

2021-2-OSSP

프로젝트 최종보고서

<T-REX-Breeder팀>

2017111599 경영정보학과 문혜리

2018112449 산업시스템공학과 이보경

2017112519 산업시스템공학과 조남현

목차

1. 프로젝트 소개

1.1 프로젝트 주제

1.2 프로젝트 개선 제안사항

2. 프로젝트 결과

2.1 UI 개선

- 메인 배경화면 및 버튼 이미지 변경

2.2 인게임 개선

- 모드 수정 및 추가
- 타이머 기능 추가
- 플레이어 및 보스의 HP bar 수정/추가
- 장애물 별 데미지 다양화
- 스테이지 별 보스 다양화

2.3 코드 구조 개선

2.4 기타 버그 개선

3. 협업 과정

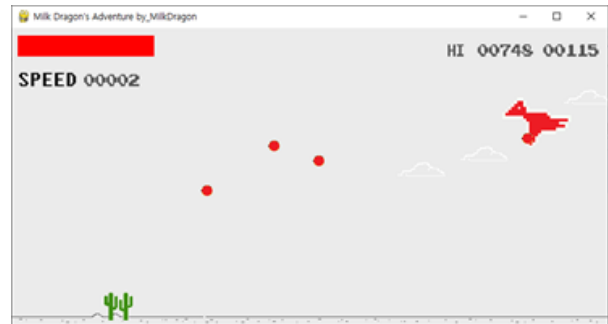
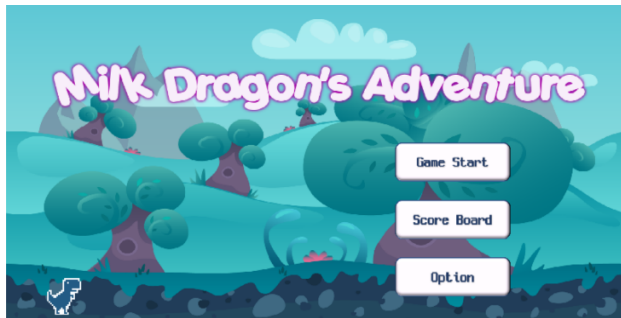
3.1 최종 타임라인

3.2 git 사용

1. 프로젝트 개요

1.1 프로젝트 소개

본 프로젝트의 목표는 오픈 소스를 이용하여 T-Rex Rush 게임을 개선하는 것이다. 2021-1-OSSPC-MilkDragon-6팀의 Milk Dragon's Adventure를 기반으로 멀티 플레이 기능을 추가하여 플레이 요소를 다채롭게 한다. 모드 별 튜토리얼을 통해 게임에 대해 쉽게 이해하고, 기존 코드에서 발생하는 버그와 UI를 개선하여 완성도 높은 T-Rex 게임을 개발한다. 이를 exe파일로 생성하여 게임실행에 대한 접근성도 높이하고자 한다.



License	MIT	OS	Windows, Ubuntu
Language	Python	Module	Pygame
Github	github.com/CSID-DGU/2021-1-OSSPC-MilkDragon-6		

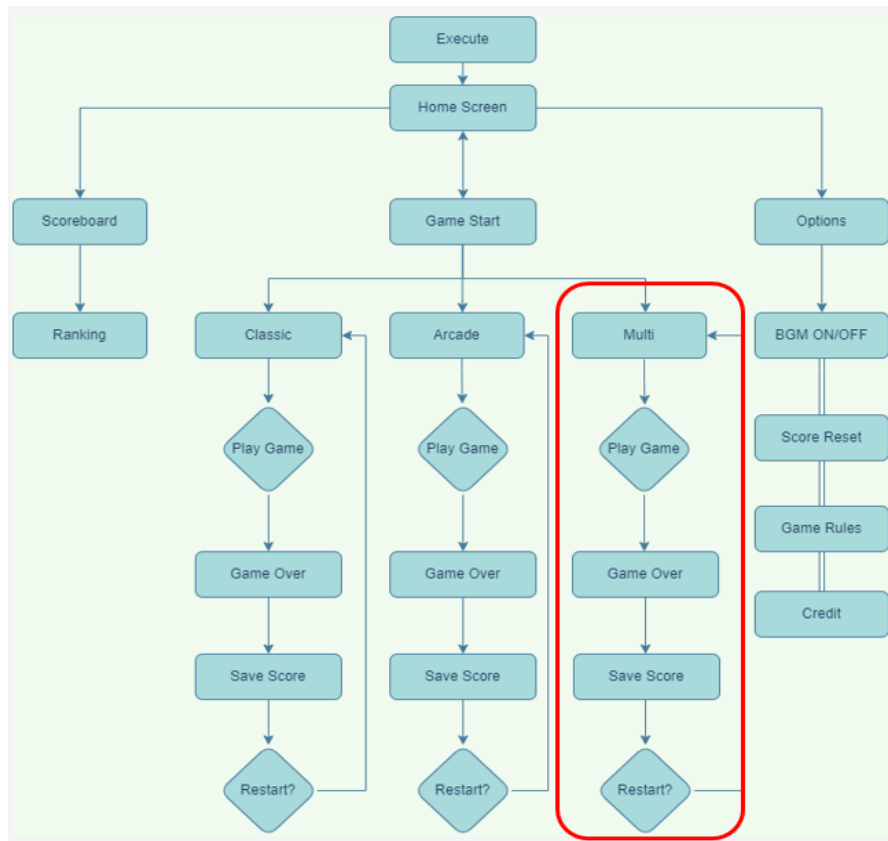
Strength	Weakness
<ul style="list-style-type: none"> - 게임의 구조가 간단하여 환경의 제약이 적음 - 기능적으로 코드가 잘 구분되어 있음 - pygame으로 작성됨 	<ul style="list-style-type: none"> - 모드별 게임 플레이 방법 설명이 없음 - 화면 밖으로 숨어있으면 죽지 않는 버그 존재 - 아이템이 잘 나오지 않고 보스의 체력이 높아 난이도가 너무 높음
Opportunity	Treat
장애물과 아이템에 대한 기본적인 기능이 구현되어 있어 새로운 테마와 아이템을 구현하기에 수월함	<ul style="list-style-type: none"> - 멀티 플레이에 대한 구현이 전혀 없음 - 다른 고전게임에 비해 레퍼런스가 많이 없음

