총장님 타이쿤 제작 보고서

2020년도 졸업작품

B577027 이재형

목차

[총장님 타이쿤 제작 보고서 1](#_Toc41066063)

[2020년도 졸업작품 1](#_Toc41066064)

[개요 2](#_Toc41066065)

[문서 개요 2](#_Toc41066066)

[배경 2](#_Toc41066067)

[구현 목표 2](#_Toc41066068)

[클래스 다이어그램 2](#_Toc41066069)

[구현 4](#_Toc41066070)

[카메라 4](#_Toc41066071)

[건축 4](#_Toc41066072)

[PiUI 5](#_Toc41066073)

[캐릭터 5](#_Toc41066074)

[튜토리얼 7](#_Toc41066075)

[시간 7](#_Toc41066076)

[메인 UI 7](#_Toc41066077)

[점수 10](#_Toc41066078)

개요

문서 개요

본 문서는 총장님 타이쿤의 구현 과정에 관한 서술을 담았습니다.

서술은 구현한 시스템 순서대로 나열하였습니다.

배경

총장님 타이쿤을 졸업작품으로 선정하게 된 배경은 제가 기획하면서 가장 구현해보고 싶었던 게임 중 하나였습니다. 게임학부 기획 공모전 금상으로 그치기엔 직접 구현해보고 싶은 욕심이 많아 군대에 가기 전에 기획했던 작품임에도 팀원들의 허락을 받아 공모전에 출품하고 이를 졸업작품으로 구현하고자 마음먹었습니다.

