



국민대학교  
소프트웨어융합대학  
소프트웨어학부

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	리듬액션 게임 제작
팀 명	45
문서 제목	중간보고서

Version	1.5
Date	2022-04-04

팀원	서 정욱 (팀장)
	김 민형
	이 승하
지도교수	김 형균 교수

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "액션리듬게임"을 수행하는 팀 "45"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "45"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역

<b>Filename</b>	중간보고서-액션리듬게임.doc
<b>원안작성자</b>	이승하, 서정욱, 김민형
<b>수정작업자</b>	이승하, 서정욱, 김민형

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-04-02	이승하	1.0	최초 작성	
		1.1	내용 수정	수정된 연구내용 추가
		1.2	내용 수정	향후 추진 계획 수정
2022-04-03	김민형	1.3	내용 수정	연구 내용 작성
2022-04-03	서정욱	1.4	내용 수정	여러 항목내용 추가 및 수정
2022-04-04		1.5	내용 수정	자료 내용 추가

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 목 차

1	프로젝트 목표 .....	4
1.1	프로젝트의 선택 이유.....	4
1.2	프로젝트의 목표.....	5
1.3	프로젝트의 개발 방향성 .....	5
2	수행내용 및 중간결과 .....	7
2.1	개발 계획 및 수행단계 .....	7
2.1.1	분업 .....	7
2.1.2	개발환경 및 협업환경 .....	8
2.1.3	수행단계 .....	9
2.2	용어 정리 .....	10
2.3	게임의 기획(Game design) 및 연구내용 .....	11
2.3.1	레퍼런스 벤치마크(Reference Benchmark).....	11
2.3.2	리듬게임의 특성 분석 .....	13
2.3.3	게임 디자인 (게임기획) .....	14
2.4	수행내용.....	24
3	개발 중 개선사항 및 문제해결 .....	30
3.1	수정사항.....	30
4	향후 추진계획 .....	32
4.1	향후 계획의 세부 내용 .....	32
5	고충 및 건의사항 .....	33

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

# 1 프로젝트 목표


## 1.1 프로젝트의 선택 이유

프로젝트를 시작하게 된 배경은 게임을 만드는 것을 목적으로 시작하게 되었으며 그림이나 음악 등의 예술적 요소들을 적극적으로 활용할 수 있는 게임을 만들기로 하였습니다. 이 중 여러 요소들 중 다음과 같은 요소들을 만족하는 프로젝트를 진행하고자 하였습니다.

1. 예술적인 요소를 접목한 게임
2. 실제로 서비스가 가능하며 경쟁력이 있을 만한 게임
3. BM모델이 확실한 게임

이것들을 만족하는 게임들 중 리듬게임을 고려하게 되었는데 리듬게임을 갖는 장점은 다음과 같습니다.

- 변화의 대응이 쉽다.  
기본적으로 게임의 구성이 Note파일을 통해 구성이 되는 형태이며 게임의 형식이 어느정도 변경되더라도 기본적인 데이터 형식들은 크게 바뀌지 않아 변화의 대응에 강합니다.
- 빠르고 쉬운 프로토타이핑  
쉬운 알파테스트(내부테스트)를 통하여 쉽게 게임에 여러 요소들을 반영할 수 있습니다. 여러 플레이 데이터가 쌓이고 나서야 직접적인 변화를 줘야하는 타 장르들에 비해 시간적으로 확실한 강점이라 볼 수 있습니다.
- 게임심의  
게임 특성상 여러 확률적 요소들(사행성)을 넣지 않더라도 충분히 완성도 있고 재미있는 게임 디자인이 가능하며 이는 게임심의에서 매우 유리합니다. 우리나라 특성상 유통되는 게임은 심의를 거치는 것이 중요하며 특히 PC플랫폼의 경우 더욱 신경 써야 하는 부분입니다. 여러가지 이유로 이용등급이 높아지거나 심의

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

가 오래 걸리는 등 여러 문제가 발생할 수 있는데 리듬게임은 이런 부분에서 비교적 자유롭다고 볼 수 있습니다.

- 분업이 쉬운 협업 친화적 개발  
게임에서 Mapper / 기획자 / 스토리 / 그래픽개발 / 에디터개발 / Stage 개발 / 작곡 / 기타 아트 등 여러가지 일을 분할하여 작업하기 쉬운 점이 장점입니다.
- 명확한 이용자층  
게임에서 명확한 이용자층과 시장이 있어 플레이어들의 니즈를 어느정도 겨냥할 수 있다는 것이 장점이며 유행을 크게 타지 않는 다는 것도 큰 장점입니다.

따라서 최종적으로 결정한 것은 리듬게임을 만드는 것으로 결정하였으며 기존에 나와있는 플레이방식에서 조금은 색다른 게임을 만드는 것을 목표로 합니다.

## 1.2 프로젝트의 목표

프로젝트의 최종 목적은 멀티플랫폼(PC, 콘솔, 모바일 등)으로 플레이 가능한 독특한 플레이방법을 가진 게임의 완성입니다.

현 프로젝트동안 아트, 스토리, 음악 등의 요소는 부족한 부분이 많아 완벽하게 해결할 수 없어 아트부분을 제외한 PC에서 플레이 가능할 정도의 게임 완성이 목표라고 할 수 있습니다.

## 1.3 프로젝트의 개발 방향성

### 1. 협업 친화적

이번 프로젝트를 진행하며 가장 큰 개발 방향성은 협업 친화적인 개발입니다. 여러 요소들 중 다른 파트와 충돌이 최대한 발생하지 않도록 게임개발을 진행중이며 이는 저희 프로젝트의 가장 큰 요소 중 하나라고 생각합니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 2. 독특한 게임 플레이 방식

기준에 많이 나와있는 형식(떨어지는 노트를 처리하는 형식)의 게임이 아닌 조금은 색다른 플레이(악기선택, 바깥 게임(Map), 액션게임의 형태)를 할 수 있는 게임을 만들고자 합니다.

## 3. 플레이어의 재미와 특성을 고려한 개발

리듬게임을 좋아하는 사람들이라는 특이한 타깃을 공략하는 만큼 그 니즈 맞는 개발을 하려고 합니다. 특히 하드코어 유저(숙련된 플레이어)와 라이트 유저(가볍게 플레이하는 유저)의 스펙트럼이 굉장히 넓은 게임의 특성상 이에 대한 레벨디자인을 고려하면서 개발 중입니다.

## 4. 실제 경쟁력이 있는 서비스를 고려한 개발

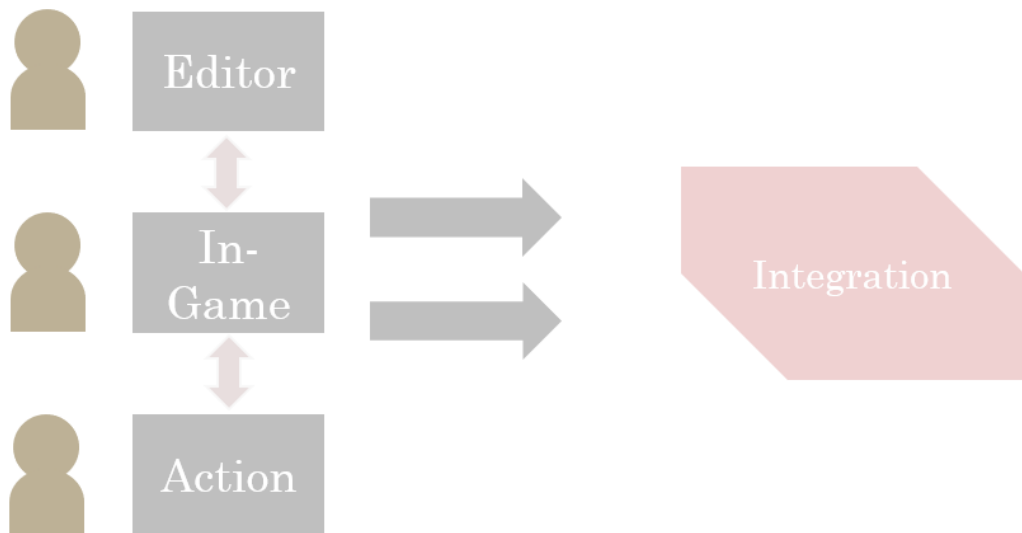
단순히 실험적인 게임을 만들어 파는 것이 아닌 실제 시장에서 경쟁력을 갖기 위한 여러가지를 고려하며 개발 중입니다. 따라서 처음부터 게임의 BM에대하여 고려하고 시작하였으며 사소한 요소 하나하나를 마땅한 근거를 통해 조심스럽게 추가시키고 있습니다.