1.2 프로젝트 개선 제안사항

UI/UX 개선	인게임 개선
<ul style="list-style-type: none"> - 메인 및 메뉴 화면 변경 - 게임배경 및 튜토리얼 변경 - 게임메뉴에서 홈 화면으로 돌아가기(추가) 	<ul style="list-style-type: none"> - 스테이지 및 타이머 기능 추가 - 공룡 hp bar 수정 - 보스 hp bar 시각화 - 속도 감속 기능 추가 - 보스 공격 패턴 변경 - 서버 구축 및 멀티모드 구현 -> 로컬 멀티모드
버그 개선	코드 구조화(추가)
<ul style="list-style-type: none"> - 화면 밖으로 나가면 죽지 않는 버그 수정 - GUI 버튼이 잘 눌러지지 않는 버그 수정 	<ul style="list-style-type: none"> - 유사한 기능을 묶고 정리하는 코드 클리닝 작업

2. 프로젝트 결과

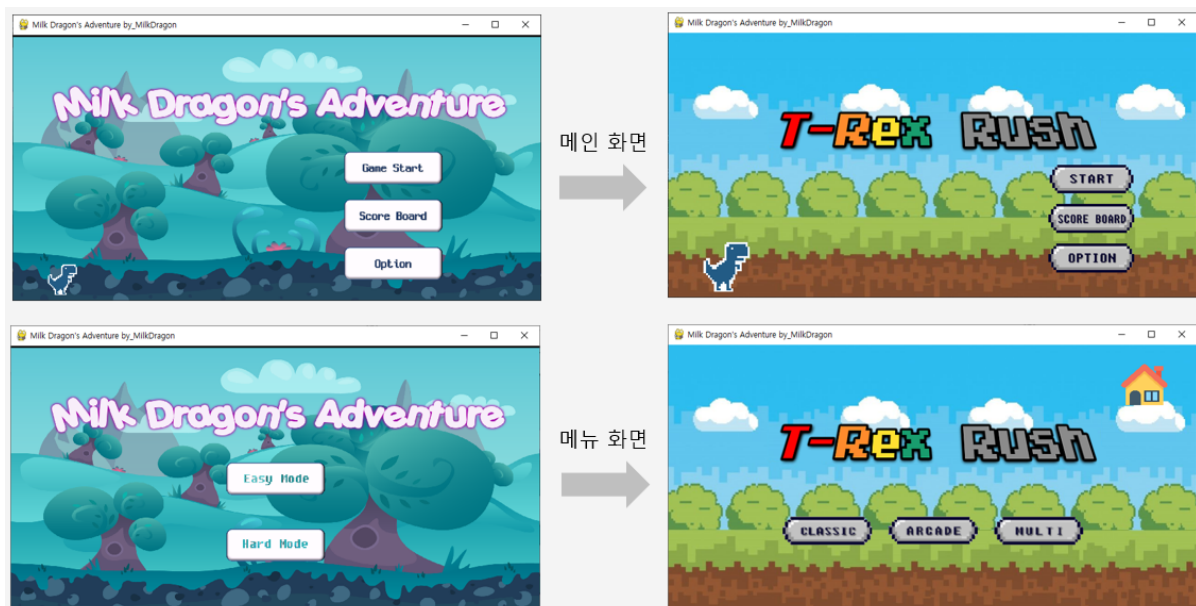
2.1 최종 게임 흐름도



▲ 최종 게임 흐름도 - 추가된 게임 흐름을 빨간색으로 표시

2.2 UI/UX 개선

○ 메인 배경화면 및 버튼 이미지 변경



- 이미지 출처 : freepik, Vintage vector created by stockgiu - www.freepik.com

- freepik의 이미지를 사용할 때 주의 사항 : 무료이용자는 사용처별 출처기재 방법을 따라 이미지 출처를 남기면 무료로 사용할 수 있음. 이미지를 수정 및 포함해 상업적 활동도 가능하지만 이미지만을 이용한 재배포, 재판매 등은 제한됨

Apps and games

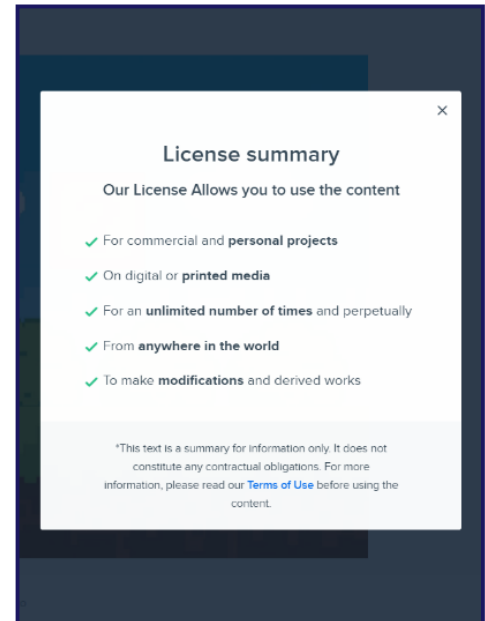
You can use our images as main or secondary element to create the interface of your application or game, as long as the service offered by your app does not involve the redistribution of our images.

