

발전을 목표로 노력하는 개발자
김웅섭

소개 및 목차

프로필

김웅섭

2006. 03. 27

010-4060-1993

이력

2023 지방 기능 경기대회 출전

2024 지방 기능 경기대회 출전(5위)

2022 교내 여름방학 게임잼 출전

2023 교내 여름방학 게임잼 출전

2024 교과 우수상(게임 프로그래밍)

GTQ Excel 자격증

1. 디지텍 컴퍼니
 2. 슬라임 디펜스
 3. 2023 전국 기능대회 연습작
 4. 2024 지방 기능대회 연습작
 5. 몬스터 스크램블
 6. 핀볼 좀비
 7. Escape From
 8. 길찾기 알고리즘 연구
-

Digitech Company

#출시 목적 #디펜스 #슬라임 #모바일

플랫폼 : 모바일

사용 언어 : C#

개발 인원 : 프로그래밍 2

그래픽 2

기획 1

개발 기간 : 2개월

역할 : 메인 프로그래밍

● 게임설명

- 졸업작품으로 발표하기 위해 만드는 프로젝트
- 수익을 달성하기 위하여 위험한 던전에서 물건을 줌의 게임

● 구현 기술

- 아이템 아이콘을 쉽게 제작하기 위하여 아이템 아이콘을 자동으로 제작하는 도구를 구현.
- UniRx에서 IEquatable를 호출할 때 boxing이 생기는 것을 막기 위하여 struct에 IEquatable을 override 함.
- 게임 중도 입장을 구현하기 위하여 Request -> Response 관계 구조를 짜서 서버를 설계해 멀티플레이를 구현.
- 아이템의 애니메이션을 유연하게 적용하기 위하여 IK를 사용하는 애니메이션 구조를 설계.



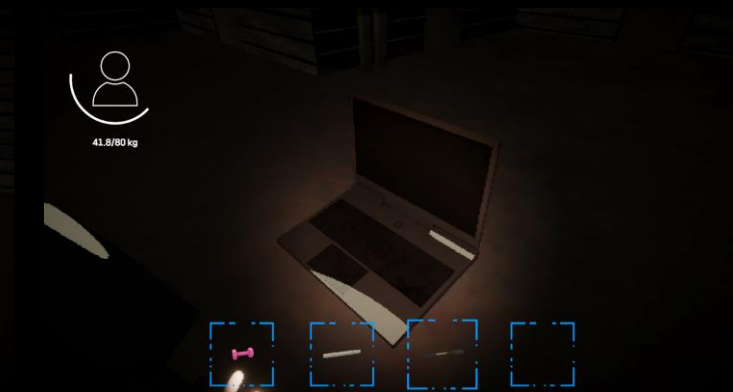
기지 안쪽



아이템 선택



인게임 1



인게임 2

Github : https://github.com/miro0325/Digitech_Company

Slime Defense

#출시 목적 #디펜스 #슬라임 #모바일

플 랫 폼 : 모바일

사용 언어 : C#

개발 인원 : 프로그래밍 2

기획 1

개발 기간 : 2개월

역 할 : 메인 프로그래밍

● 게임 설명

- 구글 플레이 스토어 출시를 목적으로 만들던 디펜스 게임
- 모든 라운드 클리어 또는 최대한 오래 살아남는 것이 목표
- 아군 슬라임은 동일 레벨, 타입의 다른 슬라임과 융합
- 증강체, 장애물 제거 등의 전략적 선택으로 스테이지를 클리어

● 구현 기술

- Batching을 줄이기 위하여 Occlusion Culling등의 최적화 기술을 적용.
- 맵의 그리드를 더욱 쉽게 제작하기 위하여 맵의 정보를 받아와 그리드를 제작할 수 있는 도구를 제작.
- 슬라임들의 스킬을 쉽게 교체하기 위하여 Strategy Pattern과 Activator를 이용하여 동적으로 스킬을 넣을 수 있게 구현.
- 슬라임의 밸런싱을 쉽게 하기 위하여 Observer Pattern 을 이용한 스탯 시스템을 설계.
- Dictionary를 사용한 시스템보다 Enum을 사용하여 Array에 접근하는 방식이 더 성능이 좋은 것을 알아내어 기존 스탯 시스템을 보완하였다.