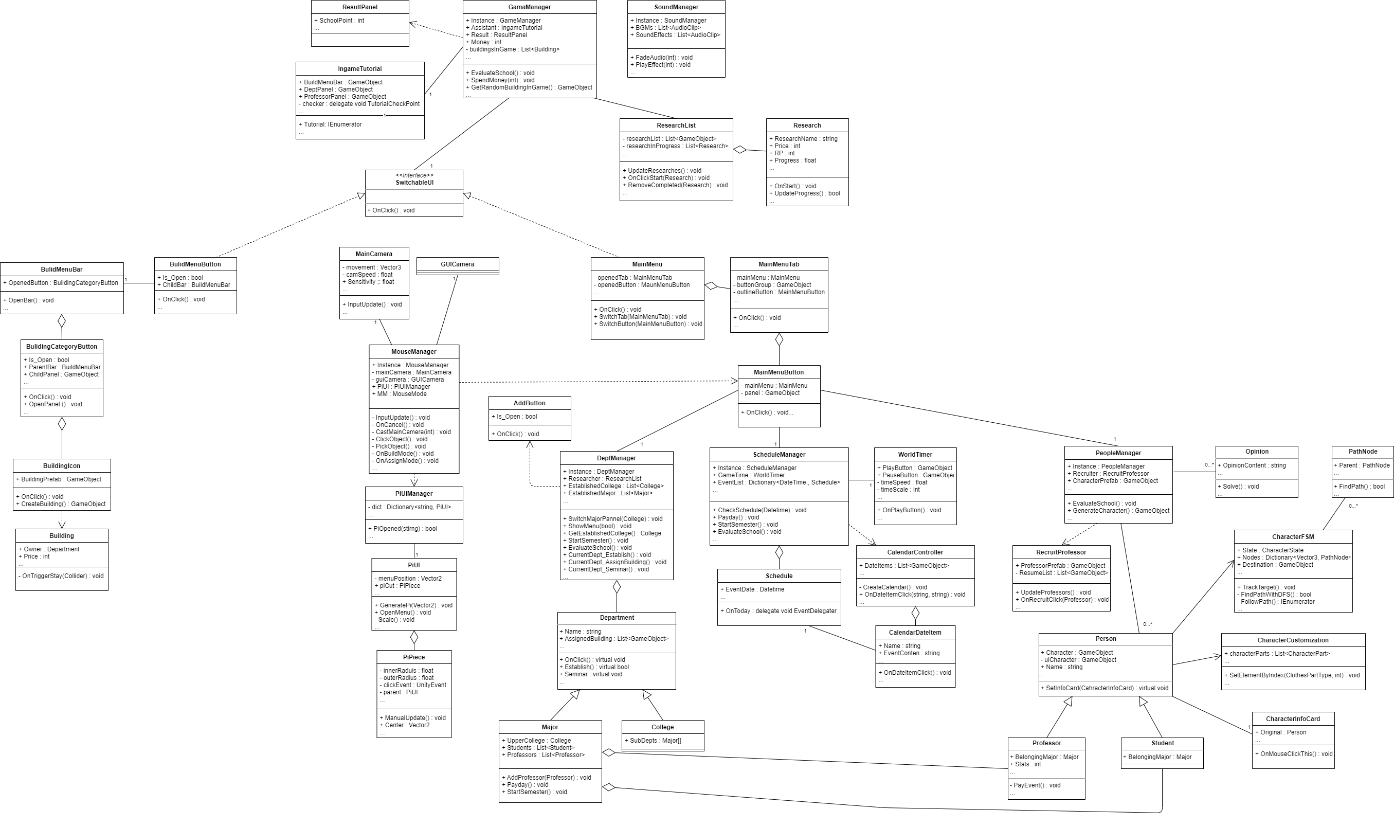
구현 목표

저희가 기획했던 컨텐츠를 최대한 많이 구현해보는 것이 목표였습니다.

여러가지 살리고 싶었던 재미 요소들이 있었는데 전부 구현하지 못해 약간 아쉬운 마음도 있습니다.

클래스 다이어그램

너무 커서 잘 보이지 않는 관계로 이미지 파일을 별첨했습니다.



구현

카메라

우선 기본적으로 타이쿤에서 사용되는 카메라 시스템을 구현하였습니다.

유니티로 여러 프로젝트를 진행하며 카메라를 많이 다뤄왔고 이를 바탕으로 쉽게 구현할 수 있었습니다.

유니티의 Input 시스템을 통해 최대한 부드럽고 편안한 조작감을 유도했습니다.

건축

타이쿤의 가장 기본이 되는 건물 건설로 기반 시스템 작업을 시작했습니다.

건물의 생성부터 시작하기 위해 패널들을 갖고 있는 바 형태의 UI를 제작하고 패널에 건물 모델들을 넣었습니다.   
3D 모델을 따로 스크린샷을 찍어 패널에 넣는 방식이 아닌, 모델 자체를 직접 패널에 넣고 싶었습니다. 하지만 Scroll Rect 내에서 움직이는 건물들이 패널 밖으로 나가도 계속 보이기때문에, 이를 Masking 해줄 필요가 있었습니다. 2D Sprite를 Masking 해본 적은 있지만, 3D 모델을 Masking 해본 적은 없었기에 시행착오를 겪었고, 간단한 Shading을 통해 가릴 수 있었습니다.



우측 초록 건물은 이러한 Shading이 적용된 물체에 의해 가려지게 되었습니다.

그리고 MouseManager 스크립트를 제작하여 이 건물들을 실제 필드로 가져와 설치하는 것을 시도했습니다.   
이를 의도한대로 깔끔하게 진행하기 위해서는, 우선 필드 카메라와 GUI 카메라를 나눌 필요가 있었습니다. 플레이어 입장에선 GUI가 항상 맨 앞에 있기 때문에, ScreenPointToRay 또한 보이는 것처럼 작동할 필요가 있었습니다.  
그리곤 UI를 클릭하여 건물을 생성하고 컴포넌트를 넣어주는 작업, 올바른 위치에선 배치되고 그렇지 않은 곳엔 배치되지 않도록, 건물의 회전도 가능하도록 MouseManager의 마무리 작업들을 했습니다. 분명 MouseManager는 건축 외에 다른 작업들도 처리할 것이기 때문에 건축 관련 메소드를 깔끔하게 정리했습니다.