## 2 수행내용 및 중간결과

### 2.1 개발 계획 및 수행단계

#### 2.1.1 분업

크게 여러 파트들이 있겠지만 현재 팀원이 3명인 관계로 업무를 크게 다음과 같이 분할하여 진행하였습니다.




- 김민형 Editor 제작부분

Map 및 레벨 디자인을 맡은 사람들이 직접 다루어 소프트웨어 개발을 못하는 사람이라도 게임을 만들 수 있도록 할 수 있는 에디터를 제작하는 부분을 담당합니다. In Game 파트와 레벨디자인, 게임 디자인 파트와 깊은 연관이 있는 파트입니다.

- 서정욱 In-Game 제작부분

그래픽의 요소가 강한 Action파트와 게임의 데이터의 요소가 강한 Editor 파트를 모아 실제 플레이 가능한 게임이 되도록 합치는 역할을 맡고 있으며 따라서 게

	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

임 Stage기물의 실제적인 작동에 대한 부분을 담당합니다. 또한 사운드 출력에 관련된 처리(Mixing)들 또한 함께 하고 있습니다.

- 이승하 Action 제작부분

게임의 그래픽적 요소들인 플레이어, 노트의 상호작용(피격, 움직임, 공격 등)과 상태, 카메라 움직임 등 효과 등 그래픽으로 보여주는 파트를 담당하고 있으며 시각적인 요소들에 대한 것들을 많이 담당하는 게임의 아트파트와 깊은 관련이 있습니다.

## 2.1.2 개발환경 및 협업환경

- 개발엔진

유니티(2021.2.14f1)


비록 다른 엔진에서는 후술할 사운드적 문제가 비교적 적지만 개발 규모, 방대한 Asset 자료의 활용 가능성, 개발난이도, 개발기간 등을 고려하였을 때 유니티가 가장 적합하다고 생각하여 해당 엔진을 통해 개발하게 되었습니다.

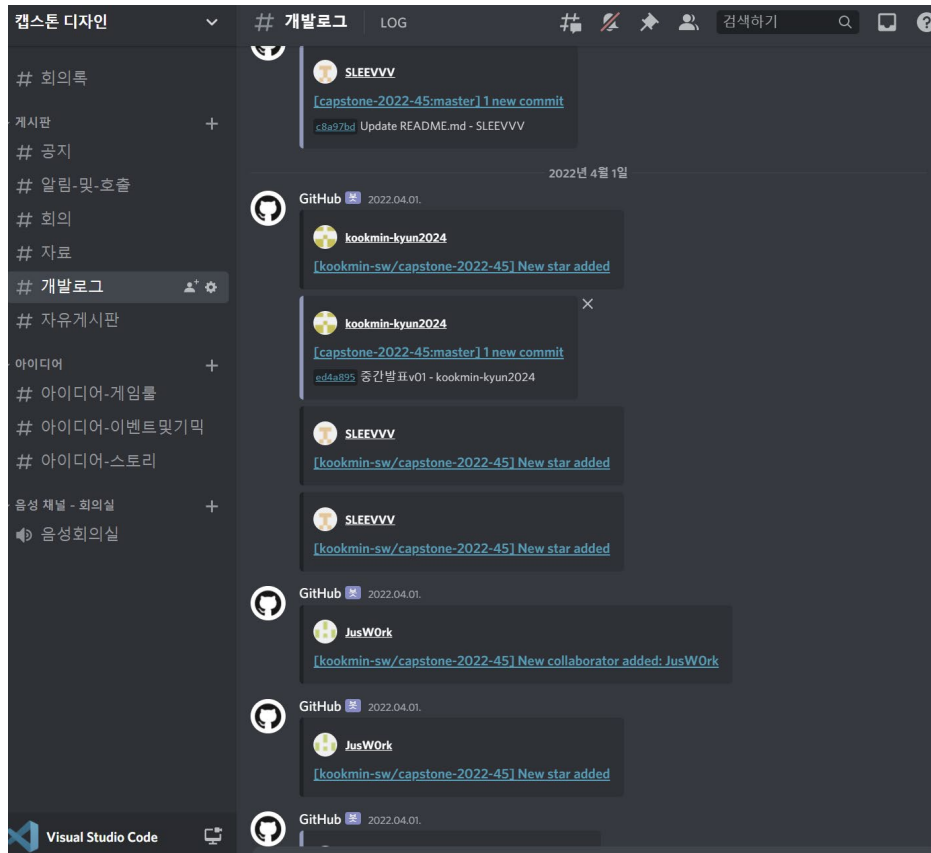
- 협업 툴

Discord

Slack과 Classum, 게더를 통해 협업할 수 있도록 공간을 마련해 주셨지만 이용경험이 있고 음성 회의 및 화면공유가 간단하며 비교적 가벼운 프로그램인 Discord를 통해 대부분의 프로젝트를 진행하게 되었습니다. 또한 깃허브와 연동을 통해 프로젝트의 진행을 확인할 수 있는 점도 장점입니다. (자료1)



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4




(자료1) 디스코드 활용장면

### 2.1.3 수행단계

게임 개발의 진행은 다음과 같은 순서로 진행하기로 했습니다

1. 프로토타입 완성을 위한 게임 룰, 기믹 등을 확실하게 정하기
2. 여러가지 메인 기믹을 추가해 prototype 완성
3. 비주얼 적 요소, 여러 부가적인 기믹 및 이벤트(카메라, 이펙트)들을 추가(게임의 틀이 어느정도 완성되었을 경우 스테이지 와 제작 툴 개발
4. 스테이지 구성, 스토리 및 게임의 프로그래밍 외적 요소 완성
5. 딜레이 문제등 세부적인 문제들을 해결해서 게임의 완성도를 높인다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 2.2 용어 정리

들어가기 앞서 해당 게임 분야에서 통념적으로 많이 사용하는 단어들이나 게임의 특성상 많이 나오는 단어에 대하여 미리 기술합니다.

- BPM : Beats per minute 한 박자(4beat)를 갖는 Beat 가 1분동안 재생되는 횟수를 타넵니다.
- 박(beat) : 시간을 나타내며 4beat 는 한 박자를 나타내며 1/2이 될수록 두 배 증가합니다. (8beat 는 반 박을 나타냅니다.)
- 노트(Note) : 게임 중 처리해야 하는 요소들(적, 스테이지기물, 발판등)을 칭합니다.
- 스테이지(Stage) : 실제 플레이가 이루어지는 화면구성을 이루는 모든 것을 뜻하며 플레이어와 상호작용하게 되는 여러 요소들을 나타냅니다.
- 채보 : 위의 Note의 정보를 담은 Stage의 데이터나 파일을 칭합니다.
- Scene : 게임을 구성하는 화면들을 칭합니다. (ex. 인게임씬(Stage), 모드선택씬 등)
- 맵(Map) : 게임에서 스테이지를 이동가능한 바깥 게임을 칭합니다.
- 타이밍바 : 플레이어에게 박자를 알 수 있도록 타이밍을 시각적으로 제공하는 오브젝트입니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 2.3 게임의 기획(Game design) 및 연구내용

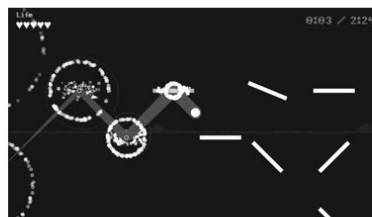
### 2.3.1 레퍼런스 벤치마크(Reference Benchmark)

- 리듬게임의 플레이에 따른 분류

리듬게임의 플레이 방법에 따라 3가지로 분류할 수 있습니다.



