와 상대의 캐릭터화된 개체(들)만이 존재하는 가상 공간을 생성 하도록 하되 단점을 보완하여 각 진영의 객체들이 모두 죽음을 맞이할 경우에 게임 공간을 종료한다는 특징점을 찾아냄.</p> <p>본 발명은 인용문헌 2의 장점을 가져와 다수의 대상체가 있을 경우 타겟팅 대상을 자동으로 지정하게 하여 유저의 개입이 없이도 정해진 알고리즘에 따라 캐릭터화된 객체들이 움직이도록 하되 인용문헌 2의 단점을 보완하여 게임이 실행되기 전에 이용자가 타겟팅을 하는 조건을 직접 지정할 수 있도록 한다. 이때 거리가 가까운 순서 혹은 그 반대로 거리가 먼 순서로 타겟팅 하도록 선택할 수 있으며 대상 객체의 종류에 따라 타겟팅 우선순위를 정할 수 있도록 한다.</p> <p>따라서 본 발명은 인용문헌 3의 장점을 가져와 게임 내 캐릭터의 모든 행동(움직이기, 공격, 아이템 사용)에 위 과정이 적용하되 단점을 보완하여 피격조건이 충족되었을 경우 자동으로 행동을 하게 함으로 게임 중 이용자가 개입을 전혀 하지 않아도 되게 하였다.</p>			

2.2 Review of Industrial Standards and Regulation

적용해당 없음.

3. Proposed Design and Solution

3.1. Overall System Architecture

Specifications

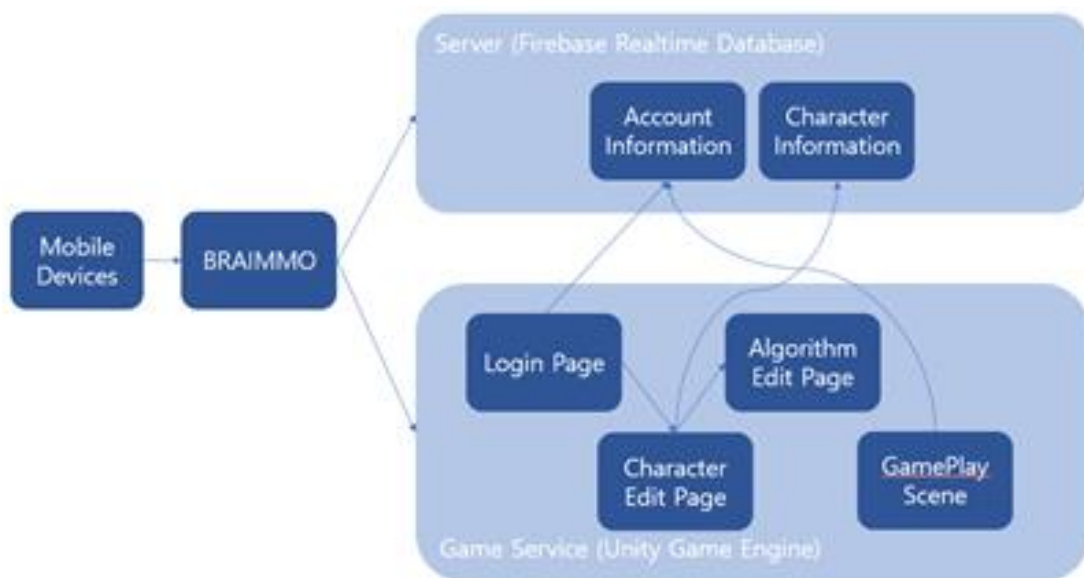
도래하는 4 차 산업혁명에 맞춰 재밌게 코딩 교육을 하는 기능성 게임이 시장에 많이 나오고 있다. 하지만, 현재 국내 기능성 게임은 알고리즘이 아닌 코딩 문법 교육에 치중되어 있어 학생들이 기능성 게임을 하며 재미를 느끼기 보다, 공부를 하고 있다는 느낌을 강하게 받는다.

우리는 이런 한계를 보완하고자 롤플레이팅 장르와 알고리즘 교육을 합쳤다. 유저는 자신이 설계한 알고리즘대로 자신의 캐릭터가 적을 무찌르고, 캐릭터가 성장하는 것을 보며 지적 쾌감과 성취감을 느끼게 된다.

- Goals
 - 게임에 몰입하며 자연스럽게 알고리즘 사고를 배운다
 - 코딩을 처음 접하는 사람이 코딩에 대한 거부감 없이 접근한다
- Product Requirements
 - 이용자들이 코딩을 하고 있다고 느끼지 않도록 한다
 - 게임에 맞는 UI/UX 디자인과 캐릭터 디자인을 통해 유저가 게임에 집중할 수 있도록 한다
 - 유저가 짠 알고리즘이 게임의 승부를 좌우하게 하기 위해 과금은 스킨, 이펙트 같은 승부에 영향을 미치지 않는 것을 지향한다
- Plan
 - 게임을 진행하며 일정 스테이지를 지날 경우, 어느 정도의 논리체계를 배웠는지 명시해 주고, 간단한 퀴즈 형식의 퀘스트를 통해 해당 내용 숙지를 확인한다
 - 교육부와 협업하여 방과 후 교실, 창의적 체험 학습 시간과 같은 시간에 코딩 교육을 보조하는 용도로 사용하도록 한다
- Out of Scope
 - 이 제품은 코딩 교육이 아니라 알고리즘 교육용이다
 - 이 제품은 학생에 제한하지 않고 전 연령층이 즐길 수 있도록 한다
- Open Questions
 - 알고리즘적 사고의 성취 지표를 게임 진행을 하며 어떻게 보여줄 것인가?
 - 사업화 방향은 어떻게 할 것인가?
 - 처음 접하는 이용자들에게 튜토리얼 진행은 어떻게 할 것인가?

- Monitoring
 - 주변 교회 혹은 아동센터와 같이 학생들이 많은 곳으로 가서 B2C 형태로 이용자들의 피드백을 받고 적극 반영한다
 - Git 을 통해 전체적인 코드를 관리한다
- Team
 - 개발 팀
 - 디자이너 팀

System Design



본 프로젝트는 모바일 디바이스를 통해 BRAIMMO 모바일 게임을 실행할 수 있다. 서버와 게임 서비스라는 두 가지의 틀로 나누어 설명할 수 있다.

서버 측에서는 Account Information, Character Information 블록이 존재한다.

- Account Information
 - 계정의 아이디
 - 계정의 비밀번호
 - 계정의 생성일
 - 계정의 레벨
 - 계정이 가진 캐릭터 아이디

- Character Information
 - 캐릭터 아이디
 - 캐릭터 아이템
 - 캐릭터 스탯
 - 캐릭터 알고리즘 노드 및 노드 위치

게임 서비스 측에서는 페이지 별로 구분할 수 있다. Login Page, Character Edit Page, Algorithm Edit page, Game Play Scene 블록이 존재한다.

- Login Page
 - 게임이 실행된 후 첫 페이지의 로그인 페이지
- Character Edit Page
 - 로그인 후 Home 화면에서 캐릭터 페이지로 이동 가능
 - 해당 계정이 가진 캐릭터의 정보 및 아이템을 표시
- Algorithm Edit Page
 - 캐릭터 페이지에서 해당 캐릭터의 (코드 수정)을 통해 이동 가능
 - 캐릭터의 알고리즘 수정 가능

상호 작용

- Mobile Device 를 통해 BRAIMMO 모바일 게임을 실행
- 실행 후 첫 페이지를 로그인 페이지로 지정
- 로그인 페이지에서 계정 생성 혹은 기존 계정 정보 입력을 통해 서버 측의 Account Information 정보 추출
 - 계정 생성일 경우 새로운 계정 번호를 할당 후 초기화 정보를 입력하여 Game Service 의 홈 화면으로 이동
 - 기존 계정 정보 입력을 통한 로그인일 경우 서버 측에 존재하는 기존 아이디 및 비밀번호에 상응하는 계정을 탐색하여 해당 계정과 캐릭터 정보를 추출 후 Game Service 의 홈 화면으로 이동
- 홈 화면에서 게임 시작 및 캐릭터 수정 페이지로 이동
- 캐릭터 수정 페이지로 이동 시 해당 계정의 캐릭터 정보를 추출.