PiUI

이후 UI 작업을 진행하기 위해 에셋 스토어를 탐색했고, 이런 효과적인 UI가 있다는 걸 발견해 제 프로젝트에 채용했습니다. 다만 이 UI가 생각보다 유연하지 않게 제작되었고, 제 플레이 해상도에 맞게 UI가 표시되도록 수정해 줄 필요가 있었습니다.

그리고 MouseManager를 통해 이 UI를 불러오고 가장 첫번째로 만든 건설 UI를 불러올 수 있도록 연결했습니다.

캐릭터

커스터마이징

캐릭터는 전부 제작된 Preset이 아닌 파츠 하나하나가 따로 제작된 Preset들의 집합입니다. 이렇게 유기적으로 제작된 Low Poly People 에셋을 활용해 캐릭터들이 무작위 의상을 착용하고 나오도록 커스터마이저를 구성했습니다.

길 찾기

졸업작품을 진행하면서 가장 많은 시간을 들이고 수정 및 리팩토링을 진행한 부분입니다.

각 캐릭터들은 목적지로 설정할 수 있는 건물을 향해 길 찾기를 수행하며 길을 성공적으로 찾으면 트랙을 따라 걸어갑니다.

Breadth First Search

처음 제작한 길 찾기 방식은 BFS 방식을 채택했습니다.

캐릭터의 위치에 루트 노드를 생성하고 설정 가능한 n방향으로 지정된 거리만큼 레이 캐스트를 실행해 목적지를 찾고 그 자리에 목적지가 없다면, 가지 노드들을 생성하며 그 가지 노드들 로부터 새로운 가지 노드들을 생성하여 목적지를 찾도록 하였습니다.

이 과정을 실행하니 너무 많은 노드를 생성하며 시간이 오래 걸려 리팩토링할 부분들을 발견했습니다.  
우선 4방향, 6방향으로 레이 캐스트를 실행하였습니다. 이 경우에서는 겹치는 노드들이 발생하기 때문에, 이 노드들을 예외로 처리해서 너무 많은 노드를 생성하지 않도록 하였습니다. 이 과정에서 생성하려는 새로운 노드가 기존 노드와 겹치는지를 판별해야 했는데 최소한의 시간이 걸리는 방식이 필요했습니다.  
최종적으로 Dictionary에 Position값을 갖는 Key를 설정하고, 노드들을 보관했습니다. Dictionary에서 ContainKey 탐색은 O(1)에 가까운 시간 복잡도를 가져 빠른 시간안에 비교가 가능했습니다.

여기까지 완성한 [영상 링크](https://youtu.be/NHlN2bl2IXI)

Best First Search

다음으로 생성되는 노드들이 목적지와 가깝다면 우선적으로 가지를 치는 대상이 되도록 직선 거리를 휴리스틱 값으로 두고 Greedy 하게 수정하였습니다. 하지만 가까운 노드를 우선 탐색 대상으로 두었더니 생각처럼 최적해를 찾지 못했습니다.   
그래서 IComparer를 확장해 거리순으로 Sorting도 시도했지만 노드의 생성 개수는 줄인 대신 정렬 오버헤드로 인해 시간이 더 오래 걸렸습니다.

결국은 빨리 다음으로 넘어가는 게 좋겠다고 생각해서 단순한 형태의 DFS로 길 찾기를 마무리하였습니다.

튜토리얼

행동 하나하나를 알려주는 대화 구문과 체크포인트를 제작해 코루틴으로 진행되도록 하였습니다. 냥비서님은 말풍선을 통해 구문을 하나씩 출력하고 플레이어 행동을 기다리는 delegate 체크포인트에 도달하는데 플레이어가 행동을 수행하면 다음으로 넘어가도록 제작하였습니다. 이후 추가되는 컨텐츠들도 체크포인트화 하여 튜토리얼로 추가하였습니다.