노트를 처리하는 형태



캐릭터가 목표지점으로 이동



복합형

#### (자료2) 레퍼런스 게임 예시

- 1 일반적인 노트를 처리하는 형태

일반적인 노트를 처리하는 형태이며 레일을 따라 떨어지는 노트를 처리하며 점수를 얻게 됩니다. 리듬게임의 거의 대부분을 차지하는 형태이며 게임이 직관적이고 정확한 타이밍에 누른다는 리듬게임형식에 가장 알맞은 형태이기 때문에 가장 친숙하고 대부분의 리듬게임 유저는 이 형태의 게임에 속합니다.

- 2 캐릭터가 목표지점으로 이동하는 형태

일반적인 노트를 처리하는 형태 다음으로 많은 형식이며 어려운 스테이지 구성으로 게임의 클리어를 목표로 두게 하는 형태입니다. 약점으로는 시각적인 정보를 해석하기 위한 시간이 많이 걸림으로 다양한 버튼을 활용한 방식을 쓰기엔 무리가 있습니다. 따



라서 해당 방식은 키입력을 하나만 사용하거나 여러 키가 다 같은 일을 수행하는 One Key 형식의 플레이 방식이 많습니다.

#### - 복합형

비교적 최근 나온 게임 형식이며 따라서 이와 같은 형태의 게임은 많지 않습니다. 확인된 게임만 5개 안팎으로 스토리 적인 연계를 긴밀하게 하거나 다양한 장르(횡스크롤 액션, 로그라이트 등)의 장점을 흡수하여 색다른 플레이 방식을 제공하는 것이 큰 차이입니다. 다만 다소 난해한 플레이 방법으로 인해 라이트 유저들의 접근이 어렵고 하드코어 유저들 또한 깊게 플레이하는 것(다회차 플레이)보다는 적은 횟수로 가볍게 즐기는 스타일이 많으며 이는 하드코어 유저층이 즐기에도 라이트 유저화 되어 게임을 온전히 즐길 수 있는 유저의 스펙트럼이 좁아질 수 있다는 것을 의미합니다. 원인으로서는 다른 장르의 혼합과정에서 직관성이 떨어지고 단순히 클리어에 중점을 두어 리듬게임의 최종 목적 중 하나인 Perfect 플레이(모든 노트의 최고 점수 클리어)혹은 올 콤보(모든 노트의 처리)에 대한 요소가 희박하기 때문이라고 보여집니다.

#### - 분석내용

대부분의 리듬게임을 차지하는 일반적인 노트를 처리하는 형태는 게임의 아트적인 컨셉, 음악, 레벨디자인으로 승부를 보기 때문에 해당 부분은 저희가 갖는 약점이라 생각하여 플레이 방식과 아이디어로 승부를 볼 수 있는 복합형을 선택하게 되었습니다. Stage 내에서의 여러 노트들은 몬스터, 함정, 지형 등의 형태로 나타내게 됩니다. 해당형식의 약점은 극복하기 위해 여러 스테이지 구성들을 명확하게 분류하고 높은 직관성과 퍼펙트 플레이로 이루어 질 수 있는 요소들을 첨가하였습니다.

#### ● 리듬게임의 컨셉에 따른 분류

#### - 1 일반적인 리듬게임(일렉트로니카)

일반적으로 빠른 BPM 을 가지고 있는 노래들이 플레이시 신나고 재밌기 때문에 해당특성을 갖는 곡들이 주력이며 일렉트로닉뮤직 세부적으로는 하드코어 테크노, 트랜



스 등의 곡을 많이 사용하게 됩니다. 따라서 전자적인, 사이버적인 느낌을 많이 주는 컨셉을 채용한 리듬게임이 많으며 대부분의 게임을 차지합니다. 또한 고난도 곡을 만들기 쉬워 하드코어 유저층이 많습니다.

## - 2 감성 계열

어려운 곡을 클리어 한다는 것 보다는 여러가지 악기를 연주해 보는 컨셉을 가진 형태가 많으며 음악이나 특유의 분위기로 승부를 보는 경우가 많습니다.

## - 분석내용

이미 대부분의 리듬게임이 가지고 있는 형태보다는 특이한 형태의 게임을 만들기로 정한 만큼 감성계열에 가까운 느낌을 주도록 진행하게 되었습니다. 악기를 연주하는 느낌을 보다 강하게 주도록 같은 곡이라도 다른 악기로 교체해서 플레이가 가능한 형태로 진행하게 됩니다.

## ● Note 데이터 벤치마크

게임의 여러 요소들을 나타내는 Note에 대한 데이터를 어떻게 표현할지에 대하여 리듬게임에서 가장 유명한 규격 중 하나인 BMS(Be-music script)라는 시뮬레이터와 노트 파일 규격을 벤치마킹하였으며 파생된 형태중 하나인 Bmson 이라는 형태와 가상 유사하게 Json의 형태로 Note 게임데이터를 입출력하게 됩니다.

## 2.3.2 리듬게임의 특성 분석

## ● 게임의 주요 플레이어의 구분

## - 라이트 유저

비교적 가볍게 플레이 하는 유저들이며 음악이 좋아서, 악기를 연주하고 싶어서, 특유의 분위기가 좋아서 하플레이 하는 유저들이 많으며 감성계열의 비중이 높은 형태

	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

입니다.

#### - 하드코어 유저

숙련자들이 많고 퍼펙트 플레이, 올 콤보, 하이스코어등 기록경쟁에 적극적이며 해당 장르에 높은 충성도를 가지고 있는 경우가 많습니다.

#### - 분석

라이트유저와 하드코어 유저의 플레이 유형과 숙련도가 크게 차이가 나며 넓은 스펙트럼으로 인해 다양한 난이도를 제공하거나 해당 유저들을 따로 공략할 필요가 있습니다.

#### ● 게임이 BPM 단위로 진행됨.

일반적인 게임과 다르게 리듬게임은 프레임단위가 아닌 BPM단위로 움직이므로 오디오와 게임 사이의 작은 딜레이에 민감하게 반응합니다. 유니티로 리듬게임을 개발할 때 기본적으로 제공되는 Update() 같은 프레임에 의존한 시간단위를 사용하게 된다면 규칙적인 실행과 정확한 시간을 보장받지 못하게 됩니다.)

### 2.3.3 게임 디자인 (게임기획)

매주 여러 번의 회의를 통하여 게임의 복잡한 룰을 명확히 하고 여러 세부적인 요소에 대한 토론을 하고 있으며 게임 디자인(기획)은 계속 진행되고 있습니다.