For example, you can create an app for the users to check the weather, to buy tickets for a concert, or any other third party activity, and also a game where the background and some of the elements are resources from Freepik.

Please be aware that **this is not applicable if you want to use our images to create image filters or stickers**. In that case, you need to make sure that **our images are not the main element** of the filter or the stickers.

If you download Freepik resources as a free user **you must include the attribution line** "Designed by Freepik" on the app credits page and on the description in the app store. If you don't want to include this attribution line, you must **purchase any of Freepik subscriptions**.

Please remember that you are not authorized in any case to make business or deliver the original (or modified) editable AI/PSD/EPS file to anyone, sub-license, resell or rent any of our resources.



▲ freepik의 이미지를 app과 game에 사용할 때 주의 사항

▲ freepik license 안내

- 버튼 출처 : 미리캔버스, <https://www.miricanvas.com/>

- 미리캔버스의 이미지는 출처표시가 강요되지 않음

누가 사용할 수 있나요?
개인/기업/공공기관/비영리단체/동호회/소모임 등 **누구나 사용**하실 수 있어요!

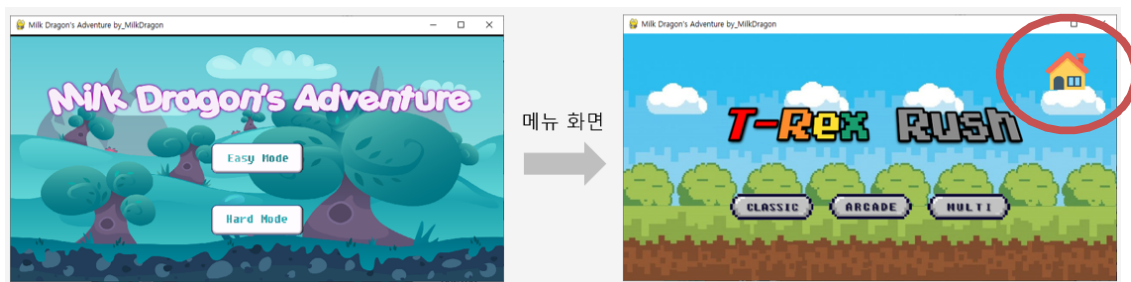
어떤 용도로 사용할 수 있나요?
상업적(홍보 및 판매 등) / 비상업적 **용도 상관없이 모두 사용**하실 수 있어요!
*잠깐! 사용목적이 **상표권 등록**을 위한 **로고 제작**이거나 **공모전 출품**이라면 **여기**를 확인해주세요.*

어디서 사용할 수 있나요?
온라인(홈페이지, 쇼핑몰, SNS 등) / 오프라인(포스터, 전단지, 스티커 등) **구애없이 어디서나 사용**하실 수 있어요!

단, 아래 두 가지는 꼭 지켜주셔야 해요!
하나, **미리캔버스에서 제공하는 템플릿 및 디자인 요소(선, 도형, 아이콘, 일러스트, 사진 등)은 반드시 미리캔버스 에디터로 편집하셔야 합니다.** 저작권사와의 라이선스 미리캔버스에서 제공하는 템플릿과 디자인 요소를 가지고 다른 편집물(포토샵, 일러스트 등)로 편집하시면 저작권 문제가 발생할 수 있습니다.
둘, 개별 디자인 요소(선, 도형, 아이콘, 일러스트, 사진 등)에 대한 저작권은 미리캔버스 혹은 미리캔버스와 계약된 업체가 가지고 있습니다. 따라서 하나의 디자인 요소만 다운로드하여 사용하는 것은 불가합니다. **반드시 2개 이상의 디자인 요소를 사용해 디자인하셔야 하는데, 이미 여러가지 디자인 요소가 결합되어 있는 템플릿을 편집해서 디자인하시면 간단합니다.**