- 대표 캐릭터의 스텟, 아이템 정보를 표기
 - 대표 캐릭터 외 다른 캐릭터는 이미지 하단 부분의 스크롤 바를 통해 캐릭터 이미지만 표시. 캐릭터 이미지를 클릭하면 해당 캐릭터의 전체 정보를 추출하여 표시
 - 코드 수정을 통해 해당 캐릭터의 알고리즘 수정
-
- 알고리즘 수정 페이지는 캐릭터 수정 페이지에서 해당 캐릭터 표시 후 코드 수정을 통해서 이동 가능. 페이지는 서버의 Character Information 을 통해 해당 캐릭터에게 할당된 알고리즘 노드와 그 위치, 연결 순서 등의 정보를 추출하고 Account Information 을 통해 캐릭터에게 할당되지 않은 Free 노드를 추출.
-
- 게임 플레이 Scene 은 홈 화면에서 Story → Stage 를 통해 이동 가능. 대전에 참여한 계정의 캐릭터 정보와 적 Non Player Character 에 대한 정보를 Game Service 에 내장되어있는 JSON 파일을 통해 추출.

3.2. Detail Design

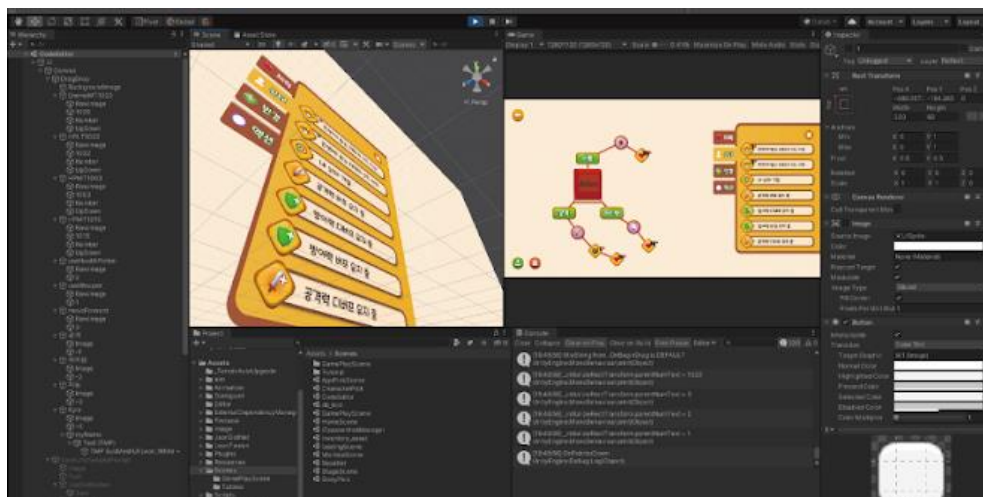
3.2.1 전반적 설계

서버는 Firebase Realtime Database 로 구현하였다.

계정에 필요한 모든 정보를 한 JSON 으로 통일. 각 계정에 필요한 정보는 오직 string, int 등 variable 로 구현 후 이미지나 캐릭터 파일 등은 게임에서 Resource 폴더를 통해 상황에 맞게 Import 한다.

계정에 필요한 모든 정보를 한 JSON 으로 통일. 각 계정에 필요한 정보는 오직 string, int 등 variable 로 구현 후 이미지나 캐릭터 파일 등은 게임에서 Resource 폴더를 통해 상황에 맞게 Import 한다.

Game Service 는 Unity 2019.4.9f1 로 구현하였다. (Unity 의 개발 언어는 C#)



3.2.2 세부 설계

Game Service 의 페이지들은 Unity 의 Scene 을 나타낸다. 각 Scene 을 연결함으로써 게임 플로우 구현이 가능하다. Scene 의 연결 방법은 아래와 같다.

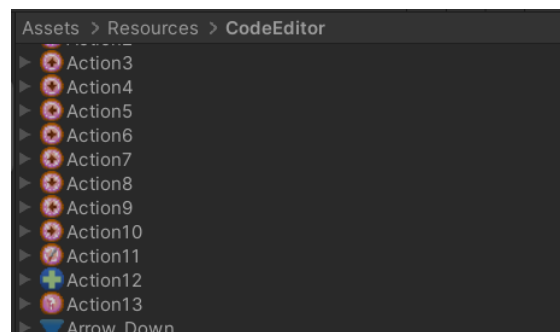
```
UnityEngine.SceneManagement.SceneManager.LoadScene("HomeScene");
```

각 Scene 과 서버와의 실시간 연동은 C#의 Start 와 Update 으로 나타낼 수 있다. 각 Scene 의 Manager Script 를 만들어 Firebase 와 연동하는 방법은 아래와 같다.

```
FirebaseApp.CheckAndFixDependenciesAsync().ContinueWithOnMainThread(task =>
{
    if (task.Exception != null)
    {
        Debug.Log($"Failed to initialize Firebase with {task.Exception}");
        return;
    }
    else{
        OnFirebaseInitialized.Invoke();
        auth = Firebase.Auth.FirebaseAuth.DefaultInstance;
        firebaseApp = FirebaseDatabase.DefaultInstance.App;
        firebaseApp.SetEditorDatabaseUrl("https://braimmo.firebaseio.com/");
        databaseReference = FirebaseDatabase.DefaultInstance.RootReference;

        accountID = GameObject.Find("AccountID_DontDestroy").GetComponent<AccountID>();
        createBtn.GetComponent<Button>().interactable = false;
        password_forCreate.onValueChanged.AddListener(delegate { ValueChangeCheck_reg();
    }
});
```

각 페이지의 디자인은 C4D, Illustrator 를 이용하여 제작 후 Export 하여 Unity 에 Import. Import 된 파일들은 각 Scene 에서 필요할 때 로드할 수 있도록 Resource 파일에 아래와 같이 저장.



알고리즘 설계

RPG 게임에서 플레이 진행 시 캐릭터에게 필요한 행동 요소를 크게 이동하기, 공격하기, 아이템 사용이라 판단하여 캐릭터 AI 생성 시 액션의 카테고리를 레이블: 이동, 공격, 아이템으로 설정한다. 그 후, 각 카테고리(이하 레이블)에 해당하는 액션과 아이템을 정의하고 각 액션이 게임 내에서 실행될 수 있는 모든 조건을 정의한다.

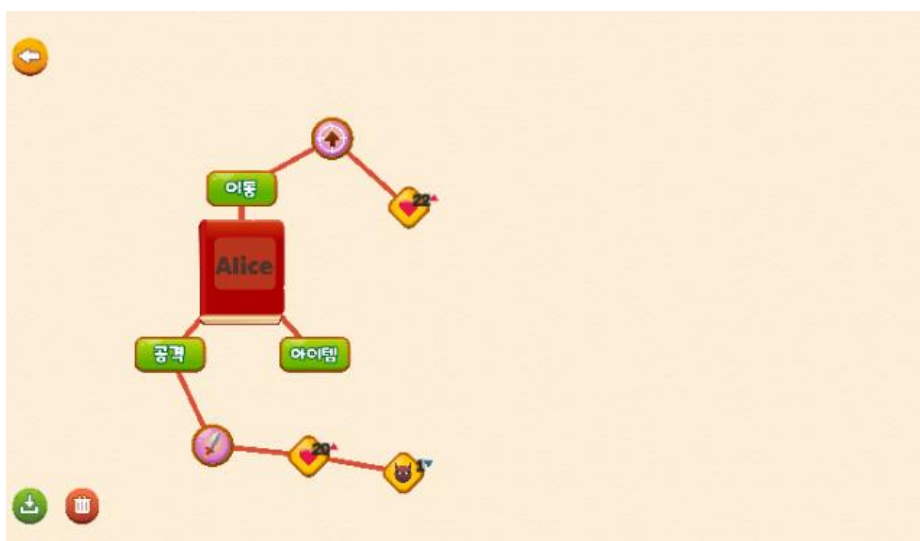
본 기술(모바일 환경 기반 드래그 앤 드롭의 클러스터 맵핑을 통한 AI 생성)은 게임 캐릭터 AI 를 위와 같이 레이블 - 액션 - 조건의 클러스터 맵핑 방식으로 손쉽게 만들 수 있어 기존의 모바일 환경에서 유저 스스로 AI 생성이 불가능했다는 문제를 해결할 수 있다.

