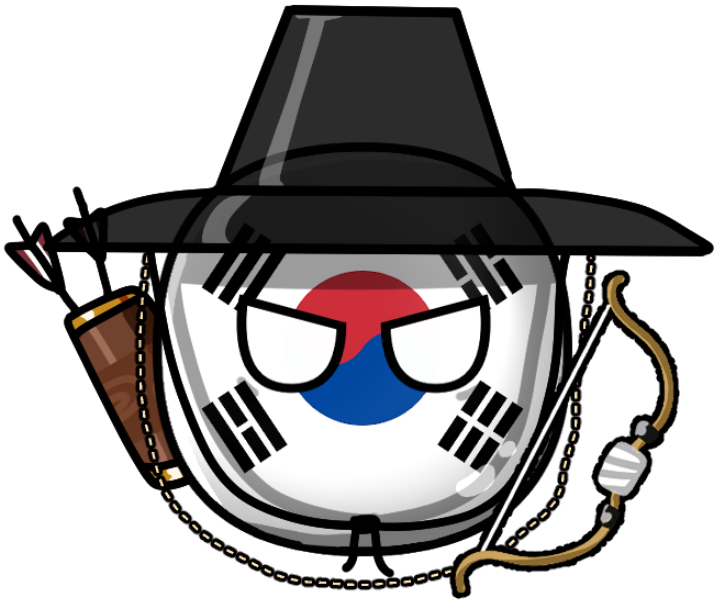
World Tower Defense(전남대 캡스톤)

FrameWork	Unity
category	게임
develop period	@2021년 3월 1일 → 2021년 6월 30일
link	https://github.com/jiyullee/WorldTD
roll	프로그래밍
size	middle
언어	C#

개요

- 전남대학교 4학년 캡스톤 졸업 작품



게임설명

월드타워디펜스는 지구에 침공한 적을 막기위해 여러 국가들이 협력하는 타워 디펜스 게임입니다.

똑같은 타워 3개를 조합하여 총 3단계 진화를 해 적을 막을 수 있습니다.

코인을 투자해 레벨을 올리면 타워를 설치할 수 있는 갯수가 많아지고 뽑을 수 있는 타워의 레벨은 올라갑니다.

유전 알고리즘을 게임 벨런스 테스트에 활용할 수 있는지 테스트 해보는 게임

- 칙센트미하이가 제시한 몰입 이론에 기반을 두고 상대가 얼마나 몰입 하게 하는지에 고안한 결과 적당한 레벨 벨런스의 필요를 느낌

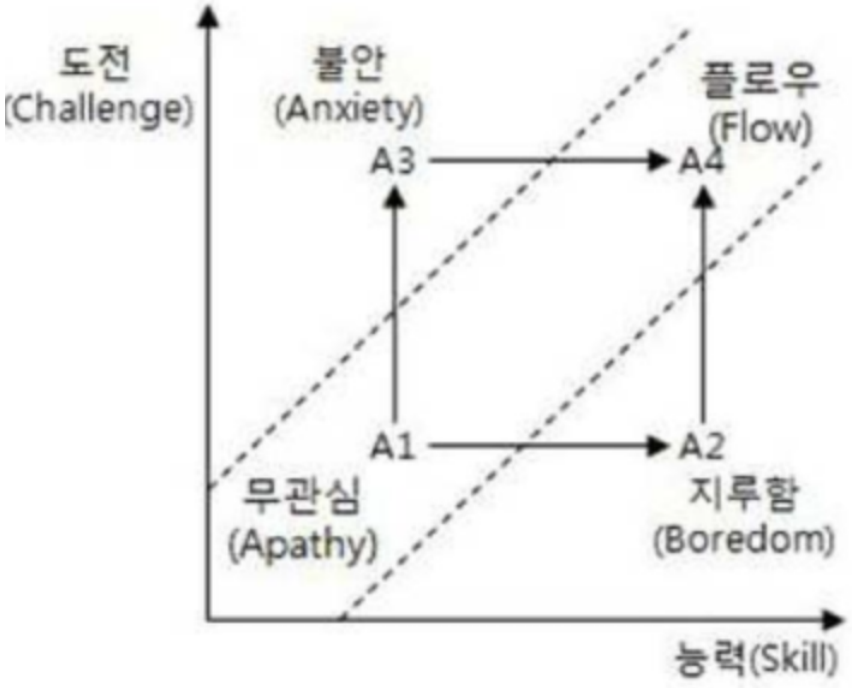
  이를 유전자 알고리즘을 통해 조절하기로 생각하여 게임을 고안하던도중 디펜스 게임이 해당 게임에 적합하다고 판단됨.

  여기서 도전 = 난이도, 능력 = 타워의 조합, 개수 등으로 판별 하였고

  적합성을 판단하는 방법으로 1. 소모된 체력, 2. 소요된 시간 등의 요소가 고안되었음.

  이중 소모된 체력과 시간등을 고민한 결과 시간이 적합하다고 판단되었음

  > 체력을 소모할 경우 비슷한 시간이 걸린것을 보게되어서 몬스터가 전부 없어지는데 까지 걸리는 시간과 가장 오래 걸리는 시간을 백분율로 나타내 적합성을 평가함



구글 앱 스토어

월드 타워 디펜스 - Apps on Google Play

Everyone Translate the description into English (United States) using Google Translate? 평화로운 지구에 적이 침공하였습니다.여러 국가의 협력을 통해 적을 물리치세요! 개발자 연락처

https://play.google.com/store/apps/details?id=com.Capstone.WorldTowerDefense

플레이 영상

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/10facc6f-4a27-49ab-920f-73f19d2f1781/WorlTd.mp4

개발

▼ 우측의 삼각형을 누르게 되면 세부 사항을 볼 수 있습니다.