▲ 미리 캔버스 이미지 사용 범위 안내

○ 홈 버튼 추가



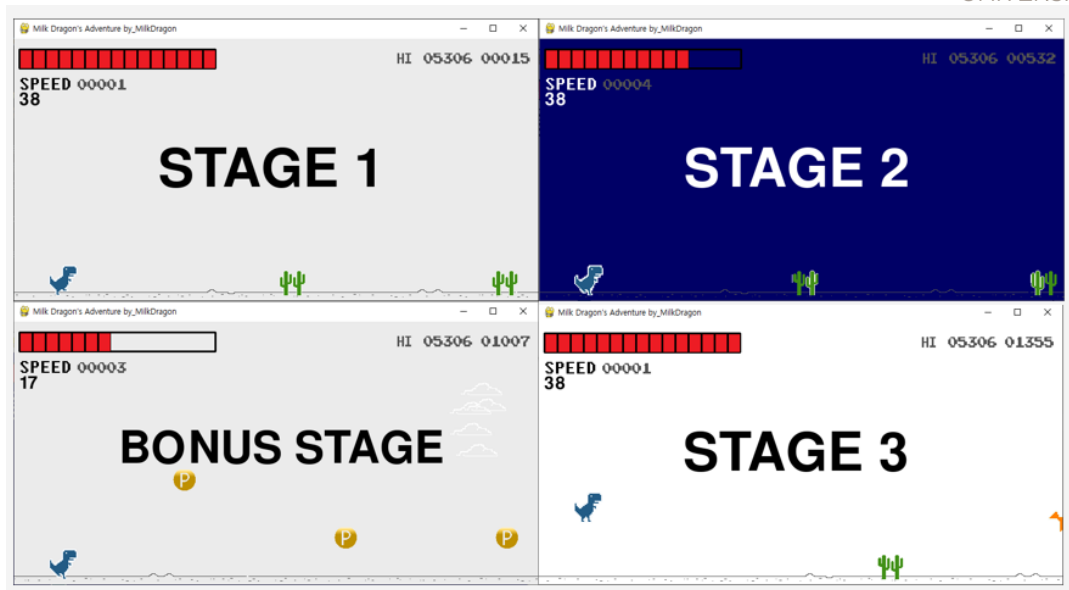
▲ 메뉴화면에서도 홈으로 돌아갈 수 있게 버튼을 만들

2.2 인게임 개선

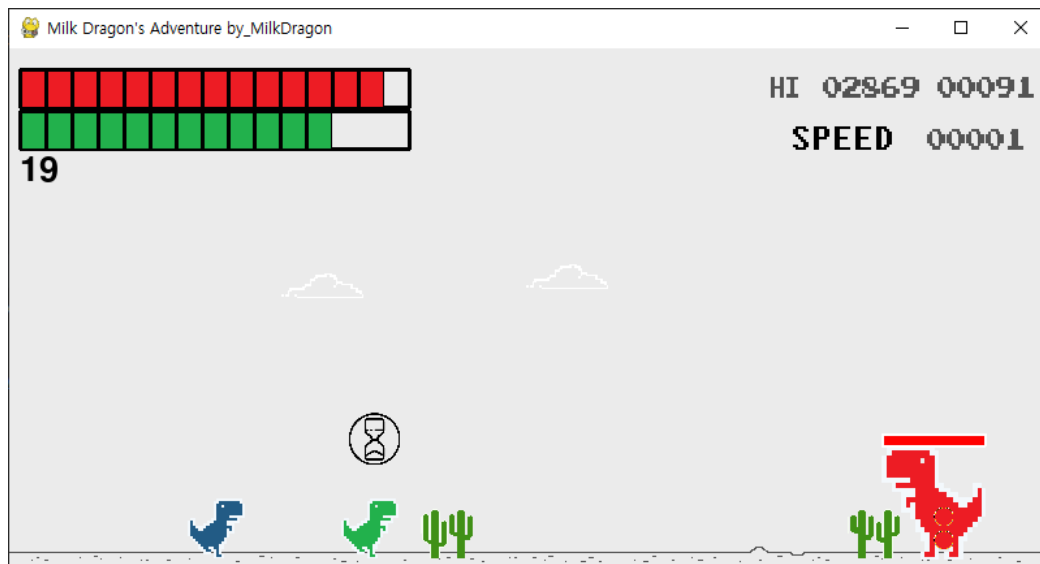
○ 모드 수정 및 추가

기존	
Easy Mode	<ul style="list-style-type: none"> - 장애물만 존재 - 점프/웅크리기만 가능 - 끝이 없이 무한히 펼쳐짐
Hard Mode	<ul style="list-style-type: none"> - 장애물+보스몬스터, 아이템 존재 - 상하좌우 이동 가능 - 한 번 보스몬스터를 물리치면 - Easy Mode와 같이 무한히 펼쳐짐

변경	
Classic Mode	<ul style="list-style-type: none"> - 기존 Easy Mode와 동일
Arcade Mode	<ul style="list-style-type: none"> - 스테이지 1, 2, 3 + Bonus Stage 구현 - 각 스테이지의 보스는 난이도에 맞게 구현 - Bonus Stage는 스테이지 2 후에 등장하며, 목숨을 채워주는 life 아이템 등장
Multi Mode	<ul style="list-style-type: none"> - 같은 컴퓨터에서 2명이 플레이하는 Arcade Mode - 한 명은 방향키, 한 명은 WASD



▲ Arcade Mode/Multi Mode의 스테이지 구현: stage 1 -> 2 -> Bonus -> 3



▲ multi Mode에서 두 명의 로컬 플레이어

○ 타이머 기능 추가

- Arcade Mode에 타이머를 추가하여 정해진 시간에 보스 등장, 정해진 시간 안에 보스를 죽이지 못하면 GameOver



▲ 타이머추가

○ 플레이어 및 보스의 HP bar 수정/추가

- 목숨이 몇 개 남았는지 알 수 없었던 기존 연속된 bar형태의 체력 바에 grid를 추가



▲ 플레이어 hp bar 변경







- 보스몬스터에도 편의를 위해 체력 바 추가



▲ 보스 체력 가시

○ 장애물 별 데미지 다양화

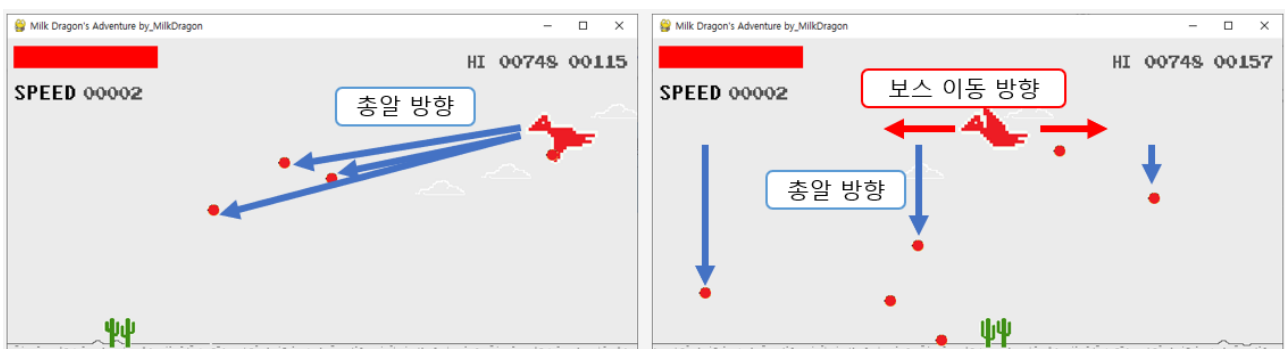
- 게임 속도가 증가하기만 해서 플레이의 난이도가 너무 높아지는 문제가 있었음 장애물을 맞으면 목숨이 깎일 때 스피드도 줄여줌 플레이어는 속도를 조절하면서 전략적인 플레이가 가능해 짐