시간

타이쿤 게임에서는 시간을 멈추거나 빨리 흐르게 하는 부분도 필수적인 요소로 여겨집니다. 이 때문에 시간 조작 UI를 넣기 위한 하단 바 UI를 만들면서 본격적으로 UI제작을 시작했습니다.

timeScale과 timeSpeed를 나눠 게임 흐름의 배속과 시간 흐름의 배속을 구분하였습니다. 그래서 플레이어는 timeScale을 조작해 게임 흐름을 임의 배속할 수 있고, 스크립트로는 연출상 필요한 부분에서 인게임 시간이 너무 빨리 지나가 이벤트들을 생략하지 않도록 timeSpeed를 조작하였습니다.

이 작업을 하면서 동시에 배속에 영향을 받지 않아야 하는 요소들 (카메라, UI연출 등)을 수정해주었습니다.

메인 UI

다른 타이쿤 게임의 메인 UI를 참고하여 큰 구분을 탭으로, 작은 구분을 버튼으로 하는 방식을 따왔습니다. 이렇게 분류된 항목들을 패널에 표시하여 플레이어가 UI를 조작할 수 있도록 Switching 하였습니다.

학사

학교 경영에 가장 기본적으로 필요한 대학, 학과를 개설하는 시스템과 그 UI가 필요했습니다. 이런 대학과 학과를 하나의 부서 (Department)로 보고 캡슐화 하여 이 부서들을 관리하는 DeptManager를 구현하여 UI 표시와 동시에 내부 시스템이 돌아가도록 하였습니다.

건물 할당

그리고 이렇게 개설된 부서가 건물들을 할당 받을 수 있도록 MouseManager에 할당모드를 추가하여 후에 부서 소유의 건물 개수와 부서 소속 학생 수에 맞게 만족도 점수가 조정되도록 하였습니다.

교수

다음으로 작업한 것은 개설된 학과에서 수업을 진행할 수 있도록 교수님을 뽑는 시스템이었습니다. 채용된 교수님들을 학교 활동의 전반적인 부분들에 영향을 미치도록 네 가지의 스탯을 나누고, 이를 바탕으로 연봉이 책정되도록 설정하였습니다.

우선 이렇게 랜덤으로 생성된 교수님을 학교의 소속으로 만들기 위해, PeopleManager와 ProfessorRecruiter를 제작해 각각 소속 교수 관리, 이력서 관리 역할을 수행하도록 하였습니다. 리크루터는 이력서 3개를 무작위로 생성하고 플레이어가 채용할 수 있도록 버튼을 추가시키는데, 이 때 캐릭터를 생성하는 일을 PeopleManager에게 의탁해 받아오도록 하였습니다.   
그리고 채용될 때, 이력서와 캐릭터 정보를 매니저에게 넘겨줌으로써 이력서는 인원 패널에 추가되고 UI캐릭터를 바탕으로 복사된 필드 캐릭터를 필드에 배치시키고 정보 카드를 부여해주었습니다.

정보 카드

메인 UI에 표시되는 이력서뿐만 아니라, 필드에 생성된 캐릭터의 정보를 바로 보여주기 위해 정보 카드도 따로 제작하였습니다. 이렇게 제작된 정보 카드는 캐릭터를 클릭했을 때 보일 수 있도록 MouseManager에 기능을 추가해주었습니다.

인원

채용했던 교수님들을 PeopleManager가 관리하는데, 이를 보여줄 UI도 필요했습니다. 유니티에서 제공하는 Scroll Rect와 Horizontal Layout을 활용하고 건물 패널을 만들 때 같이 제작했던 3D Object Mask도 재활용하여 교수님들의 정보를 나열하였습니다.