기존의 모바일 게임은 제작자가 만든 고정된 AI 를 유저들이 사용하는 방식으로, AI 를 유저가 수정할 수 없었다는 한계를 가지고 있었으나, 위 기술로 사용자가 직접 적합한 AI 를 생성 및 수정하여 게임을 플레이 할 수 있다는 측면에서 혁신적인 차이를 보인다.

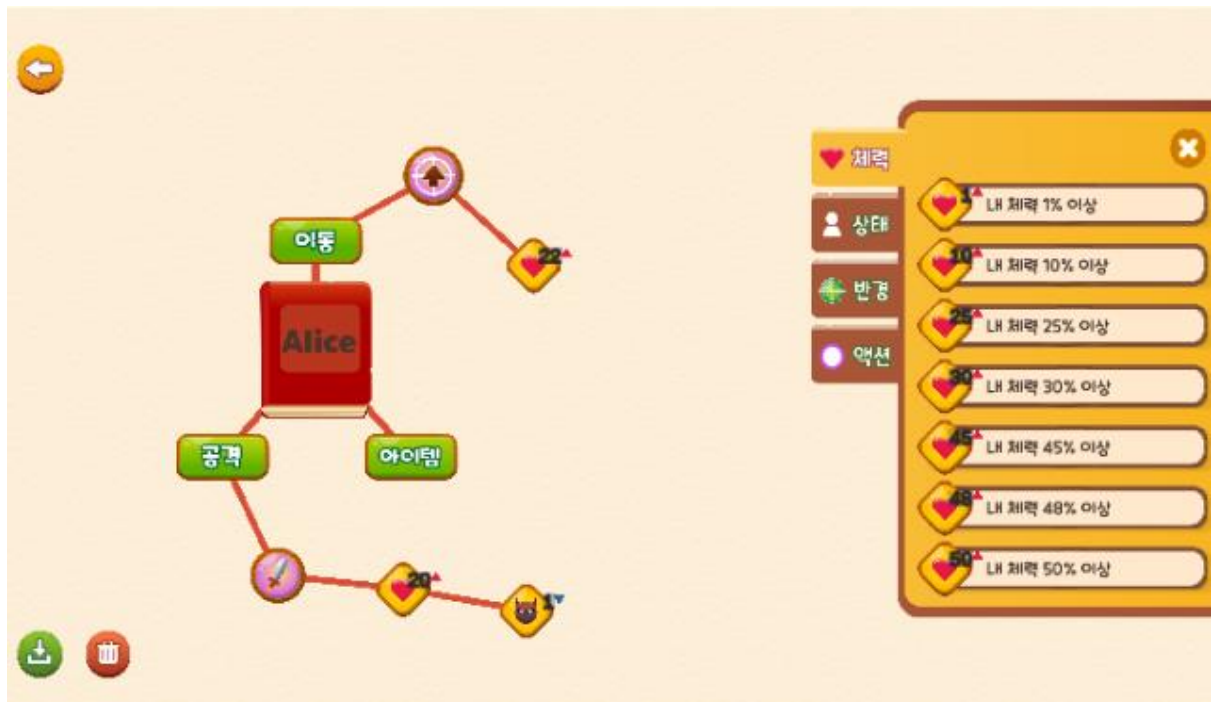
캐릭터의 AI 를 생성하는 핵심 기술 '드래그 앤 드롭을 사용한 클러스터 매핑 기반 AI 생성'을 기반하여 만든 알고리즘대로 캐릭터가 실제 플레이에서 움직이는 것을 확인할 수 있다.

핵심 기술을 활용한 캐릭터 AI 수정 페이지

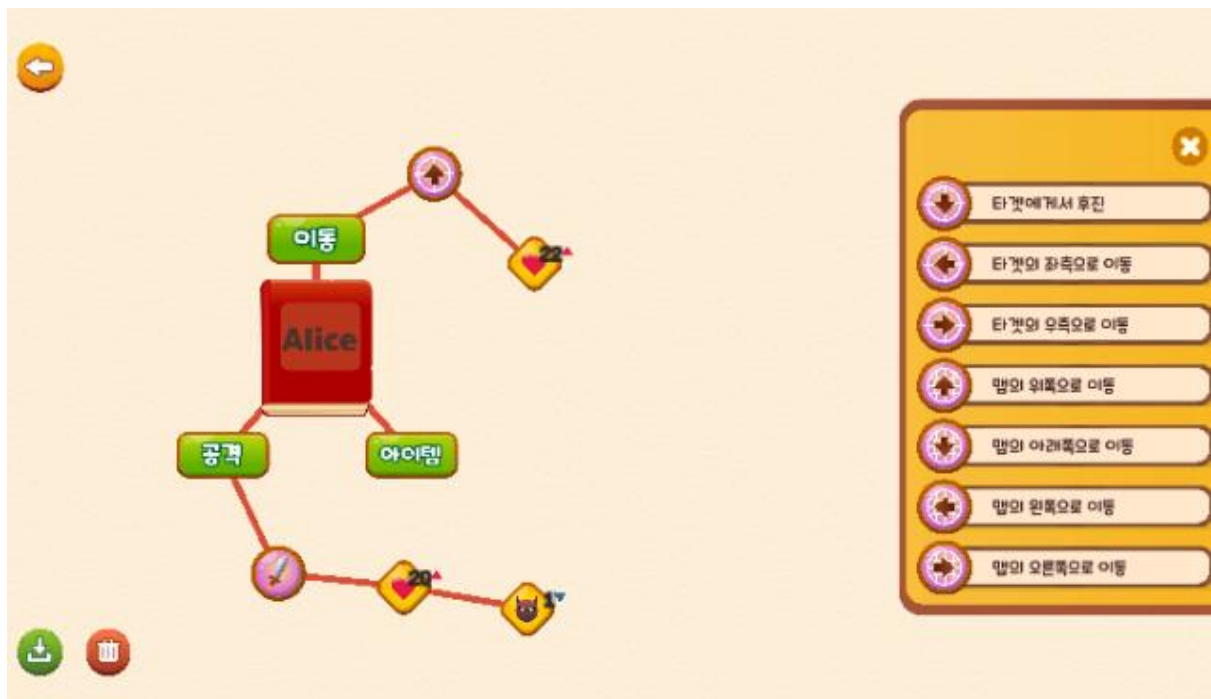
가운데 책 아이콘은 사용자가 AI 를 생성하고자하는 캐릭터를 나타낸다. 각 캐릭터는 이동, 공격, 아이템 세 가지 레이블을 가지고 있고 각 레이블에 실행하고자 하는 액션을 이을 수 있다. 액션에는 액션이 실행되는 조건을 이을 수 있다. 이 때 액션은 오직 레이블에만, 조건은 액션에만 연결할 수 있다. 유니티는 매 초 30 프레임씩 조건을 체크하면서 조건이 만족될 때 액션을 수행하게 된다.



아직 할당되지 않은 노드 리스트는 편의를 위해 슬라이딩을 통해 표시/숨기기 가능.



액션과 조건의 노드들은 구분하여, 레이블을 클릭할 경우 액션 노드를 표시, 액션이나 조건을 클릭할 경우 조건 노드를 표시한다.



캐릭터 AI 생성 예시) 적에게 다가가 공격하는 알고리즘



<이동에 대한 알고리즘을 적용한 예시>



<공격에 대한 알고리즘을 적용한 예시>

'앞으로 이동하기' 라는 액션이 적용되는 조건은 '체력이 50% 이상' 조건과 '공격 반경 안의 적이 1명 이하' 조건이 'AND'로 연결되어 두 조건이 모두 만족될 경우이다.

'무기 사용' 이라는 액션이 적용되는 조건은 '공격 반경 내에 적이 1명 이상일 때' 조건과 '방어력 버프를 받았을 경우' 조건이 'OR'로 연결되어 두 조건 중 하나 이상 만족될 경우이다.

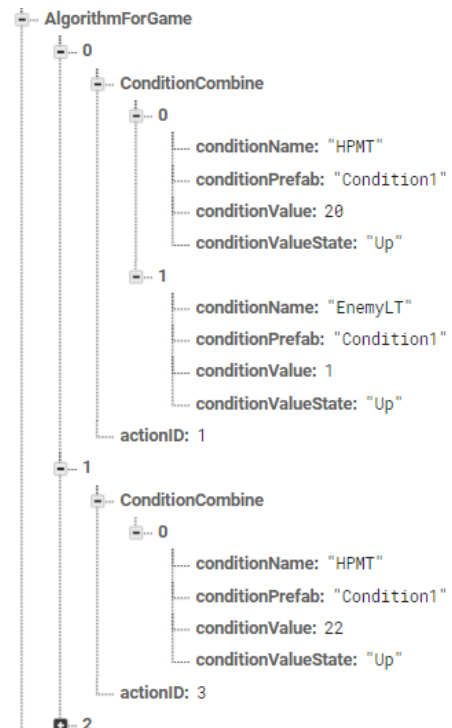
각 레이블에 해당된 조건들을 매 프레임마다 반복적으로 확인하며 캐릭터의 알고리즘이 적용된다.