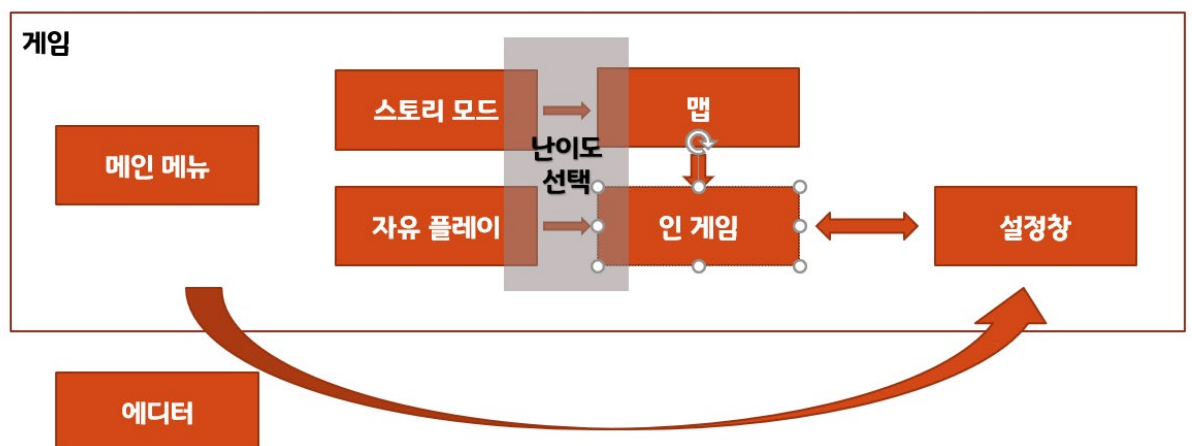
(자료3)게임 디자인 자료

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

### ● Scene의 흐름

화면을 구성하는 Scene의 구성은 다음 과 같습니다(그림3). 게임의 설정창은 메인 메뉴와 인 게임 Stage Scene에서 이동이 가능하며 본 게임의 입장 전 플레이어는 스토리모드 또는 자유 플레이를 선택할 수 있습니다. 스토리 모드는 서사 구조로 이루어지기에 이 게임의 바깥게임이라 할 수 있는 맵Map을 플레이 할 수 있으며, 자유 플레이에서는 원하는 곡 선택을 통해 본격적인 게임에 진입할 수 있습니다.

## 게임구성 – Scene의 흐름



### (자료4) Scene 흐름

또한 중간 스토리모드에서 Map으로 넘어가는 시점, 자유플레이에서 인게임으로 넘어가는 시점에 난이도를 선택할 수 있는데 난이도를 선택은 다음과 같이 이루어집니다.



(자료5) 난이도 선택과정

- 키 입력

플레이어는 다음과 같은 키 입력으로 게임을 진행합니다

D키	왼쪽 이동
K키	오른쪽 이동
FJ키	연속 공격 및 앞으로 전진(몬스터를 여러 번 눌러서 처치할 경우)
Spacebar	상호작용 및 회피, 점프

(자료6) 키 입력



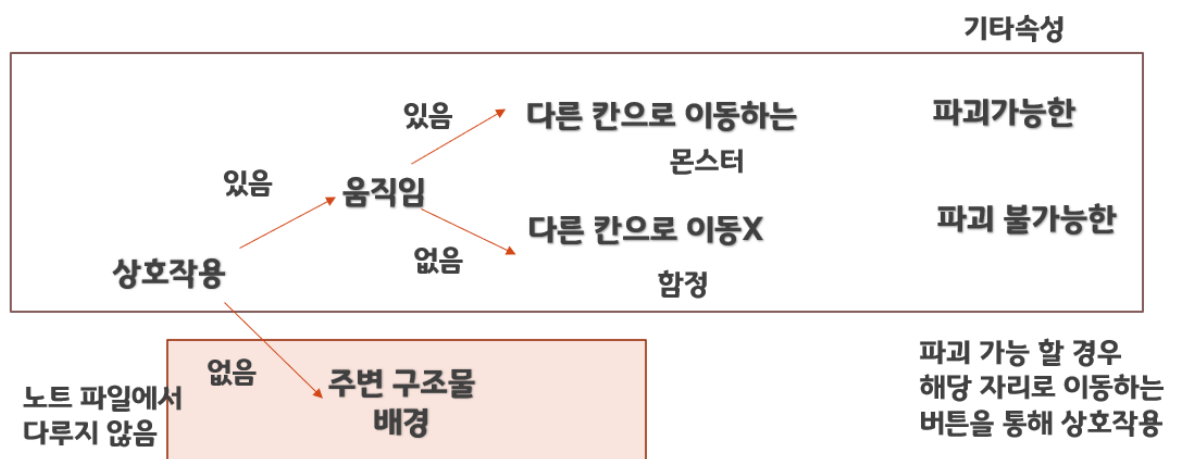
● Note Object

게임에 출현하는 노트 오브젝트 종류는 다음과 같습니다.

Note Object		Note가 아닌 Object
몬스터	처리해야 하는 종류의 노트	기타 지형지물
함정	캐릭터가 피해야하는 종류의 노트	이펙트
지형	캐릭터가 이동 가능한 종류의 노트	


(자료7)Object 특성

플레이어와 오브젝트 간의 상호작용은 다음과 같습니다(그림4). 아무 상호작용이 없는 경우 주변의 구조물이나 배경 및 지형이며, 있을 경우 움직임에 따라 두 가지로 나뉩니다. 움직임이 있는 것은 몬스터로 간주하며 플레이어는 해당 칸으로 이동해서 키 입력을 통해 파괴할 수 있습니다. 움직임이 없는 경우에는 함정이며 해당 칸을 회피하거나 가만히 있어야합니다.



기타 -- 지형object

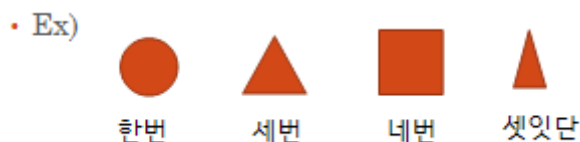
(자료8)Object 속성

	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

- 몬스터(Note)

- 속성 : 움직임 및 이동가능 / 파괴가능
- 동작 및 상태 : 공격 / 이동 / 피격(good) / 피격(perfect) / 회피(bad) / idle
- 종류 : Nbeat Ntime note / Nbeat delay note

몬스터 오브젝트는 파괴가 가능하며 피격, 처리, 공격 애니메이션을 가지고 다른 칸으로 이동 가능한 몬스터일 경우 화살표 또는 예비동작으로 예상 경로를 플레이어에게 보여줍니다. 몬스터는 종류에 따라 타격해야 하는 횟수와 박자가 다르며 그 예시는 다음과 같습니다.(그림5). 플레이어가 몬스터에 대한 판정이 miss일 경우 공격하며, 플레이어의 공격이 Good나 Perfect 판정일 경우 피격 애니메이션이 재생되며, bad 판정일 경우 회피합니다. 판정을 통한 점수에 크게 연관이 되며 주로 하드코어 유저를 저격한 레벨디자인을 할 때 사용될 노트 종류입니다. 현재 구상되어 있는 종류는 두가지이며 Nbeat Ntime노트는 정박에 N번 눌러야 하는 노트타입이며 Nbeat delay note는 엇박자 타이밍에 관련된 노트입니다.

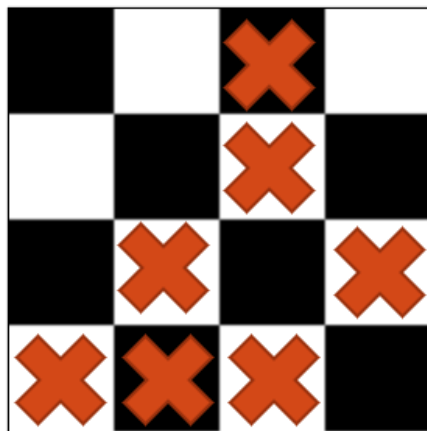


(자료9) 몬스터 종류

- 함정(Note)
- 속성 : 이동 불가능 / 파괴불가능
- 동작 및 상태 : 피격 / idle
- 종류 : Trap / Stop

게임 함정은 기본적으로 파괴가 불가능하며, 키 입력을 통해서 피하거나 아무 행동을 하지 않아야 제대로 통과되는 패턴이 있습니다. 이와 같이 몬스터를 잡음으로써 점수를 획득할 수 있으며, 함정은 점수에 연관되지 않지만 게임 클리어와 관련이 있는 구조입니다. Trap은 회피해야 하는 종류이며 Stop은 해당 박자동안 멈춰야 하는 종류입니다

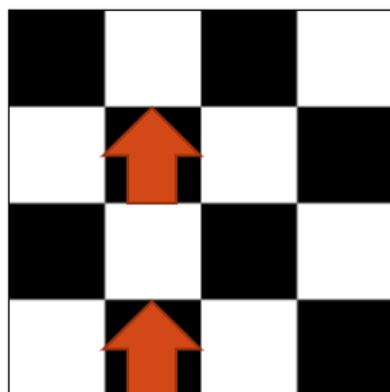
- 이동 예시 Trap (해당 스테이지를 돌파하려면 다음과 같은 키로 진행해야 합니다.)