인 게임



스테이지 선택



슬라임 디테일



증강체 선택

Github : <https://github.com/Proffeine0327/slime-defense>

2023 전국 기능대회

#기능대회 #타워디펜스

플 랫 폼 : PC

사용 언어 : C#

개발 인원 : 프로그래밍 1

개발 기간 : 3일

역 할 : 메인 프로그래밍

● 게임 설명

- 전국 기능경기대회 연습작으로서 만든 타워디펜스 게임
- 타워 배치 및 아이템을 사용하여 적으로부터 메인 타워를 지키는 게임

● 구현 기술

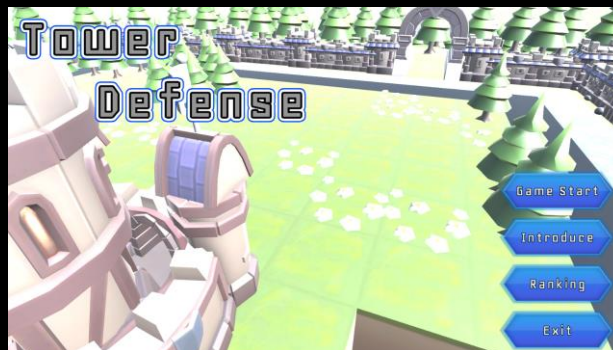
- 엑셀로 맵 시트를 만든 후에 엑셀 파일을 외우는 것이 더 효율적이었기에 엑셀 시트를 읽어서 맵을 만드는 툴을 제작.
- 3D의 리소스가 부족하였기에 2.5D 형식으로 게임을 구성.
- 게임이 조금 더 화사하게 보이기 위하여 기능대회에서 쓸 수 있는 간단한 Bloom Shader를 작성.



인 게임



타워 선택 시



타이틀



랭킹

Github : <https://github.com/Proffeine0327/TowerDefense>

2024 지방 기능대회

#레이싱 #기능경기대회

플 랫 폼 : PC

사용 언어 : C#

개발 인원 : 프로그래밍 1

개발 기간 : 1주

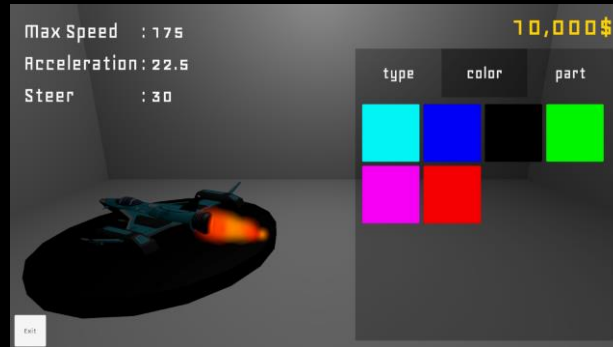
역 할 : 메인 프로그래밍

● 게임 설명

- 지방 기능대회에 출전하기 위해 연습한 레이싱 게임
- 상대보다 먼저 결승선에 들어가면 승리하는 게임
- 돈으로 기체를 업그레이드하면서 적절하게 아이템을 얻어 승리하면 되는 게임

● 구현 기술

- 대회에서도 Tween을 사용하기 위하여 Static Class Extension 을 사용하여 구현.
- Wheel Collider의 단점인 스크립트가 복잡해 진다는 점을 Sphere Collider를 사용하여 해결.