적용된 알고리즘은 Realtime Database 에 캐릭터에게 적용될 알고리즘으로 변형되어 저장된다. 방법은 아래와 같다.

- 총 조건 리스트를 다 확인하면서 쓰인 조건이 무엇인지 확인한다.
 - 쓰인 조건의 parent node 가 존재할 경우 쓰인 조건이라는 것을 확인할 수 있다.
 - 조건과 액션 모두 위의 방법으로 쓰였는지 확인할 수 있다.

쓰인 node 가 조건인지 액션인지를 구분하여 sort 를 실행한다. 조건의 경우, parent 의 노드가 액션인 경우가 있고 조건이 될 수도 있어서 구분 지어준다.

- 만약 조건의 parent node 가 조건인 경우, parent node 의 parent 를 다시 탐색하여, parent 가 액션으로 끝날 때 까지 계속 탐색한다. (여기서 parent node 가 조건이라는 뜻은 'and' 로 연결된 조건임을 알 수 있다.)



<[ConvertToJson.cs] 스크립트 중>

```
public void SortUsed()
{
    //initialize
    conditionParentCondition = new List<Condition_CodeModifier>();
    conditionParentAction = new List<Condition_CodeModifier>();
    conditionUsed = new List<Condition_CodeModifier>();
    actionUsed = new List<Action_CodeModifier>();
    conditionList_forGame = new List<Condition_forGame>();
    conditionList_Total = new List<List<Condition_forGame>>();
    finalizeString = new List<Finalizer>();

    for(int i = 0; i < conditionManager.conditionList.Count; i++) // 총 conditionList 를 다 돌면서 체크
    {
        if(conditionManager.conditionList[i].conditionParent != 0) // 만약 parent 가 달려 있을 경우
        {
            conditionUsed.Add(conditionManager.conditionList[i]); //달려있는 애 이름인 conditionUsed 에 저장
        }
    }
    for(int i = 0; i < actionManager.actionList.Count; i++) //condition 도 했으니까 action 도 해주자
    {
        if(actionManager.actionList[i].actionUsed != 0)
        {
            actionUsed.Add(actionManager.actionList[i]);
        }
    }
    SortConditionParent();
}
public void SortConditionParent()
{
    for(int i = 0; i < conditionUsed.Count; i++) //conditionUsed 도 한번에 짝 돌리면서
    {
        if(conditionUsed[i].conditionParent > 1000) //만약 condition 의 parent 가 condition 인 경우
        {
            conditionParentCondition.Add(conditionUsed[i]); //conditionParentCondition 에다가 저장
        }
        else
        {
            conditionParentAction.Add(conditionUsed[i]);
            //아닐 경우, 즉 condition 의 parent 가 action 인 경우 conditionParentAction 에 저장
        }
    }

    for(int i = 0; i < conditionParentAction.Count; i++) //condiitonParentAction 이 가장 먼저 시작
    {
        //print("New Branch Starts Here");
        conditionList_forGame = new List<Condition_forGame>();
        conditionList_Total = new List<List<Condition_forGame>>();
        conditionList_forGame.Add(new Condition_forGame(conditionParentAction[i].conditionName, conditionParentAction[i].conditionValue, conditionParentAction[i].conditionPrefab, conditionParentAction[i].conditionValueState));
        int checkEqualFirst = conditionParentAction[i].conditionUniqueID;
        int tempCount = conditionList_forGame.Count; //1
        do
        {
            tempCount = conditionList_forGame.Count; //1
            for(int j = 0; j < conditionParentCondition.Count; j++)
            {
                if(checkEqualFirst == conditionParentCondition[j].conditionParent)
                {
                    conditionList_forGame.Add(new Condition_forGame(conditionParentCondition[j].conditionName, conditionParentCondition[j].conditionValue, conditionParentAction[i].conditionPrefab, conditionParentAction[i].conditionValueState));
                    checkEqualFirst = conditionParentCondition[j].conditionUniqueID;
                    print("checkequalfirst = " + checkEqualFirst);
                    conditionParentCondition.RemoveAt(j);
                }
            }
            //print("conditionList_forGame Count is " + conditionList_forGame.Count);
        } while (tempCount != conditionList_forGame.Count);
        //conditionList_Total.Add(conditionList_forGame);
        Finalizer finalizer = new Finalizer
        {
            actionID = conditionParentAction[i].conditionParent,
            ConditionCombine = conditionList_forGame
        };
        finalizeString.Add(finalizer);
    }
    print("conditionList_Total Count is " + conditionList_Total.Count);
    WriteJson();
}
```

4. Implementation and Evaluation

4.1 Implementation(구현)

4.1.1 구현에 사용된 개발 도구 및 구현 환경

- Unity: 게임을 개발하기 위한 핵심 엔진. 이를 활용하여 게임 apk 를 만듦.
- Firebase: 서버 연결
- Figma: 디자이너와 개발자 협업 tool
- Json DB: 계정의 모든 정보를 Json Format 을 이용하여 데이터베이스 관리.
- git: source code 관리.

4.1.2 구현 결과

목표: 이용자가 본인의 캐릭터의 AI 를 직접 생성하고 이를 전장에 내보낸다. 또한 본인이 짠 AI 를 통한 전투로 얻은 보상을 통해 캐릭터의 성장을 기대한다.

애초 목표는 1 vs 다수 형태의 게임까지 개발하는 것이었으나 구현하지 못하였다.