K  
D  
K  
ForJ

(자료10) Trap 이동

- 이동 예시 Stop



ForJ  
\*(멈춤)  
ForJ  
\*(멈춤)

(자료11) Stop 이동

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

참조)

몬스터와 함정의 점수분리는 하드코어 유저(게임의 숙련도가 매우 높은 유저)와 라이트 유저(숙련도가 낮거나 가볍게 즐기는 유저)의 분리를 위한 디자인입니다.

하드코어 유저의 경우 고득점, 퍼펙트 플레이(모든 판정을 최고점수로 내는 것)에 중점을 두고 라이트 유저의 경우는 스테이지의 클리어에 목적을 두는 경우가 많아 다양한 몬스터와 획득 점수는 하드코어 유저, 단순히 클리어에만 관여하는 함정들은 라이트 유저를 위한 오브젝트들이며 나중에 스테이지의 레벨 디자인에 좀 더 용이 해 질 것입니다.

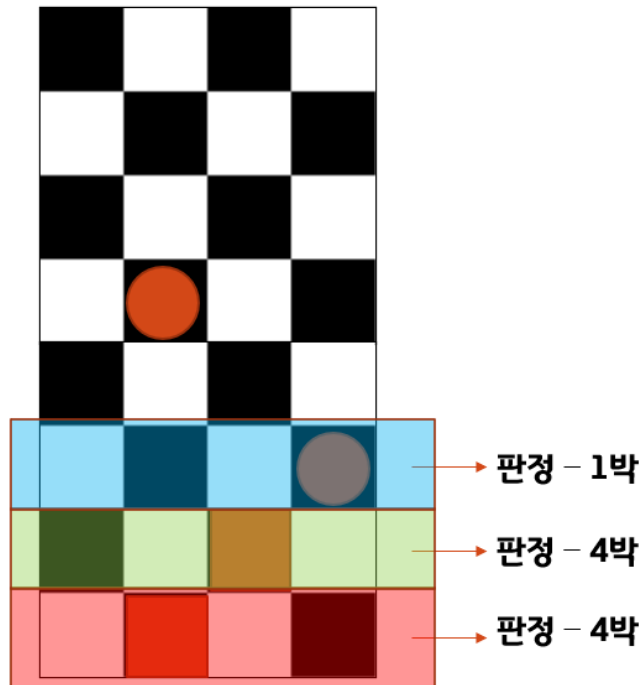
● 지형(Note)



(자료12)기본생성단위

게임의 지형은 4개 칸이 1행으로 연결된 것이 한 단위이며, 스테이지가 스크롤 되는 시점에서 판정이 됩니다. 스크롤(지형의 이동)의 경우 기본적으로 1박자 타이밍(4beat)이며 몬스터 노트의 박자에 따라 스크롤 간격이 바뀔 수 있습니다. 더 빠른 스크롤을 원할 경우 Bpm을 두배로 설정하면 됩니다

- 판정 시점과 스크롤되는 박자



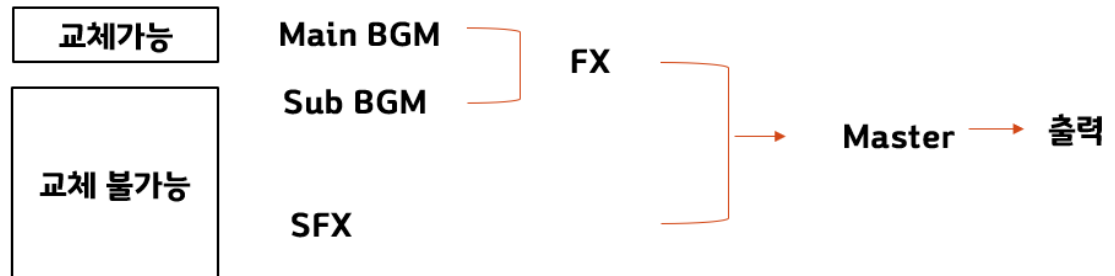
### (자료13)지형구성

- 오디오 멀티트랙
  - Main BGM : 교체가능 / FX(효과)추가가능
  - Sub BGM : 교체불가능 / FX(효과)추가가능
  - SFX : 효과음 교체불가능

게임의 오디오 구성(그림7)은 Multi Track이며 Main BGM, Sub BGM, SFX 효과음, FX로 이루어집니다. Main BGM은 교체 가능한 메인 악기 연주에 따른 음악이며 Note와 매칭이 되는 소리입니다. Sub BGM은 Main이외의 배경음들이 되겠으며, SFX 효과음은 타격음, 버튼을 눌렀을 때의 소리 등 특수효과음들이 되겠습니다. 기본적으로 SFX는 교체가 불가능하나 버튼 타격음의 경우엔 예외적으로 교체가 가능합니다. FX는 음의 Pitch 조절, Distortion 과 같이 소리에서 사용되는 직접적인 특수효과들을 나타냅니다.

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

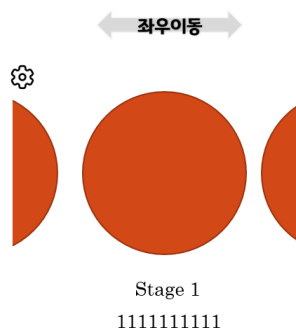
- 사운드 플로우



(자료14)사운드 플로우

- UI(Prototype) – 맵

- 설정 / Stage선택 / 점수 / 이름 / Scene진입:



(자료15)프로토타입 UI구성

UI(Prototype)의 구성은 Space키를 통한 곡 선택 또는 스테이지 진입, 설정, 점수 목록, Scene 진입으로 임시로 만들기로 했습니다. 현재 이부분은 바깥게임(맵)의 형태로 구현 예정이지만 기획되지 않은 부분이기에 프로토타입 UI를 통해 맵의 역할을 대체하고 있습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

- 비즈니스 모델(BM)

본 게임은 많은 게임에서 차용중인 형태인 패키지형태로 완벽한 클리어가 가능 하도록 계획 중이며 여러 플랫폼으로의 포팅 가능성을 위해 호환성에 문제가 될 수 있는 외부 라이브러리 및 Asset을 최대한 자제하며 개발 중에 있습니다.

패키지 형태는 완성된 본편과 확장팩(DLC - 다운로드컨텐츠)로 이루어져 있으며 현재 게임시장에서 주류를 이루는 BM형태인 시즌패스(일정기간 동안 발매된 확장팩을 구독제와 비슷한 형태로 판매하는 형태)의 도입도 염두 해 두고 있습니다.

- Editor

소프트웨어 개발 직군이 아닌 레벨디자인을 하는 파트에서도 참여할 수 있도록 에디터를 개발중이며 Note에 관한 정보를 쓰고 불러올 수 있습니다. 데이터는 JSON형태로 저장 중이며

다음의 정보를 포함합니다.

```

"songSample": 44100,
"bpm": 162 ,
"name": "test",
"startTime": 59886,
"notes": [
  {
    "E": 1,
    "T": 0,
    "S": 4
  },
  {
    "E": 2,
    "T": 0,
    "S": 4
  },
  {
    "E": 3,
    "T": 0,
    "S": 4
  },
  {
    "E": 4,
    "T": 0,
    "S": 4
  },
  {
    "E": 1,
    "T": 0,
    "S": 4
  }
]

```

(자료16)노트 데이터파일

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

-스테이지 데이터-

[Stage]

Chapter - int 챕터정보

Stage - int 스테이지정보

>게임파트에서만 활용

Clear - int (0플레이안함 / 1스테이지를 클리어 못함 / 2 클리어함 / 3Full Combo / 4 Perfect)

Score - int 플레이를 했을 시 점수저장

= = = 위 내용은 플레이 화면에 들어가지 않음 = = =

-곡정보-

[Song]

Chapter - str

Stage - str

SongName - str

Artist - str

마디 - int

BPM - int(변속에 따라 float 처리 가능)

Rank - int (0쉬움 / 1보통 / 2어려움)

>인게임 내에서 보이지 않을 수 있기 때문에(스테이지 선택 단계에서만 보일 수 있음)

위의 것들은 무시하고 일단 노트데이터만 활용한다는 가정하에 인게임 플레이를 만들어 주시면 됩니다.