	목숨-1			목숨-1
	목숨-1			목숨-1, 스피드-1
	목숨-1			목숨-2, 스피드-1

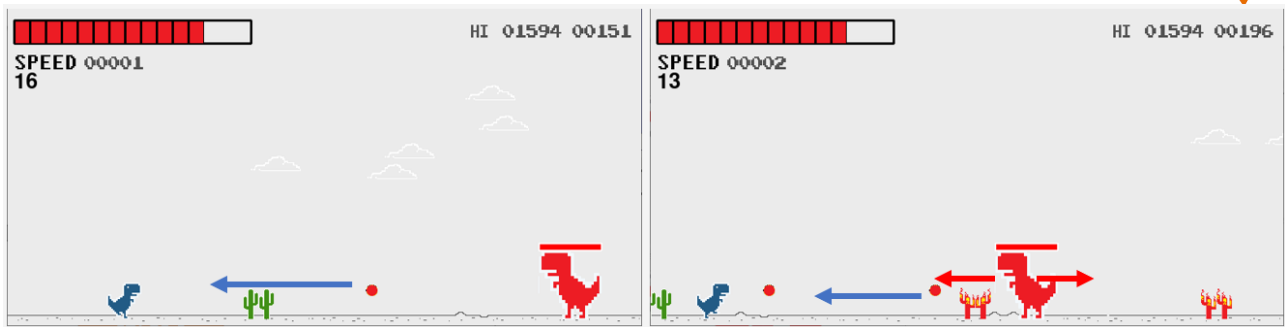
▲ 장애물별 목숨 및 속도 데미지 차별화

○ 스테이지 별 보스 다양화

- 기존에 보스가 등장하는 모드에서는 보스가 한 번만 등장함 스테이지를 추가한 이후에 스테이지마다 보스의 형태 및 공격패턴을 다양하게 하면서 난이도와 재미를 더함