Home Screen - 유저가 처음 들어왔을 때 page



Inventory Page



Cluster Mapping Page



레이블

이동 / 공격 / 아이템으로
카테고리화



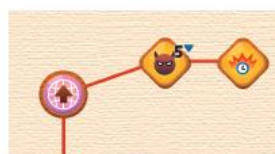
액션

각 레이블에 해당하는 액션



조건

액션이 게임 내에서
실행 될 수 있는 모든 조건

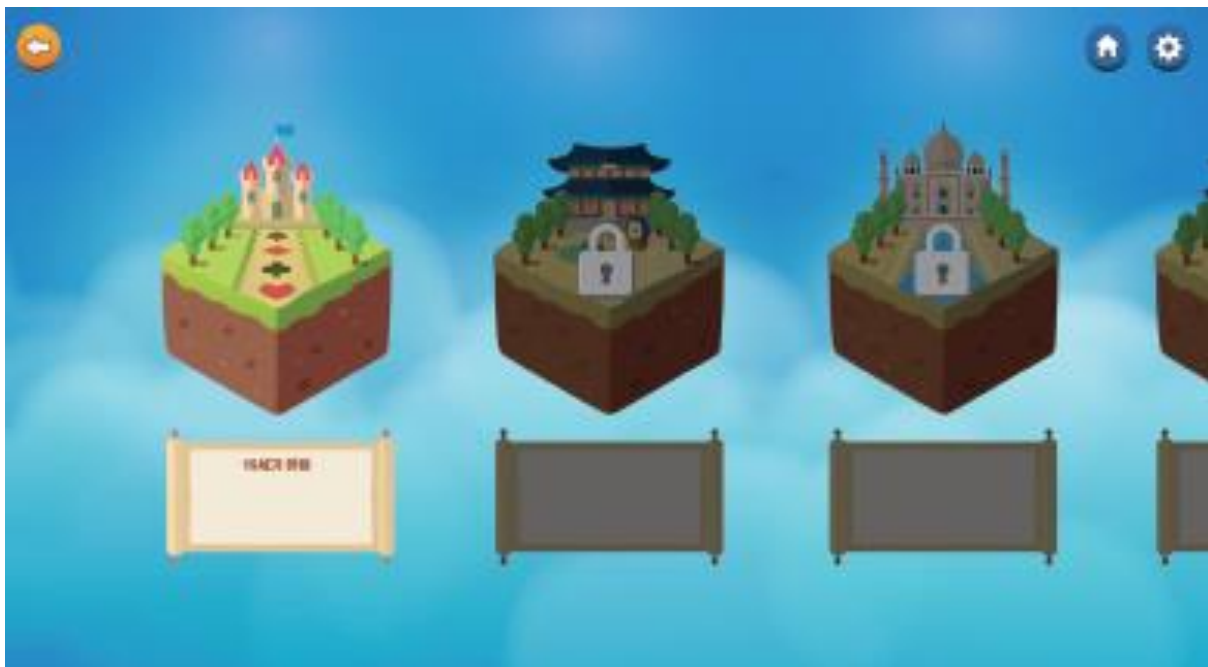


<AND 조건>

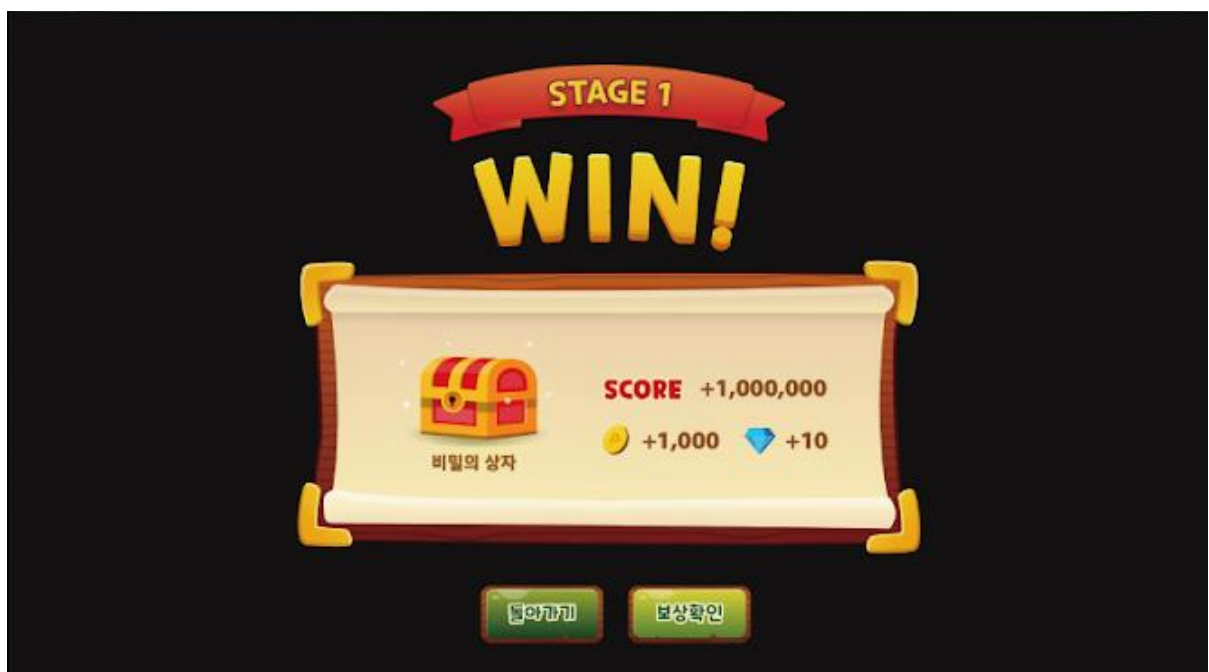
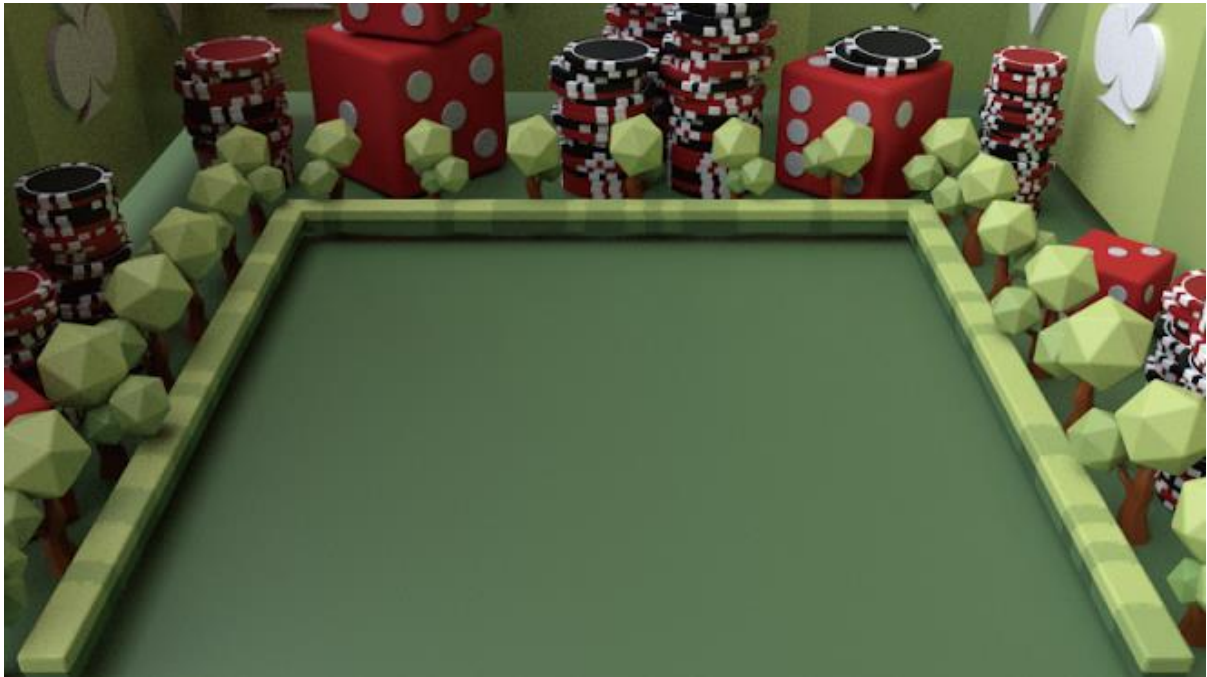


<OR 조건>

Stage, Age Pick Page

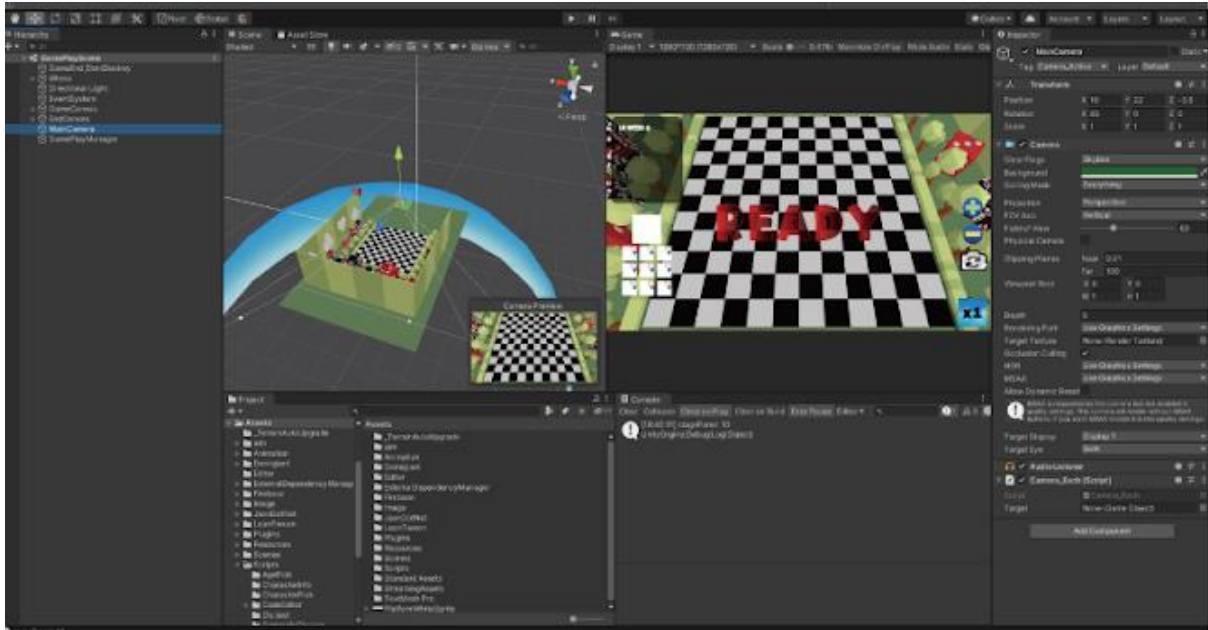


Battle, Reward page



Design





4.2 Experimental Setup

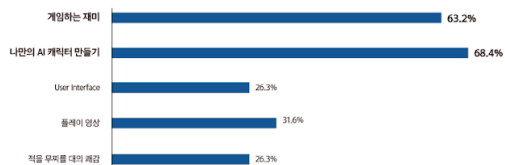
- Google Sprint Camp 를 통한 user feedback 을 통해 진행
- 로그인 화면에서 시작.
- 별다른 지시 없이 어플리케이션을 5 분동안 작동해보도록 하고 feedback 진행.
- 20 여명의 사람들의 feedback