-게임 정보-

[Note]

[(박자 - int, 노트타임 - int, 이벤트, offset(초) - float)] 형태

\*offset은 정확하게 떨어지지 않는 박자를 위한 값입니다. 단위는 초입니다.

## (자료17)데이터 정리 텍스트 파일


ETS는 각각 Enemy Trap Stage로 몬스터, 함정, 지형을 의미하며 생성시의 타입을 나타냅니다. StartTime은 곡마다 재생이 시작되는 부분이 다르기 때문에 시작되는 부분이며 곡의 TimeSample에 맞는 재생을 위해 샘플 레이트를 따로 저장합니다.(기본 44100)

## 2.4 수행내용

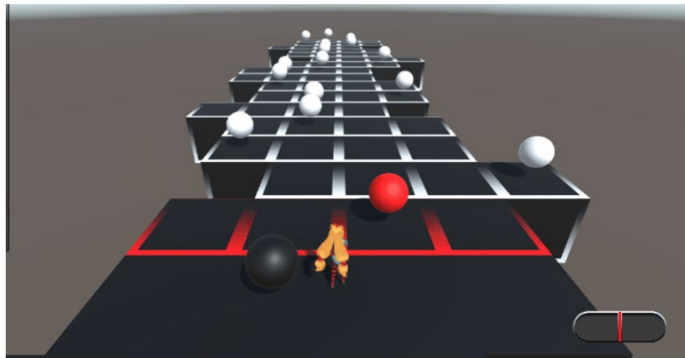
현재 계획 중 2번째 단계인 프로토타입의 완성을 진행중에 있습니다.

수행내용을 3개의 파트별로 나열하고자 합니다.

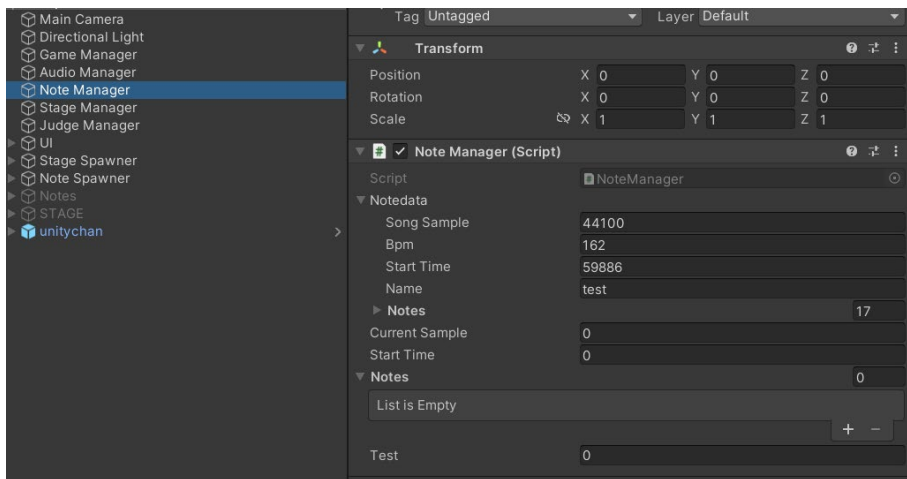


	중간보고서		
	프로젝트 명	리듬액션 게임 제작	
	팀 명	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 서정욱 : In Game 인게임내의 스테이지 구성 및 작동




(자료18)스테이지 구성



(자료19)여러 파라미터를 변경가능한 Inspector

Scene에서 보이는 인 게임 스테이지 화면을 개발 중에 있으며 직접 Stage 구성을 맡은 만큼 Editor에서 받을 내용들과 Action파트를 결합하는 부분을 맡고 있습니다. 또한 스테이지의 여러 노트들에 나오는 데이터를 어떻게 출력할지, 그 과정에서 처리될 사운드, 시간(BPM)에 대한 처리를 맡고 있습니다. 유니티를 사용하는 만큼 강력한 직렬화 (Serialization)기능과 같이 Singleton패턴을 활용하고 있으며 인스펙터(Inspector) 창에 여러 필요정보들이 보일 수 있도록 하고 필요한 데이터들을 다른 사람들이 수정하기 쉽도록 하고 있습니다. 여러가지 Manager를 통해 각각 다른 파트에서 만든 기능, 데이터들을 처리 하도록 하고 있으며 다른 파트들과 소통이 많이 필요한 편입니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

현재 게임실행 / 오디오정보 / 노트정보 / 스테이지정보 / 판정정보를 관리하는 Manager들과 관련 기능을 개발 중에 있으며 계속 기능들을 추가하고 있습니다.

GameManager :게임의 실행과 설정

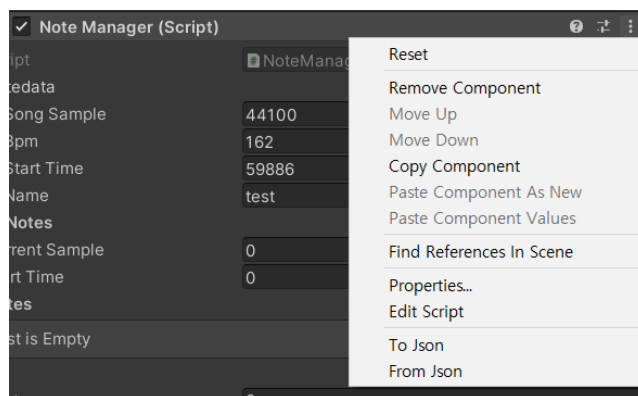
AudioManager : 음악의 재생과 Sample계산

NoteManager : 노트의 정보와 타이밍, 시간 데이터를 관리

(Editor파트와의 기능을 통합하기 위해 Inspector의 옵션(From JSON, To JSON)을 통해 간단한 입출력 또한 할 수 있습니다.)(자료20)

JudgeManager : 판정정보 관리

StageManager : 눈에 보이는 오브젝트관리



(자료20)파일 입출력 기능

NoteSpawner, StageSpawner에서는 각각 노트 지형 요소들을 출력하는 역할을 하게 됩니다. 여러 오브젝트들은 화면에 많이 표시되지 않는 만큼 생성과 파괴와 같은 비용이 큰 방법들을 쓰지 않고 미리 생성된 Pool 내에서 활성화, 비활성화를 통해 관리됩니다. (Object Pooling)

직접 계산되고 시간에 따른 판정을 구해야 하는 성능에 직접적인 영향을 주는 부분이기 때문에 그래픽에 관련된 기능을 최대한 사용하지 않고 있습니다. 현재 정확한 시간을 구하기 위해 Update에서 TimeSample을 구해 사용 중입니다.