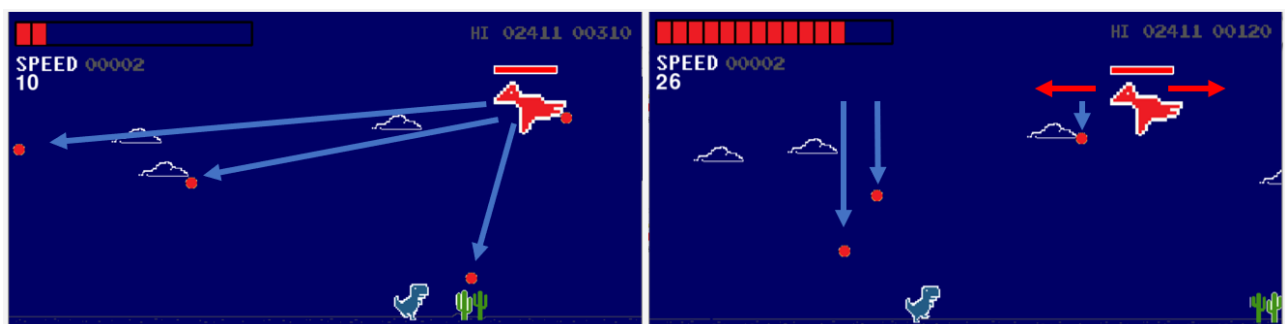


▲ 기존 보스 이미지와 공격 패턴



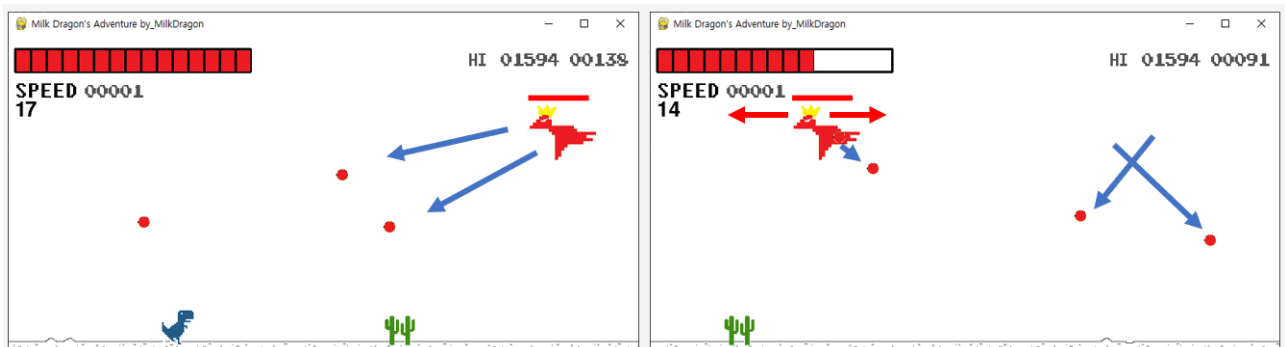
▲ stage 1 보스 이미지와 보스 공격 패턴

stage 1에서는 플레이어보다 훨씬 크고, 바닥에서 좌우로 움직이는 보스공룡이 등장함 총알은 앞으로만 쏘며, 총알 속도는 랜덤임



▲ stage 2 보스 이미지와 보스 공격 패턴

stage 2에서는 기존과 동일한 하늘을 나는 익룡보스가 등장함

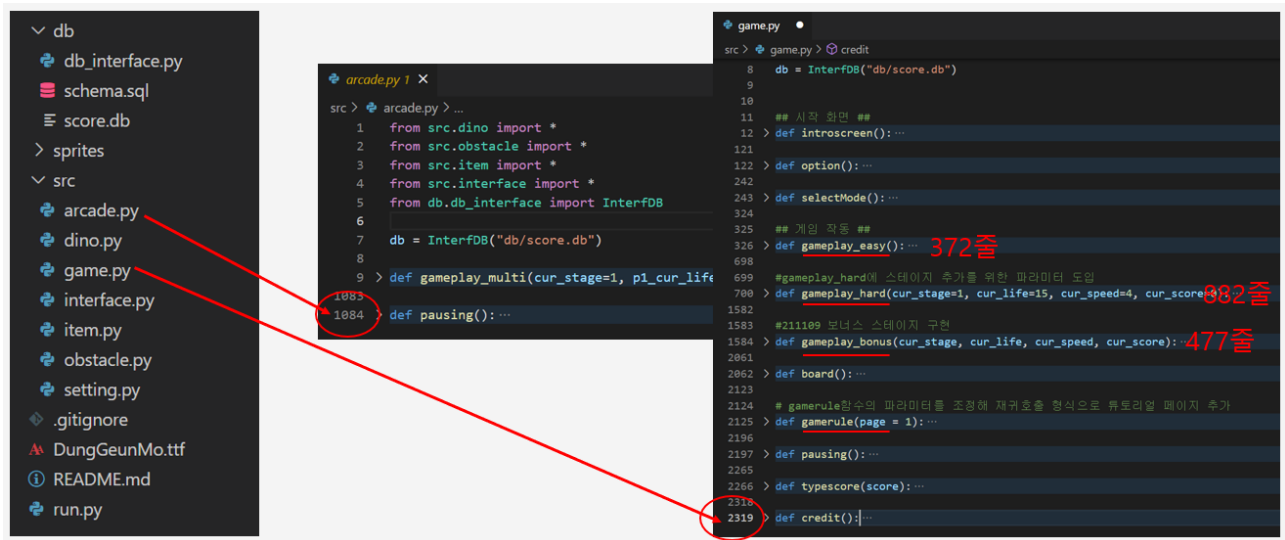


▲ stage 3 보스 이미지와 보스 공격 패턴

stage 3에서는 익룡보스의 공격이 업그레이드 됨 좌우로 움직일때도 랜덤 방향과 속도로 공격함

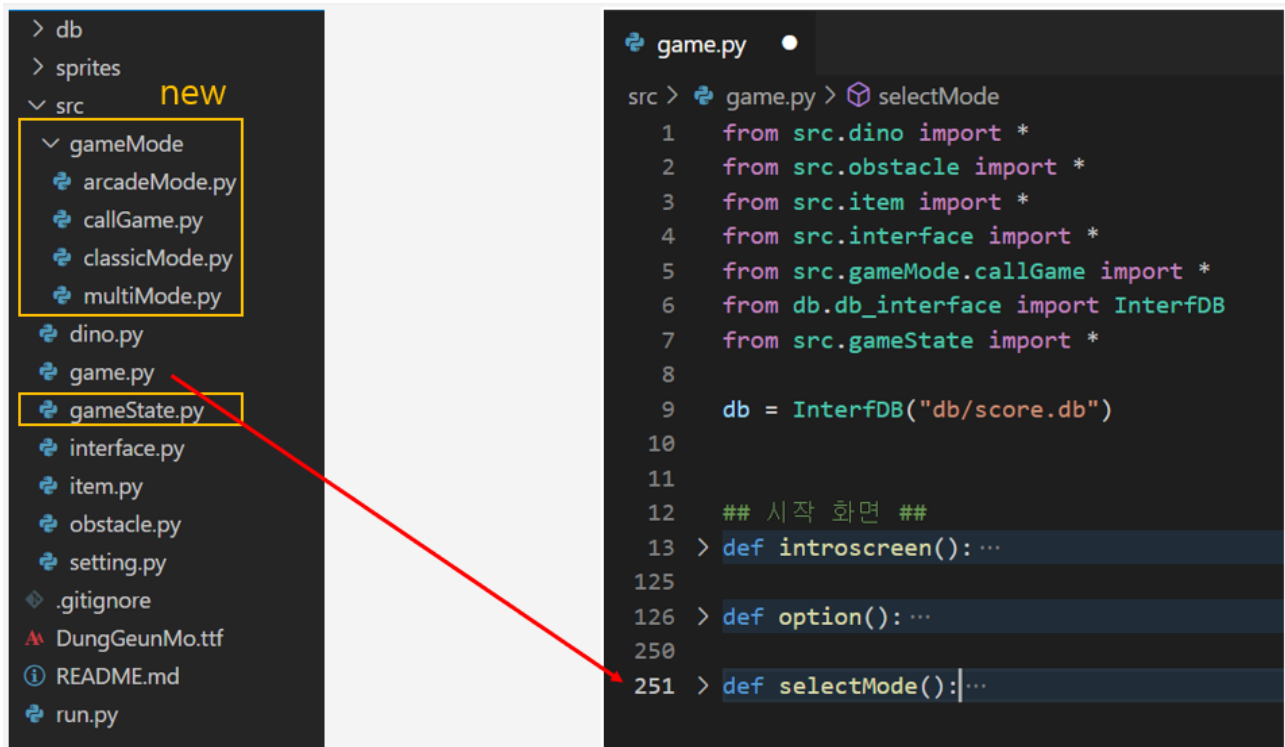
2.3 코드 구조 개선

- 스테이지와 모드가 추가되었기에 보다 효율적인 코드를 위해 유사한 코드끼리 묶어 구조화 진행

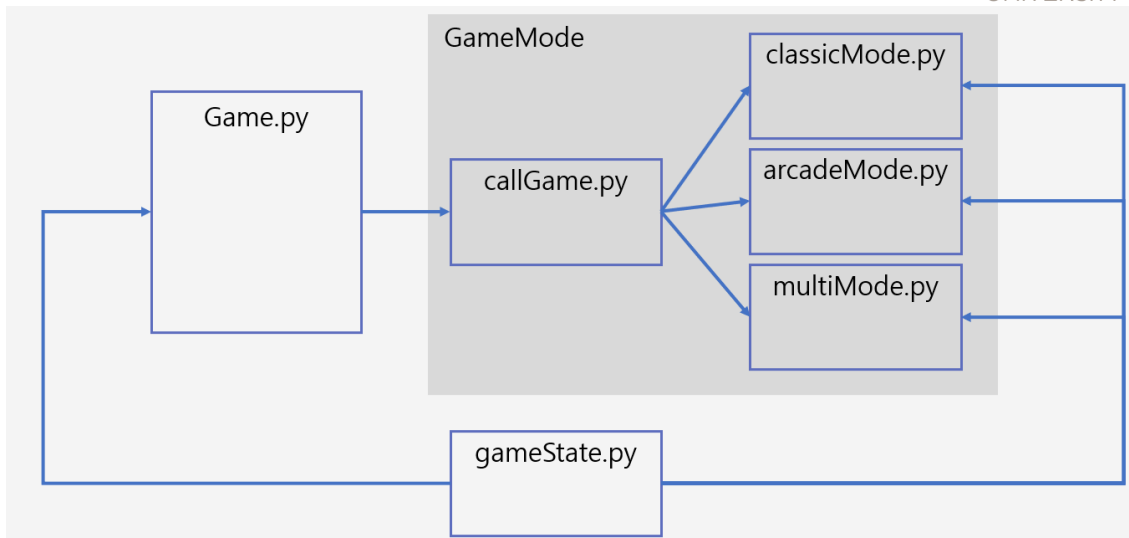


▲ 프로젝트 개선 후 처음코드

원시 코드의 game.py에는 easyMode(), hardMode(), introscreen() 등 게임과 관련된 함수가 모두 여기에 구현되어 있음 프로젝트를 진행하면서 여기에 그대로 stage 도입 등의 기능을 추가하면서 이 파일에만 2300줄이 넘어 가독성이 매우 떨어지는 상태임