4.3 Evaluation (설계평가)

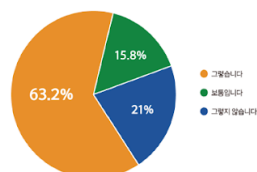
- 측정목표: 코딩 및 알고리즘 사고에 익숙지 않은 유저가 게임을 하며 자연스럽게 알고리즘적인 사고를 배울 수 있는가?
- 평가: 20 여명의 대상으로 어플리케이션을 스스로 작동하도록 해본 결과 80% 이상의 사람들이 위 게임이 코딩 및 알고리즘을 이해하는데 도움이 되는가에 대해 긍정적인 평가를 내렸다.

User Test 설문조사 결과

게임의 매력포인트를 모두 선택하세요.



이 게임이 코딩을 이해하는데 도움을 주나요?



2020.10 초등학교 5학년 대상으로
유저테스트 진행

게임의 디자인에 특별히 호감을 보였으며,
세세한 튜토리얼이나 설명 없이도 전반적인
게임 사용 방법에 대해 터득을 하였다.

5. CONCLUSION

프로젝트의 목적

코딩 및 알고리즘 사고에 익숙하지 않은 유저가 게임을 하며 자연스럽게 알고리즘적인 사고를 배우고 돈이 없어 코딩 사교육을 받지 못하는 학생들이 사교육을 받지 않고도 알고리즘적인 사고를 배울 수 있다.

평가

20 여명의 대상으로 어플리케이션을 스스로 작동하도록 해본 결과 80%이상의 사람들이 위 게임이 코딩 및 알고리즘을 이해하는데 도움이 되는가에 대해 긍정적인 평가를 내렸다.

의의

4 차산업혁명 시대를 맞아 코딩교육에 대한 관심이 폭발적으로 증가하고 있는데 대부분의 사람이 코딩을 국영수과와 같이 공부해야 할 재미없는 것으로 인식하고 있다. BRAIMMO 는 알고리즘과 RPG 게임을 결합하여 알고리즘에 익숙하지 않은 사람이 게임을 하며 자연스럽게 알고리즘을 배우게 한 것에서 의의가 있다.

TODO:

- 교육적 성취 평가 지표: 몇 단계의 stage 를 갓을 때 몇 학년 혹은 책의 어느 단원 수준의 알고리즘적 사고를 배웠는지에 대한 지표를 나타내 준다.
- 1vs1 과 1vs 다수 시스템을 도입해 유저에게 더욱 더 다양한 콘텐츠 제공.
- 주 타겟 층이 될 초등학생 및 중학생을 대상으로 한 feedback
- 국내 특허 출원
- 영어 버전 업데이트
- 해외 특허 출원

6. REFERENCES

6.1 기술표준문서, 특허 및 참고논문

- (주)위메이드엔터테인먼트 (2013), 10-2013-0066757, 한국등록특허
- 안재필, 이윤주, 홍동철 (2002), 10-2002-0081733, 한국등록특허
- (주)위메이드엔터테인먼트(2013), 10-2013-0039515, 한국특허등록
- (주)넥슨코리아(2008), 10-2008-0042943, 한국특허등록

참조논문

- 논문명: 기능성게임 성과분석 및 활성화 방안 연구 (2018.3)
KOCCA17-53
발행처: KOCCA(한국콘텐츠진흥원)

6.2 도서

6.3 웹사이트 및 기타

웹사이트

왜 링피트는 재밌지

- <https://news.appstory.co.kr/report13509>

노마드

- <https://noahnomad-ads.com/home>

기술과 교육의 만남

- https://happyedu.moe.go.kr/happy/bbs/selectHappyArticle.do?bbsId=BBSMSTR_000000000281&nttId=8831

엔트리 나무위키

- [https://namu.wiki/w/%EC%97%94%ED%8A%B8%EB%A6%AC\(%EA%B5%90%EC%9C%A1%20%ED%94%8C%EB%9E%AB%ED%8F%BC\)?from=%EC%97%94%ED%8A%B8%EB%A6%AC%28%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D%20%EC%96%B8%EC%96%B4%29#s-6](https://namu.wiki/w/%EC%97%94%ED%8A%B8%EB%A6%AC(%EA%B5%90%EC%9C%A1%20%ED%94%8C%EB%9E%AB%ED%8F%BC)?from=%EC%97%94%ED%8A%B8%EB%A6%AC%28%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D%20%EC%96%B8%EC%96%B4%29#s-6)

computer bot turing

- https://en.wikipedia.org/wiki/Computer_game_bot_Turing_Test

git

- <https://blog.pocu.academy/ko/2019/01/06/how-to-invite-user-to-git-repo.html>

데브코리아

- http://www.devkorea.co.kr/bbs/board.php?bo_table=m03_qna&wr_id=72665

UNICEF

- https://www.behance.net/gallery/95418337/UNICEF-Teens-at-Home?tracking_source=search_projects_recommended%7Ccharacter

ICE Awards

- https://www.behance.net/gallery/62663241/Ice-Awards?tracking_source=search_projects_recommended%7Ccharacter

Unity

- <https://www.slideshare.net/unity3d/refresh-what-you-know-about-assetdatabaserefresh-unite-copenhagen-2019>

Appendix :

Part A 공동작성

A.1 How to Use It

게임을 시작하면 첫 페이지는 아래의 home page이다. 처음 시작하면 먼저 내 캐릭터의 AI를 만들어야 한다. 그를 위해 오른쪽 하단의 캐릭터룸을 클릭해 캐릭터룸으로 이동한다.



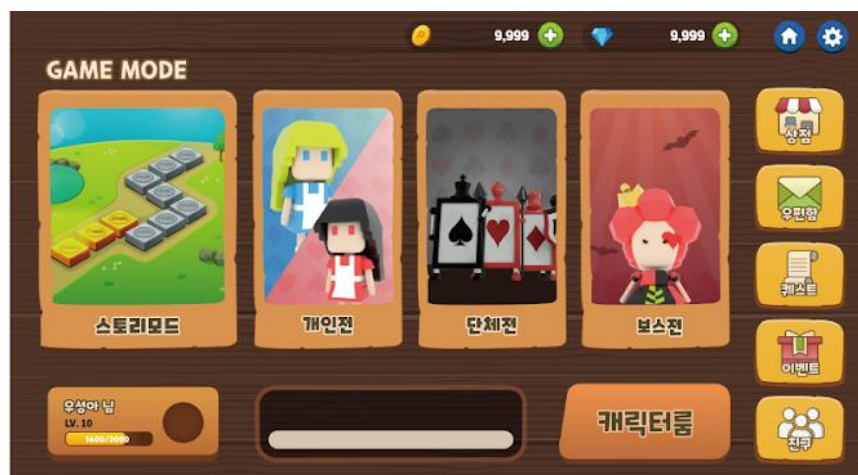
캐릭터화면에선 내 캐릭터를 선택하고 캐릭터에 맞는 장비를 착용할 수 있다. 여기서 화면 좌측 하단에 있는 코드수정을 눌러 코드수정창으로 간다.



코드수정창. 여기서 캐릭터의 AI를 만든다. 처음에는 '앞으로가기' 액션과 '체력이 20% 이상일 때' 조건을 연결하고 다시 홈 화면으로 간다.



다시 홈화면으로 와 스토리모드를 누른다.



스토리모드에서는 먼저 age를 선택한다.



그 이후 stage를 선택한다.



이후 게임을 시작한다. 보상을 받고 홈화면으로 가 보상으로 얻은 코드를 활용하여 코드를 짜며 스테이지를 클리어해간다.