또한 팀장으로 회의내용이나 자료준비, 게임 디자인 회의진행을 맡고 있으며 팀원과의

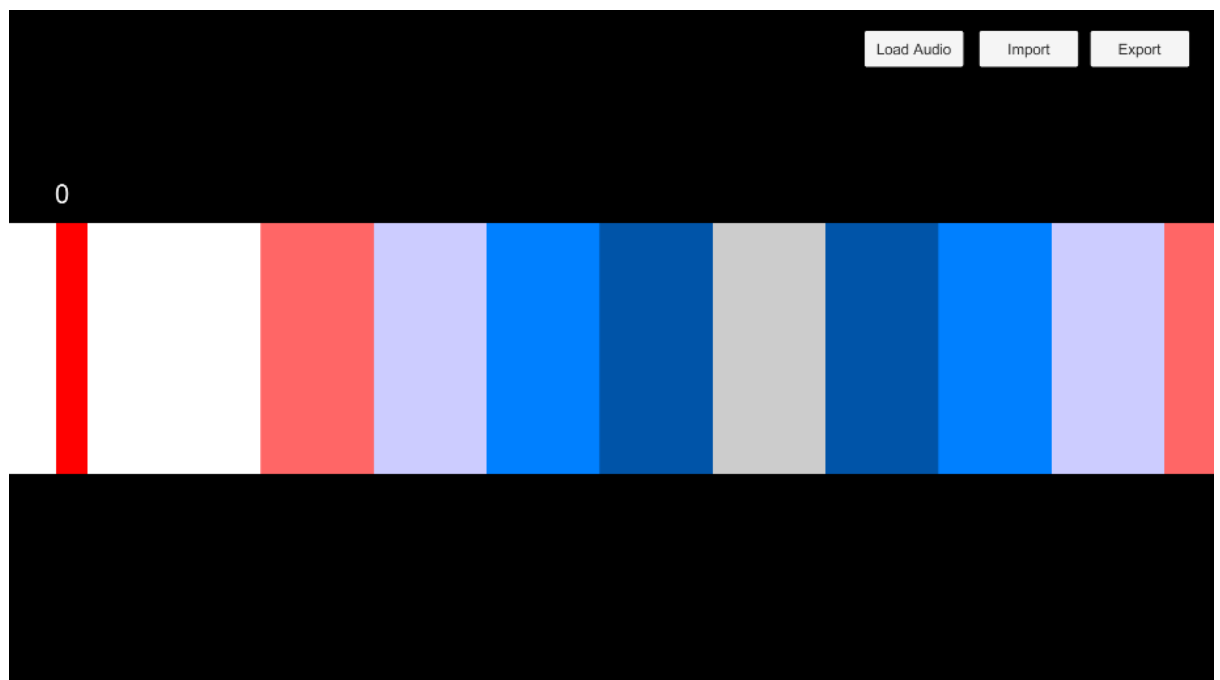
 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

소통이 중요한 게임개발인 만큼 게임의 규칙이나 자료 등을 정리하고 있습니다.

## 김민형 : Editor 제작(게임 노트에 관련된 정보)

화면 상에 어떻게 표시하는 방법을 개발했습니다. 화면의 일정 거리를 오디오의 일정 시간에 맞추는 형식으로 식들을 작성하였고, 이를 기반으로 코딩하여 에디터의 기본인 노트 체계를 만들었습니다.

- 에디터 실행화면



## (자료21)에디터 실행화면

화면상에 보이는 한 개의 노트가 가지는 너비는 이 노트가 가지는 일정 길이의 시간과 비례합니다. 이때 주어진 위치를 계산하기 위해 BPM, 박자, 인덱스, Delay가 필요하며 이는 유니티 게임 오브젝트의 스크립트 컴포넌트에서 가지고, 이를 동적으로 생성하도록 하였습니다. 이후 음악이 재생됨에 따라 처음 위치에서 재생 시간만큼 일정 거리를 이동시키도록 코딩하여 음악과 그래픽이 싱크가 맞게 플레이 되게 됩니다.

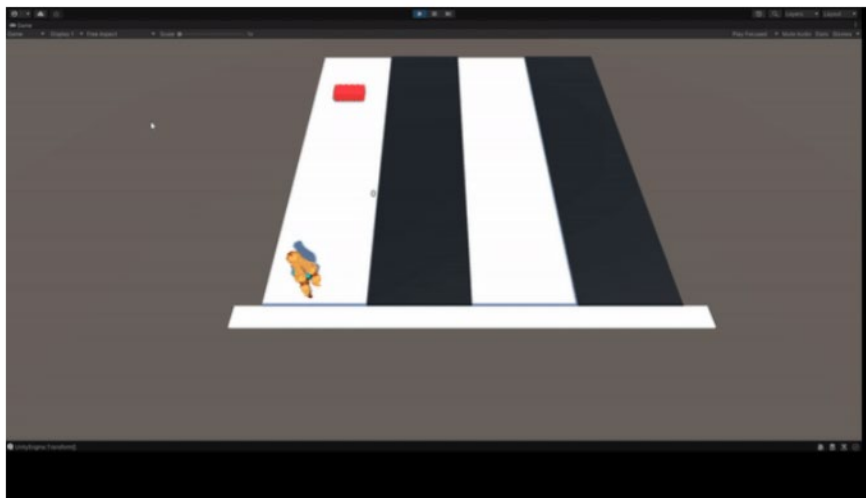
사운드 출력 오브젝트에서는 음악을 멈추고, 음악 시점을 옮길 수 있도록 키 입력을 받아 적절히 처리했고, 동적으로 생성하는 각 노트는 클릭 이벤트를 상속받아 클릭 시 노트

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

의 존재 여부를 토글 하도록 작성했습니다.

## 이승하 : Action 게임내의 그래픽(플레이어, 몬스터, 함정) 상호작용


- 캐릭터 피드백 구현화면



캐릭터의 이동과 피격 피드백

### (자료22)이동, 피격 피드백 구현화면

유니티 엔진의 Asset Store에서 플레이어 애니메이션 구현을 위해서 무료로 많이 사용되는 Unity-Chan 모델을 선택했습니다. 이동 및 점프 모션을 만들기 위해 Animator Controller를 이용해 Animator 창에서 기본적인 State Machine을 구현했습니다. 어느 상태이든 점프나, 공격키를 입력 받으면 애니메이션이 동작하도록 했으며, 걷는 모션의 방향은 D, K입력에 따른 좌측 또는 우측을 바라보도록 유니티 엔진의 Blend Tree를 이용해서 속도와, 수평 값에 따라 조절되도록 하였습니다. Blend Tree의 장점은 트랜지션과 다르게 여러 개의 애니메이션에 대해 각 애니메이션의 부분들을 다양한 비율로 나타낼 수 있는 것이며 매끄러운 애니메이션을 표현하기 위해 사용됩니다. 몬스터와 함정의 모델은 기본 3D 오브젝트인 Cube를 대신에 사용했으며, 고유의 Component와 Script를 추가했습니다. SpawnController Script에서는 랜덤하게 벽 함정 또는 몬스터가 4개의 노트 출현 장소에서 나오도록 했습니다. 노트

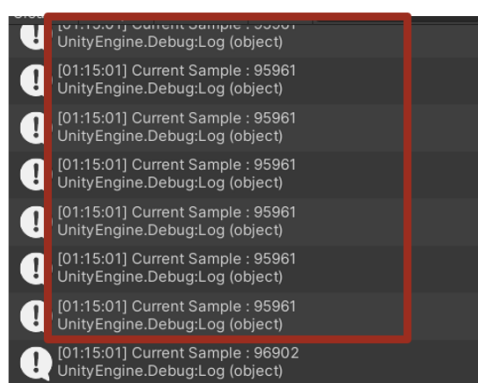
 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

가 사라지는 판정은 플레이어가 함정을 피했을 경우, 몬스터를 처리했을 경우, 모두 놓쳐서 Miss가 난 경우이며, 이때 유니티의 OnTriggerStay()와 OnTriggerEnter()를 통해 사라지도록 했습니다. WallController에서 놓친 몬스터 또는 함정이 도달해서 OnTriggerEnter()로 감지하는 즉시 Destroy()를 통해 오브젝트를 없애도록 하였습니다. PlayerController()에서는 플레이어의 이동, 점프, 공격입력 키를 통해 애니메이션을 SetTrigger()로 재생시키면서 OnTriggerStay()로 몬스터와 함정에 부여한 Tag 이름을 비교해서 함정의 경우 Spacebar를 유효 판정 범위 내에서 입력 시 없애도록, 몬스터는 공격 키를 통해 각 몬스터의 Hp가 0이 되도록 만들면 없어지도록 했습니다. 각 몬스터의 체력(Hp)에 관한 정보는 유니티의 GetComponent<>()를 통해 충돌이 일어난 오브젝트의 몬스터 Script를 가져오도록 하였으며, 공격 시 생성된 모든 몬스터가 아닌 개별의 체력이 깎이도록 하였습니다.