▲ 새로운 코드 파일 구조 및 game함수 코드 정리



▲ 새로운 코드 구조의 관계도

▲ gameMode와 gameState의 추가

- 플레이어 객체를 게임 함수 밖에서 생성해주며 넘기도록 하여 각 함수마다 공통객체를 매번 생성해줄 필요가 없도록 함 이 과정에서 기존 멀티 모드에서 “isDead” parameter로 공통의 생사를 판단하고 스테이지 별로 넘겨준 과정을 -> 플레이어 객체를 바로 넘겨서 파라미터를 추가하지 않고도 해당 상태를 인식할 수 있게 함
- DB와 gamestate, gamemode에서 사용하는 공유가 필요한 대부분의 변수들을 함수마다 return 해서 받아오도록 구현함
- 기능들을 파일 별로 분리함 (특히 introscreen으로 돌아가는 함수가 거의 모든 중요 파일들에서 다 쓰이는데 import로 해결하지 못해서 마련한 대책)
- CallGame이라는 핸들러 파일을 만들어 game과 gamemode의 circular import문제를 해결함 dictionary를 통해 각각의 mode를 불러와서 기존의 코드보다 훨씬 가독성이 좋아짐
- dino객체 등 게임모드마다 생성되어야 하는 함수는 최대한callgame핸들러에서 한꺼번에 부르도록 함