A.2 상세설계 및 소스코드 블록도 설명

본문 3에서 설명.

A.3 기술적 추가설명

해당 없음.

A.4 본 프로젝트에서 생성된 지식재산권

저작권 등록 (아래 페이지)

제 C-2020-048986 호



저작권 등록증

- | | |
|-----------------|-------------------------------------|
| 1. 저작물의 제호(명칭) | BRAIMVO(브레이모) |
| 2. 저작물의 종류 | 컴퓨터프로그램저작물>응용프로그램>교육>On-Line 교육 S/W |
| 3. 저작자 성명(법인명) | 홍원표
서울특별시 서초구 잠원로 |
| 4. 생년월일(법인등록번호) | 1993년05월16일 |
| 5. 창작연월일 | 2020년12월03일 |
| 6. 공표연월일 | - |
| 7. 등록연월일 | 2020년12월09일 |
| 8. 등록사항 | 저작자 : 홍원표,
창작 : 2020.12.03 |

「저작권법」 제53조에 따라 위와 같이 등록되었음을 증명합니다.

2020년 12월 09일

한국저작권위원회



Part B 개인 작성

B.1 Relationship of Individual Project Assignment to Overall Project

- 팀원 1: 홍원표

서버와 게임 서비스라는 두 가지의 틀로 나뉜다. 서버 측에서는 Account Information, Character Information 블록이 존재하고 게임 서비스 측은 Login Page, Character Edit Page, Algorithm Edit page, Game Play Scene 블록이 존재한다.

- Account Information

- Json 형태로 유저의 account 정보가 저장된다.
- 유저 계정 정보가 담긴 DB 이다. Json 형식으로 만들었고 보안을 위한 암호화가 가능하도록 하였다. 계정의 아이디, 비밀번호, 생성일, 레벨, 계정이 가진 캐릭터 ID 들에 대한 정보가 들어있고 로그인을 하면 DB 에서 위 정보들을 불러와 캐릭터 데이터를 생성하고 이를 토대로 플레이 하도록 하였으며 플레이 도중 데이터가 바뀔 경우 즉시 즉시 저장되도록 하였다.

- Character Information

- Json 형태로 캐릭터 정보가 저장된다.

- Login Page

- 게임이 실행된 후 첫 페이지의 로그인 페이지
- 게임이 실행된 후 첫 번째로 실행되는 페이지로 이때 유저의 정보 DB 와 유저의 정보를 통해 얻을 수 있는 캐릭터들의 IDs 를 통해 캐릭터 정보들을 가져오도록 하였다.

- Character Edit Page

- 로그인 후 Home 화면에서 캐릭터 페이지로 이동 가능
- 해당 계정이 가진 캐릭터의 정보 및 아이템을 표시
- Login 할 때 얻은 캐릭터 정보들을 보여주는 곳으로 이곳에서 각 캐릭터의 스탯 및 착용하고 있는 장비들을 확인할 수 있고, 캐릭터에게 장비를 착용,해제 할 수 있도록 하고 이를 하는 즉시 DB 에 적용한다.

- Algorithm Edit Page

- 캐릭터 페이지에서 해당 캐릭터의 (코드 수정)을 통해 이동 가능
- 캐릭터의 알고리즘 수정 가능

- Game Play Scene

- Algorithm Edit Page 에서 짠 코드를 이용한 캐릭터들이 전투

- 팀원 2: 임예찬
 - Character Information
 - 캐릭터 정보가 담긴 Data Base 이다. 유저의 캐릭터와 적 캐릭터가 존재하기 때문에 Json 형태의 DB 두 개를 만들어 저장하였다.
 - 항목은 캐릭터 아이디, 캐릭터 아이템, 캐릭터 스텟, 캐릭터 알고리즘 노드 및 노드위치가 있고 이는 모든 캐릭터에 적용된다.
 - Algorithm Edit Page
 - 핵심기술인 캐릭터의 AI 를 만드는 페이지이다.
 - Json 형태의 DB 에 저장된 유저가 가진 액션 및 조건블록들을 각 분류별로 나열하였고 이를 드래그 앤 드롭하여 본인 캐릭터의 코드에 붙이면 붙인 코드대로 해당 캐릭터가 움직이도록 디자인되었다.
 - Game Play Scene
 - Algorithm Edit Page 에서 작성한 캐릭터의 알고리즘대로 캐릭터가 상대 캐릭터와 전투하는 씬이다.
 - 이를 위해 캐릭터 알고리즘 DB 와 연동하여 내 캐릭터가 매 프레임마다 코드를 체크해 코드대로 움직이도록 하였다.
 - 또한 3D 를 사용한 전투씬이기 때문에 코드대로 움직일 때 캐릭터의 각도, 속도, 카메라의 이동 등등을 고려하여 내 캐릭터와 상대 캐릭터가 한 씬에서 존재하도록 하였다.

B.2 Economical, Industrial and Social Effects:

- 팀원 1: 홍원표

- 기술적 효과: 핵심기술인 cluster mapping 은 단순히 게임 캐릭터의 알고리즘을 시각화한 것에 그치지 않고, 각종 사물 인터넷의 행동 패턴을 일반인(컴퓨터공학 비전공자)에게 쉽게 이해시킬 수 있는 중요한 수단이 될 것이라 기대한다.
- 경제적 효과(시장형성): 앞서 말한 바, 코딩 기능성 게임 시장은 증가하고는 있지만 새롭거나 획기적인 방법은 전무하다. 프로그래밍을 배우는 것이 꼭 언어를 배우는 것이 기초가 아닌, 알고리즘이 어떻게 진행되는 것인지를 알려주는 것은 BRAIMMO 가 가진 차별성이다. 이를 바탕으로 코딩 기능성 게임 시장이 새롭게 개척될 것으로 판단한다. 나아가, 기능성 게임 뿐 아니라 현재 모바일 게임 시장을 장악하고 있는 RPG 산업에서도 자동 사냥에 해당되는 캐릭터 AI 를 개발자만 수정할 수 있는 것이 아니라 유저가 직접 자신의 캐릭터를 수정할 수 있는 권한을 주어 더 다양한 방법으로 게임 플레이가 가능해지는 것도 기대해 볼 수 있다.
- 사회적 효과(보건, 안전, 환경, 문화): 초등학생의 게임 시간은 일주일에 14 시간이라고 한다. 이 시간이 만약 교육에 밀접한 관련이 있다면 분명 학부모님들도 자녀가 게임을 하는 것에 큰 지장이 없을 거에 동의한다고 생각한다.

기독교 세계관적 함의: 배워서 남주자 - BRAIMMO 는 사용자에게 알고리즘에 대한 기본 이해도를 선물할 수 있다. 사용자들은 이 이해도를 바탕으로 자신이 새로 개척한 알고리즘을 통해 각 캐릭터의 특성에 최적화된 새로운 알고리즘을 만들어, 공유하며 다른 사용자들에게 선물할 수 있다. 이렇게 서로의 생각이 쌓이고 공유가 되어, 더욱 더 사람처럼 행동할 수 있는 AI 가 만들어질 수 있다. 이러한 AI 는 게임 속 캐릭터가 아닌 로봇에게도 영향을 줄 수 있다. 이렇게 배우고 남주는 모토는 우리 프로젝트를 통해 전달될 수 있다.

- 팀원 2: 임예찬

- 기술적 효과: 핵심기술인 cluster mapping 은 블록코딩 형식이 대부분인 온라인 코딩 교육 플랫폼에 새로운 대체제가 될 것이라 기대된다.
- 경제적 효과(시장형성): 이때까지는 코딩교육 시장에서 대부분이 집중한 것이 코딩을 짜는 기술이었지만 BRAIMMO 는 코딩에서 핵심인 알고리즘 능력을 향상시켜주는 것으로, 알고리즘 교육이라는 새로운 시장을 개척할 것으로 기대된다. 이에따라 단순히 코딩을 배우고 싶어하는 학생 뿐만 아니라 알고리즘적인 사고력을 키우고 싶어하는 유저 혹은 단순히 게임을 즐기는 유저 모두를 포함하는 시장이 형성될 수 있다.