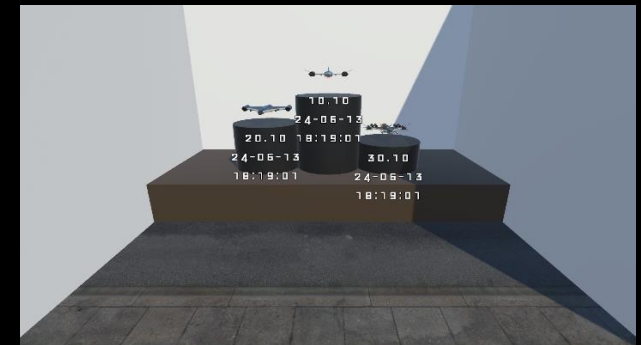
정비소



인게임 1스태이지



타이틀



랭킹

Github : <https://github.com/Proffeine0327/2024-region-tech-competition>

Monster Scramble

#뱀서류 #Brotato #뱀서라이크

플랫폼 : PC

사용 언어 : C#

개발 인원 : 프로그래밍 1

그래픽 1

개발 기간 : 3주

역할 : 메인 프로그래밍

● 게임 설명

- 그래픽 한명과 프로그래밍 한명이 팀이 되어서 진행되는 프로젝트에서 만든 게임
- 스킬을 업그레이드 하여 적을 잡아서 10분을 버티는 게임

● 구현 기술

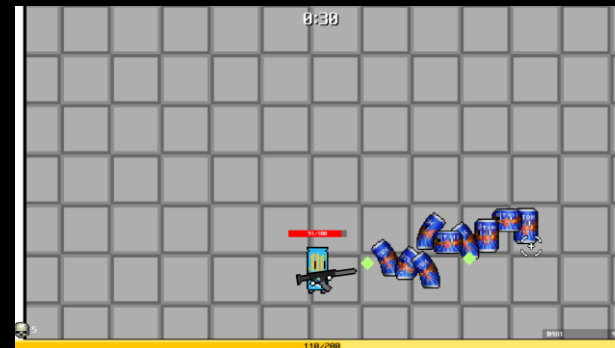
- 테스트 결과 데이터베이스는 관리 방안에서는 효율적이지만 개발 속도가 느린다는 것을 발견. 따라서 Scriptable Object와 Prefab을 데이터베이스 대신 활용하여 개발 속도 향상
- Singleton의 유연성과 결합도에 관련된 문제를 해결하기 위하여 Service Locator 패턴을 사용.



타이틀



캐릭터 선택창



인게임



선택

Github : <https://github.com/Proffeine0327/SchoolPresentationProject>

Pinball Zombie

#핀볼 #좀비 #게임잼

플랫폼 : PC

사용언어 : C#

개발인원 : 프로그래밍 2

그래픽 1

기획 1

개발기간 : 2일

역할 : 메인 프로그래밍

● 게임 설명

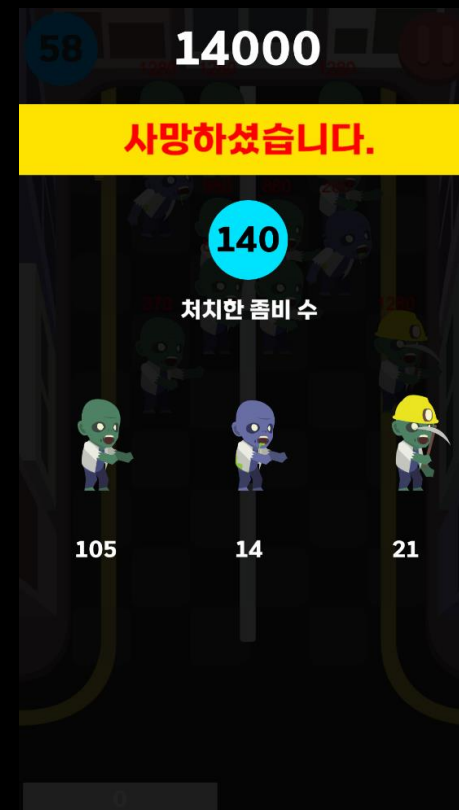
- 교내 게임잼에서 만든 프로젝트
- 계속해서 내려오는 적들을 죽이면서 자신을 강화하고 최대한 오래 살아남으면 되는 게임

● 구현 기술

- 오브젝트를 생성하고 파괴하는데 드는 오버헤드를 줄이기 위하여 Object Pooling 패턴을 사용.
- Singleton의 유연성과 결합도에 관련된 문제를 해결하기 위하여 Service Locator 패턴을 사용



타이틀



엔드 화면



아이템 선택

Github : <https://github.com/Proffeine0327/2022-July-GameJam>

Escape From

#공포 #Pixelate #IMSCARED

플랫폼 : PC

사용언어 : C#

개발인원 : 프로그래밍 2

그래픽 2

기획 1

개발기간 : 1개월

역할 : 1층, 2층, 플레이어 담당

● 게임 설명

- 동아리 게임잼에서 만든 프로젝트
- 각 층의 기믹을 알아내어 다음층으로 올라가는 열쇠를 찾아 최종적으로 탈출해야되는 게임이다.

● 구현 기술

- 게임에 더욱 공포스러운 연출을 내기 위하여 Retro Shader Graph를 제작하여 카메라에 사용.
- Retro Shader Graph를 적용하기 위하여 프로젝트를 URP로 이전해야만 하였고 이로 인해 기존의 Shader를 적용시키기 위하여 Custom Render Feature를 구현.