일정

건물과 학과, 그리고 교수님들까지 마련되었으니 이제 학교가 수업을 시작할 기본적인 준비가 되었습니다. 학사 일정을 수행하기 위해, ScheduleManager를 생성하고 정해진 날짜가 되면 이벤트가 시작되도록 Schedule을 생성해 날짜를 key로, Schedule을 Value로 갖는 Dictionary에 보관하였습니다.

날짜가 흐를 때 마다 이 Dictionary에 Schedule을 탐색해 학사 일정을 실행하도록 Schedule에 수행해야 할 일정 Delegate를 만들어 Schedule이 Observer, Manager가 Subject인 옵저버 패턴을 구축하였습니다.

그리고 일정 UI에서 달력을 생성할 때, 해당 날짜에 적힌 일정 내용을 띄울 수 있도록 저장된 Schedule들의 값을 참조하여 생성되도록 구현하였습니다.

연구

학교 평가 점수를 높이기 위한 수단으로 연구 컨텐츠도 추가하였습니다. 대학이 설립되면 해당 대학이 진행할 수 있는 연구 프로젝트를 받을 수 있습니다. 이 프로젝트들은 각각 진행에 필요한 비용과 그 비용의 1/10 만큼의 얻을 수 있는 연구점수를 랜덤으로 갖고 생성됩니다.

시간이 지남에 따라 이 연구점수를 채우기 위한 연구 진행도가 차오르며, 연구 진행도는 해당 연구를 진행하는 대학 소속의 교수들이 가진 연구 능력들의 총 합이 높아질수록 더욱 빠르게 진행됩니다.

의견

플레이를 진행하다 보면 PeopleManager가 갖고 있는 Person들 중 한 명이 학교에 의견을 제시합니다. 이 Person은 학생일수도, 교수님 일수도 있습니다.

사람들은 학교에 필요한 부분들을 제안함으로써 플레이어에게 처리할 일들을 제공하고 이를 통해 만족도 점수를 얻게 됩니다.

이렇게 얻게 된 만족도는 메인 메뉴의 인원-개요에서 얼굴 표정과 게이지의 형태로 표시되도록 UI를 제작하였습니다.

점수

학교의 등급은 5가지 점수의 총 합으로 산정했습니다.

재정 점수

재정 점수는 플레이어가 점수 산정 시점에 보유한 현금의 액수에 따라 결정됩니다. $200,000를 최적의 현금 보유량 기준으로 두고, 이보다 작거나 크다면 점수가 점점 줄어드는 방식을 적용했습니다.  
이를 구현하기 위해, 정규분포의 그래프를 채용하였습니다. 평균을 200,000으로, 분산을 100,000으로 두어 적당히 점수가 고르게 분포되도록 그래프를 유도하였습니다. 이 그래프의 x값에 현금 보유량을 대입해 보유한 현금이 20만 달러와 멀어지게 된다면 계산되는 재정 점수는 낮아지도록 계산하였습니다.

수업 점수

수업 점수는 교수님들의 수업 점수의 총합으로 결정했습니다.

만족도 점수

만족도는 두가지 항목의 합으로 산정했습니다.

* 부서에 할당된 건물의 충족도
* 제시된 요구들을 해결할 때 마다 얻는 만족도

연구 점수

연구가 완료될 때 마다 얻게 되는 RP의 총 합으로 산정했습니다.

명성 점수

명성 점수는 학교 교수님들의 명성 스탯의 평균으로 산정되는데 ‘학교 점수에 영향력을 미치는 수치’ 라는 컨셉으로 기준을 정했습니다.

총 점

이렇게 산정된 점수들은 아래와 같은 식으로 총 점을 매기게 됩니다.

SchoolPoint = (ResearchPoint + TeachPoint) \* (FamePoint / 10) + SatisfactionPoint + FinancialPoint

등급

이렇게 총 점수를 5000점씩 얻을 때 마다 한 등급씩 높게 책정되도록 하였습니다. 총 20,000점 이상을 달성한다면 A등급을 받게 됩니다.