- 사회적 효과(보건, 안전, 환경, 문화): 대부분의 학부모님들이 게임은 쓸모없는 것이라 생각하고 자식들이 게임하는 것을 꺼려한다. 하지만 BRAIMMO 를 통해 게임을 즐기고 집중해서 하면서도 알고리즘을 배울 수 있다는 것을 인식한다면 BRAIMMO 뿐 아니라 다른 교육용 어플리케이션이 시장에 많이 나와 게임에 대한 인식을 바꾸고 즐기고 몰입하면서 교육을 할 수 있는 콘텐츠들이 많이 생길 것이라 기대된다.
- 기독교 세계관적 함의: 배워서 남주자-코딩교육이 주목을 받으며 코딩 사교육 시장이 계속해서 커지고 있다. 하지만 많은 수의 학생들은 코딩 사교육을 받을 만한 경제적 상황이 뒷받침이 되지 않아 교육의 빈부격차가 생길 것이다. BRAIMMO 는 배운 컴퓨터공학적 지식을 활용해 학생들이 무료로 알고리즘적 사고를 키울 수 있는 콘텐츠를 배포하여 배운것을 나눈다.

B.3 Engineering Ethics

■ 팀원 1: 후원표

1. 학생, eSports, 공교육 시스템(지도 교사, 교수 등), 사교육 시스템, 사물 인터넷 개발자
2. 캐릭터 육성에 있어서 알고리즘이 아닌 아이템에 너무 의존하게 되는 게임 밸런스 붕괴. 아이템 과금과 획득 방법을 사행성에 연관하여 매출을 유도할 경우 주 타겟층인 학생들에게 악영향을 끼칠 수 있다.
3. 아이템, 스탯에 치중되지 않도록 캐릭터의 알고리즘과 노드들에 대한 연구와 꾸준한 업데이트 필요.

아이템 획득 방법에 있어서 과금은 제외한다.

결제 한도를 세운다.

사행성이 아닌 사용자의 이용도와 레벨에 따라 획득권을 준다.

■ 팀원 2: 임예찬

1. 코딩교육 플랫폼 제작사, 공교육 코딩 교사, 학생, 사교육 코딩 교사
2. 수익을 창출하기 위해 게임의 결과에 영향을 끼치지 않는 선에서 캐릭터 스킨 및 이펙트에 과금을 부가하는데 경제관념이 없는 저학년의 학생들이 과도한 과금을 할 수 있다.
3. 월 결제 한도를 낮게 설정한다.
 - 과금시스템을 없애고 제품을 한 번 구매하면 평생 이용하도록 한다.

B.4 전문인으로서 자기주도적 평생학습 계획

● 팀원 1: 홍원표

2019년 한동대학교에 편입하며, 필수 교양 수업들은 모두 획득한 상태여서 전공 수업으로만 학업을 진행하였다. 빠르고 방대하게 전공 지식을 쌓으며 '나의 지식을 어떻게 활용할 수 있을까'라는 의문을 창업이라는 키워드로 풀 수 있었다. 현재 발전의 속도가 가장 빠르고 정부의 막대한 지원을 받고 있는 AI를 어떻게 하면 대중들에게 쉽게 다가가게 할 수 있을 지 생각을 하며, 알고리즘에 대한 이해를 쉽게 풀어주는 기능성 게임에 해답을 찾을 수 있었다.

여태 게임 개발을 처음부터 끝까지 스스로 해본 적이 없었던 나는 C# 이라는 언어를 처음 접하게 되었고 독학하여 개발을 진행했다. 2D와 3D 캐릭터 제작과 캐릭터에 필요한 연산 등을 연구하며 RPG 장르에서 요구되는 factor들을 스스로 찾아 공부했다. 실시간 대전 게임이라는 특성을 고려하여 연산 속도를 최대한 빠르게 하기 위해 최적화된 알고리즘 등을 스스로 찾아 공부를 해야 하는 상황도 있었다. 게임에 필요한 Asset, Script를 직접 개발하며 500개가 넘는 Script들이 만들어졌다. 단순히 아이디어만 가진 것이 아닌 충분한 개발을 거친 우리 프로젝트는 여러 공모전에서 수상을 하고 캡스톤 페스티벌 대상이라는 개발 업적을 낼 수 있었다.

하지만 AI라는 키워드를 사용하면서 정작 개발자인 내가 진정한 AI가 무엇인지 모른다고 판단하였고, 머신 러닝 AI쪽으로 대학원을 가서 AI라는 단어를 이해함에 있어 부족함이 없는 사람이 되고 싶다는 결정을 하였다.

이번 학기를 한동대학교의 마지막 학기로 두고 있는 시점에서, 성균관대학교의 인공지능 융합학과에 가서 공부를 지속할 예정이다. 이 프로젝트를 통해, 그리고 조성배 교수님을 통해 창업의 기본에 대해서 많은 것을 배울 수 있었다. 나의 이러한 경험을 가지고, 그리고 인공지능에 대한 심층적 이해도를 쌓은 후 나는 새롭고 더 참신한 BRAIMMO를 만들어내고 싶다.

- **팀원 2: 임예찬**

2015년 1학기 컴퓨터 공학 기초 수업들을 듣고 2019년 2학기에 컴퓨터 공학 전공으로 다시 전과를 했기 때문에 전공과목들에 대한 기초가 부족했다. 따라서 이후 전공과목을 듣는데 제한점이 많았다. 따라서 먼저 전공 과목들을 들으며 정공 기초를 먼저 따로 공부하였다.

먼저, 코딩에 대한 기초가 부족하다고 판단되어 C언어 및 Java언어를 공부하였다. 먼저 시중에 있는 C,Java 언어 책들을 구매하여 공부를 진행하였다.

공부를 하던 중 보안분야가 궁금하여 보안분야 공부를 하였다. 보안분야에 대해서는 수업을 들은 적도 없었고 이 분야에 아는 사람이 없어 먼저 구글링 및 유튜브를 통해 공부 방법을 배웠다. 그 후, 배운 공부 방법을 토대로 웹 보안부터 공부를 하였다.

또한 캡스톤에서 C#을 사용하며 Json 및 Git을 사용하였는데 이와 관련된 수업을 들은 적이 없어 구글링 및 유튜브를 통해 공부를 하여 캡스톤에서 활용하였다.

요즘은 알고리즘 공부를 하고 있다. leetcode 코드를 통해 꾸준히 코딩문제를 하고 있다.

보안 분야에 관심이 많았지만 어떤식으로 공부의 방향을 잡기 굉장히 힘들었다. 올해 12월 말부터 포렌식 회사에서 현장실습을 한다. 현장실습을 하면서도 개발자로서 필요한 코딩 기초를 쌓기 위해 leetcode를 통해 꾸준히 코딩공부를 할 예정이다. 또한, 공부를 하다보면 앞으로 개발자로 살아가는데 있어 어떤 공부를 해야할 지에 대한 가닥이 잡힐 것이라 생각된다. 현장실습 이후에는 부족한 부분을 꾸준히 공부할 것이다.

또한 꾸준히 영어공부하는게 중요할 것이라 생각된다. 따라서 2019년 복학을 한 이후로 꾸준히 영어드라마를 통해 영어공부를 하고 있다. 영어작문쪽이 부족하다고 생각돼 현장실습을 하며 작문을 공부할 예정이다.

B.5 프로젝트 수행 개인별 소감

- **팀원 1: 홍원표**

항상 창업에 대한 생각만 가지고 실천을 제대로 할 기회가 없었다. 누군가와 생각을 나눈다고 해서 상대가 제대로 들어줄 것이라고 생각하지 않았다. 그리고 창업을 실천하려면 자본도 부족하고 웬만한 근성이 아니면 성공할 수 없다고 생각했었다. 하지만 이번 프로젝트를 수행하면서 내가 가진 부정적인 생각들을 고칠 수 있었다. 나의 아이디어를 진심으로 들어줄 사람, 나와 뜻을 같이 해줄 사람, 나를 격려해주고 응원해주는 사람은 항상 존재했었는데 내가 찾지 않았던 것이었다. 앞으로 나를 좀 더 믿고 나를 믿는 사람들에게 보답하는 삶을 살 수 있도록 노력할 것이다.

- **팀원 2: 임예찬**

코딩에 대한 자신이 없는 상태에서 캡스톤을 진행하였다. 캡스톤을 하며 부족한 부분이 굉장히 많았는데 캡스톤을 하며 코딩을 공부하며 실력을 키울 수 있었다. 또한 코딩뿐만 아니라 우리 프로젝트는 창업을 준비하는 것이었기 때문에 아이템을 어떻게 기획하고 기획한 것을 모두 함께 기획하는지에 대해 배울 수 있었다.