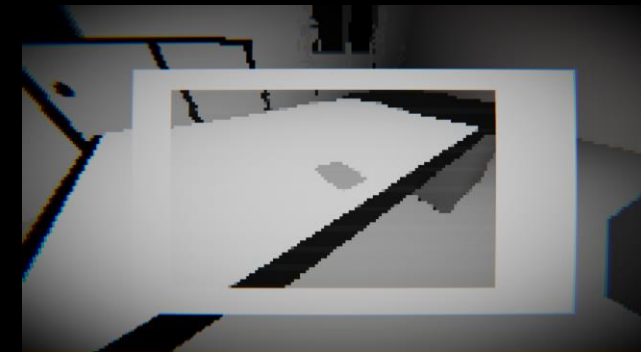
입구 계단



아이템 획득 목록 UI



2층 교실



아이템 획득 장면

Github : <https://github.com/Proffeine0327/EscapeFrom>

Pathfinding Research

#길찾기 #Astar #JPS

플 랫 폼 : PC

사용 언어 : C#

개발 인원 : 프로그래밍 1

개발 기간 : 1일

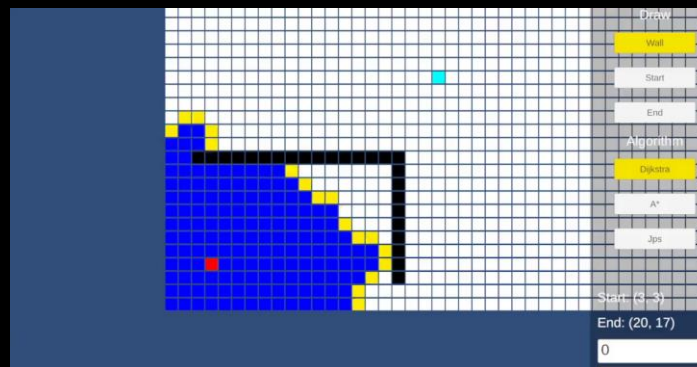
역 할 : 메인 프로그래밍

구현 과정

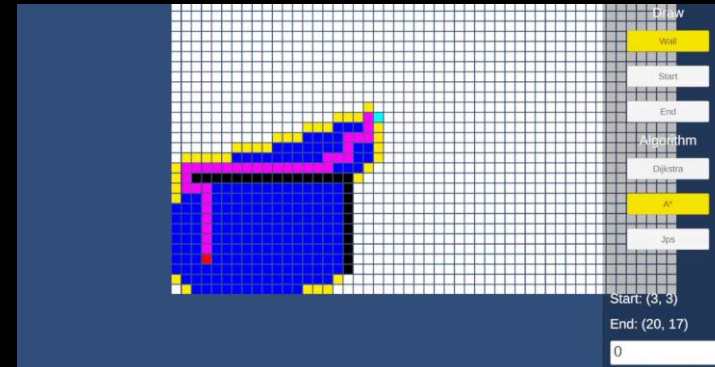
- Pathfinding Algorithm을 실제 게임에 적용해보기 위해 연구한 프로젝트
- Astar와 JPS에 대한 알고리즘을 Unity와 C#에서 구현하여 비교 및 분석
- 맵이 복잡하거나, 한쪽 방향으로 맵이 크게 넓거나 하는 등의 상황에서 JPS가 Astar보다 느린 상황이 발생

결론

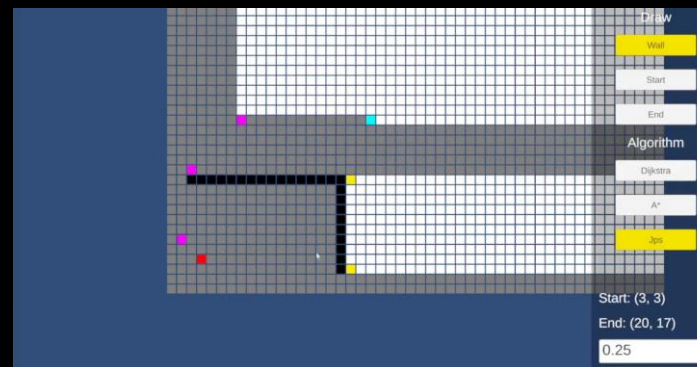
- 맵이 매우 작은 경우, 지형이 복잡한 경우 또는 타겟이 가까이 있는 경우에는 Astar
- 맵이 넓고 지형이 단순한 경우 JPS



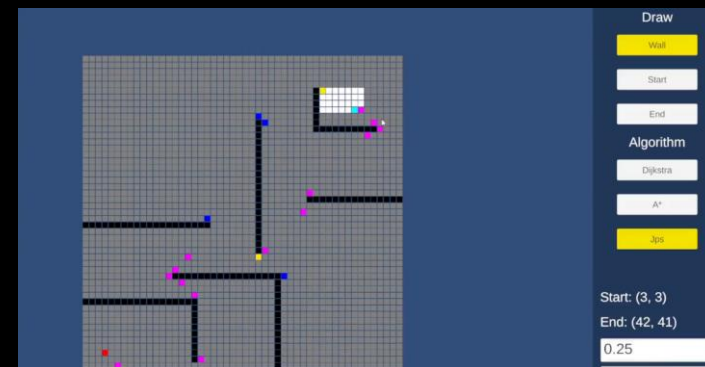
다익스트라



Astar



JPS



JPS 2

Github : <https://github.com/Proffeine0327/pathfinding-research>