### 3 개발 중 개선사항 및 문제해결

#### 3.1 수정사항

서정욱 : In Game – Sound Sample 계산 문제, Sound Delay문제



(자료23)샘플 갱신 문제

여러 기능 구현중 Update() 에서 매번 불러오는 Time Sample에 큰 문제가 있었습니다. 프레임은 성능에 따라 크게 바뀔 수 있고 일정한 간격, 시간을 보장받기 어렵기 때문에 딜레이가 발생하더라도 일정한 간격을 체크할 수 있는 Time Sample을 활용하게 된 것인데 오히려 Time Sample을 사용하게 되면 프레임단위 갱신인 Update보다 갱신주기가 느린 것을 확인할 수 있었습니다. 물론 게임에서 요구하는 40ms 는 아슬아슬하게 충족하나 넉넉하지 못하며 게임 플레이 환경에 따라 정확한 시간을 계산하지 못하거나 여러 단계의 Sample을 스킵 할 가능성도 충분해 보입니다. 또한 직접 Sample 에 따라 여러 애니메이션들을 적용할 수가 없다는 문제점도 있는데 마치 스테터링(화면이 끊기는 현상)으로 보일 가능성이 높습니다. 따라서 지금은 그래픽과 타임을 완전 구분해서 일정 시간마다 Sample을 확인해 타이밍을 맞추고 있으며 만약 추후 BPM의 변화(변속)등의 기능을 구현하기 위해 연속적인 Time Sample 이 필요할 경우 갱신이 안 일어나는 구간을 구해 보간 하는 방법을 고려해 볼만 합니다.

별개로 위의 해결법은 유니티 고질적 문제인 사운드 레이턴시(음성지연)을 줄이지 못하게 되는데 해당건의 해결을 위해 Wwise 나 Fmod 와 같은 미들웨어를 이용하는 방법도 예정 중입니다.



## 김민형 : Editor 제작 – Winform Audio 문제 및 데이터 처리


프로젝트 진행 초기 라이브러리를 Winform으로 진행하고자 하였습니다. 하지만 이미지 처리 관련하여 지원이 미흡했고 외부 오디오 라이브러리인 NAudio 라이브러리는 Documentation이 최근 업데이트로 인해 없어서 개발하는데 문제가 많을 것으로 판단했습니다. 때문에 유니티 플랫폼으로 변경해 에디터를 다시 제작하게 되었습니다.

앞으로 작업할 것은 오디오 파일을 불러오고 노트 파일을 저장하고 내보낼 수 있도록 만드는 것입니다. 오디오 파일을 불러오기 위해 유니티에서 파일 입출력을 연구하고 있고, 노트 파일은 노트 오브젝트 리스트로부터 JSON 형태로 인코딩 하도록 프로그래밍 중입니다.

## 이승하 : Action – 충돌 처리 문제의 발생과 해결

충돌 관련 처리를 OnTrigger 내에서 해결하려고 했지만, Update() 간의 오차 문제로 키 입력에 따른 즉각적인 플레이어와 몬스터 간의 상호작용이 일어나지 않는 문제가 발생했습니다. 문제를 해결하기 위해 충돌 관련 처리를 최소화하기로 했습니다. 기존에는 충돌 구간에서 키 입력을 계속 감지했지만, OnTriggerEnter로 한 번의 충돌에 대한 처리로 바꿨습니다. 예시로 플레이어가 점프를 하면 애니메이션 외에 좌표도 같이 바뀌게 해서 함정이 지나가는 사이에 충돌이 일어나면 화면이 흔들리고, 무사히 통과하면 안흔들리게 했습니다. 플레이어의 몬스터에 대한 공격 판정도 따로 충돌 구간에서 키 입력을 감지하는 것이 아니라 박자에 맞게 몬스터가 움직인다는 가정하에 스테이지의 중간 지점에 도착하면 플레이어에 대한 공격 모션을 취하도록 했습니다. 그리하여 제한시간 내에 플레이어가 하단에 생성된 노트에 대해 F나 J키 입력을 통해 Miss가 안 뜨면 몬스터가 제거되면서 점수가 오르고 아닐 시 그냥 사라지도록 했습니다.

Vector3 구조체의 MoveTowards 함수를 통해 플레이어의 현재 위치에서 목표 지점으로 부드럽게 이동하도록 구현해보려고 했으나, 어떤 속도값을 넣어도 순간 이동하는 문제가 발생해서 대신에 Lerp 함수를 사용했습니다. 현재 지점과 목표지점 사이에 보간법을 이용하여 부드러운 움직임을 Time.deltaTime을 Update()에 따라 사용했습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

이렇듯 오브젝트간의 상호작용 구현은 진전이 있지만, 모델링 관련해서 저희 팀은 따로 맡아서 할 수 있는 인력이 없는 상황입니다. 만일에 Asset Store에서 구매를 해서 사용하게 될 경우 보통 상업적으로 문제되는 부분이 없지만, 민감한 주제인 만큼 자세히 알아봐야 할 것입니다. 특히 독창성 관련 문제로 게임의 스토리 방향성이 잡히면 그에 맞는 플레이어 및 몬스터, 지형지물을 만들어야 하는데 그에 알맞은 것들을 직접 찾는 것은 매우 힘들 것입니다. 그렇기 때문에 외주를 맡기는 경우를 염두해두고 있으며, 상황에 따라 유동적으로 대처하고자 합니다. 아무리 괜찮은 게임을 만들더라도 시각적으로 사용자에게 만족감을 못 줄 경우 아무 소용이 없기에 생각보다 신경을 많이 써야 할 것입니다

## 4 향후 추진계획

### 4.1 향후 계획의 세부 내용

현재 프로젝트의 진행 정도는 계획서 상의 2번에 해당하는 기본적인 기믹(플레이어 이동, 점프, 몬스터와 노트 생성, 노트 판정에 따른 몬스터 처치)을 추가해 prototype을 완성시켰습니다. 이것을 바탕으로 3,4,5번 계획(2.1.3)을 수행할 것이며, 에디터에서 만들어진 JSON 파일의 정보를 게임 파트까지 넘겨서 플레이어, 몬스터, 노트, BPM 정보에 따라 스테이지의 구성 및 플레이가 원활히 하는 것이 향후 계획입니다.

앞서 설명했듯이 모델링 관련해서 경험이 있는 인원이 없으므로 기한 내의 프로젝트 완성을 위해 스토리에 맞는 괜찮은 모델들을 구매하거나 외주를 맡겨서 원하는 것들을 사용하고자 합니다(비용적인 문제가 많이 작용할 것입니다). 스토리의 구성은 게임에서 자주 사용되는 전설, 신화들을 참고할 것입니다. 예시로 7대 죄악, 그리스 로마 신화, 북유럽 신화, 삼국지와 같은 이야기들의 큰 틀을 참고하여 만들 것이며, 그에 따라 서사 구조에 맞게 음악 선정을 하고 스테이지 제작을 할 것입니다.

그리하여 PC환경에서의 게임이 완성이 되면 다음으로 모바일 환경을 준비하는데, 알아본 결과 유니티 엔진에서 모바일 환경 개발로 설정해서 키 입력에 따른 플레이를 터치 형식으로만 바뀌도 된다고 합니다. PC와 모바일 환경에서의 게임이 완성이 되면 최종 발표일까지 5번 계획인 딜레이 문제 및 각종 일어날 수 있는 버그, 최적화 문제등을 해결하는 것이 주요 관건이 될 것입니다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	리듬액션 게임 제작	
	<b>팀 명</b>	Tema 45	
	Confidential Restricted	Version 1.5	2022-04-4

## 5 고충 및 건의사항