2.4 기타 버그 개선

○ 화면 밖으로 나가져 숨을 수 있는 버그 수정

원인: 좌우 방향키를 움직일 때 최소/최대값을 설정하지 않았음

해결: 좌우 방향키에서 최소값과 최대값을 설정해 줌

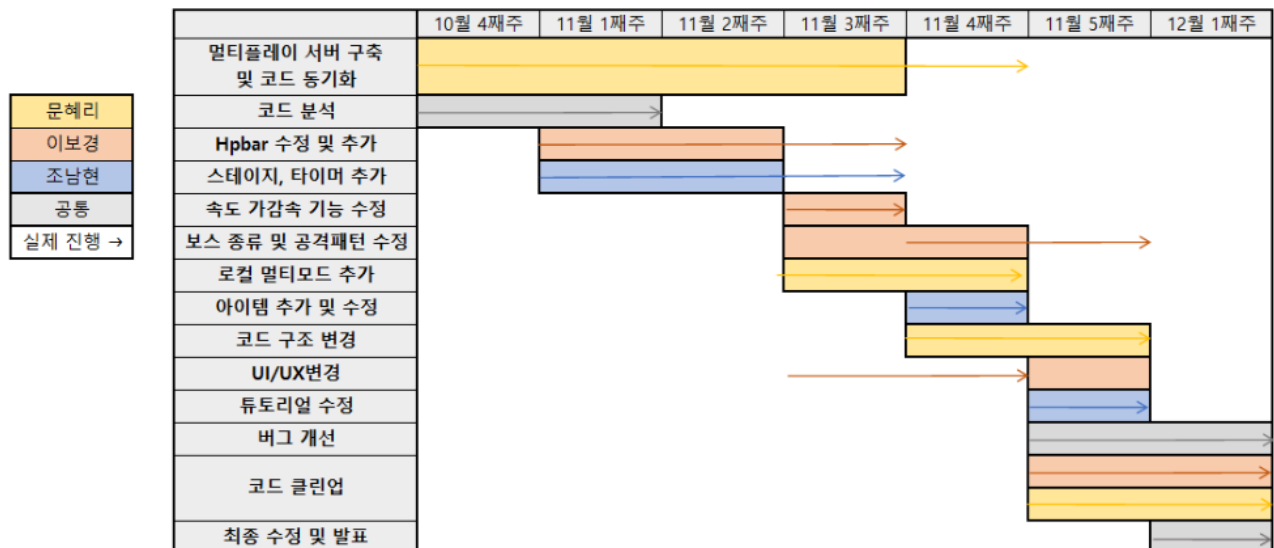
○ GUI 버튼이 잘 눌러지지 않는 버그 개선

원인: 마우스 클릭을 인지하는 공간과 버튼의 위치와 맞지 않았음

해결: UI를 개선하면서 새로운 버튼의 위치와 버튼 클릭을 인지하는 공간을 맞춰 줌

3. 협업 과정

3.1 최종 타임라인

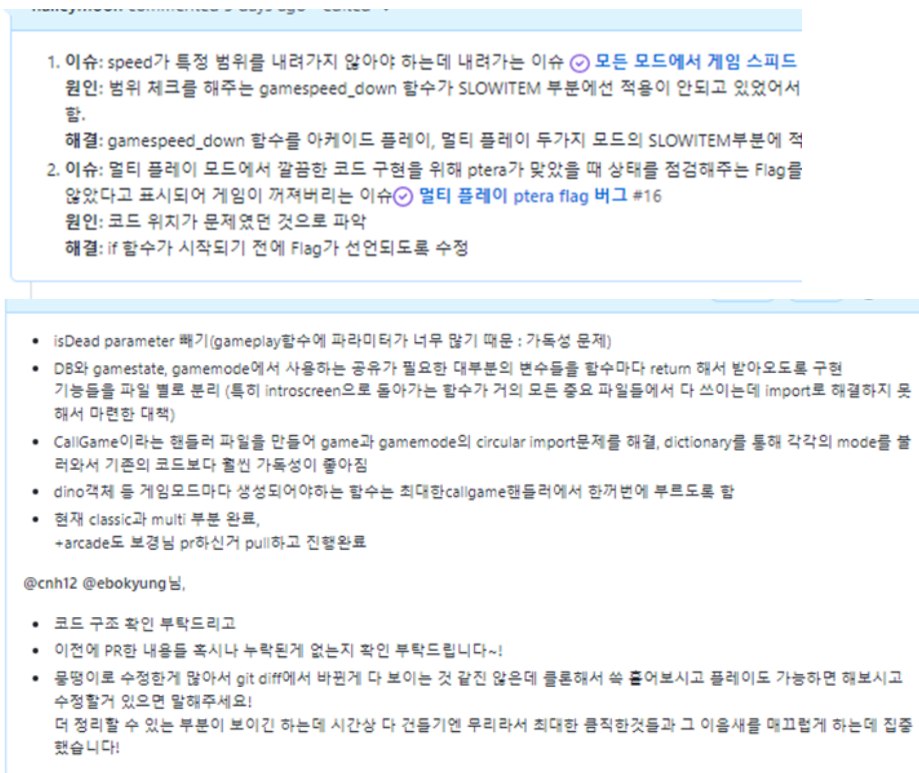


3.2 git 사용

- 협업 툴로 git 사용, Pull Request 전에 모든 팀원의 코드리뷰를 거쳐 확인 후 Merge 진행
- PR 총 37번, 이슈 총 13개, 기록된 Conversation과 코멘트 총 39개, 이외에도 카톡, 화상회의와 화면 공유, 대면 회의등 여러번 만나 진행상황을 공유했고, 피드백이 원활하게 오고가며 활발한 소통 속 협업이 원활하게 이루어졌음
- PR을 날릴때에는 최대한 자세하게 왜 해당 PR을 날리게 되었는지, 어떤 부분이 수정되었는지, 어떤 우려점이 있는지 등의 내용공유를 했고,

- 이슈를 해결한 경우의 PR은 이슈를 PR에 링크하여 해당 이슈의 원인은 무엇이였는지, 어떤 방법으로 해결했는지를 기록했음

PR관련 첨부 사진:

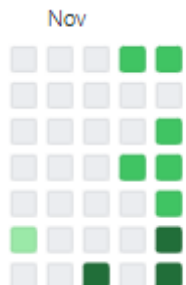


○ 잔디밭

- 문혜리



- 이보경



- 조남현



팀원 중 문혜리는, 서버 구축 및 동기화에 시간을 많이 투자하며 해당 기간에는 로컬에서 테스트하는 기간이라 잔디밭을 심을 수 없는 상황이었지만 코드리뷰와 코멘트, 이슈 등 많은 기여를 하였으며 이슈와 코드리뷰 숫자가 이를 증명해줌

또 팀원 중 이보경은, 우분투에서 깃 작업을 하며 계정 연동이 잘못되어 커밋하고 푸쉬까지 정상적으로 했음에도 불구하고 잔디밭이 심어지지 않는 문제가 있어 기록이 잘 되지않았음 하지만 꾸준한 커밋 기록은 남아있기 때문에 증빙자료로 아래 사진을 첨부합니다.

아래는 잔디밭이 심어지지 않는 이유 증빙:

위 두개의 이미지와 같은 문제가 반복스럽게 발생했다. 알 뵈닌 녀석이 거밋 이력을 무시해도 쫓미시 않은 것이나. (특정 Repository 에 따라서 안 되는 문제) 구글신에게 물어보니 **user.name**과 **user.email** 설정이 계정 정보와 일치하지 않아 그렇다는 내용이 많다.