아래의 여러 항목에서 이 삼각형을 누르면 세부 사항을 볼 수 있습니다.

git Flow를 사용하여 여러 브런치를 통해 개발하고 서로의 코드를 보고 수정 및 피드백을 카카오톡을 통해 공유함

본인 파트

유전 알고리즘을 게임 벨런스 테스트에 활용할 수 있는지 테스트 해보는 게임
AI를 사용하지 않고 코드를 통해 군집의 조합을 통하여 벨런스를 조절하는지 테스트해 보는 게임

- 유전 알고리즘을 통한 게임 밸런싱 테스트에 관심을 둠

    1. 게임 벨런스 방향 확립
       알고리즘을 생각하다 생물의 진화를 모티브로 한 유전 알고리즘을 찾음 ⇒ 강자 생존
       유전 알고리즘을 적용할 방안을 적과 아군의 스탯 조절로 생각함
       아군은 시너지에 의해 크게 좌우될 수 있고 적합한 기준을 찾기 어려움
       적군의 스탯을 설정하기로 하였으나 간단하게 수치만 조절할 경우 플레이어가 눈치 채지 못하고 너무 단조로워짐
       ⇒ 적의 군집을 설정하여 군집 조합을 유전자로 표현하여 최적의 유전자 선택, 교배, 조합을 베이스로 기획자가 생각한 최적의 시간에 도달하게 유도
    2. 적용 방법
       앞서 이야기 한 대로 시간을 기점으로 적합성을 판단함
       최고 플레이 시간은 몬스터가 전부 나오는 데까지 걸리는 시간 + 몬스터가 처음부터 끝까지 가는 시간을 설정하고
       플레이 타임은 큐에 담긴 몬스터가 전부 사라질 때까지 소요된 시간으로 측정함

- Design Patterns

    - 싱글톤
      싱글톤을 활용하여 게임매니저들의 호출 및 구현을 용의하게 함
      일반 싱글톤 코드를 제네릭 클래스로 구현할 수 있도록 팀원의 도움을 받음
    - 오브젝트 풀링
      오브젝트 풀링을 이용하여 몬스터와 총알을 소환할 수 있도록 풀링 매니저를 구현함
      풀링 매니저는 해당하는 총알, 몬스터, 타워의 상위 오브젝트를 가지고 있으며 해당

오브젝트를 String을 통해 부를 경우 딕셔너리에서 해당하는 큐를 찾아서 오브젝트를 반환하고 큐에 없을 경우 새로 만들어줌
오브젝트를 반환하는 과정에서 데미지, 이미지 등을 할당하게 도와줌

- 데이터 관리 및 파서 생성
  CSV 파일을 통해 파싱하고 해당 객체를 만드는 딕셔너리를 구현함
  <String, Class>를 하여 String(Key) 값을 통해 해당 클래스를 참조하는 형식을 채용하였고, 이는 몬스터, 타워, 레벨 디자인 등등 여러가지 객체에 사용됨
  딕셔너리를 적극적으로 활용하고, 파싱한 객체를 class 화 하여 저장함

2020.08.28회고

새로 알게된점

1. 파서를 좀 더 능동적으로 다룰 수 있게됨
2. 싱글톤과 딕셔너리형 자료구조를 이용하면 오브젝트 및 데이터 관리에 용의하다는 점을 알게됨
3. 제네릭 클래스를 알고는 있었지만 사용법에 대해 크게 고민하지 않았으나 이것을 통해 알게됨
   - Where을 이용한 제약조건

아쉬운점

1. 레벨 디자인 테스트를 자동화 해주는 코드를 만들어서 했으면 좋았을것 같다.

   ⇒ 타워를 자동으로 배치해주는 코드를 고안하다가 머신러닝을 이용하지 않을 경우 어렵다고 생각함. 유니티 텐서플로우 책도 있으니 찾아볼것
2. 자신의 부족함을 체감하게 되었다
   - 제네릭 클래스의 사용법, 유니티에서 모노비헤이비어를 상속하는데 상속하지 않은 클래스의 필요성(파서)
3. 다른 사람의 코드를 건드릴 경우 미리 상의해서 건드리는게 좋다는점
   - 개발 이외의 부분

     프로그래머의 코드는 한사람의 작품이고 이를 상의도없이 건들게 되면 사람의 자존심을 상하게 하고, 의욕을 떨어뜨릴 수 있다는 점을 느낌
   - 개발 부분

자신이 짠 코드를 남이 리팩토링 하게 되면 자신의 코드를 이해하기 어려울 경우가 생기고, 버그가 생길경우 불화가 생길수도있다

의문점

- 이러한 방법(벨런스 디자인 툴을 만들어서) 실제로 적용 가능하는지
- 지금까지 유니티 인스팩터를 통해서 이벤트 설정(ex : 버튼 클릭)을 하였는데 코드로 하게되니 신박하였고 이를 사용해보고싶다는 생각이들었고, 실무에서는 어떻게 사용되는지 궁금해짐

  현재 생각해본 장단점

  1. 장점
     - 인스팩터 즉 유니티 자체 기능에 의존을 줄여줌
     - 해당 버튼의 코드만 찾게 되면 여러가지 기능을 할당할 수 있다고 생각
  2. 단점
     - 직관적이지 않다(인스펙터에 보이지 않아 버그가 생길 경우 찾기 어려울 수 있음)
     - 유니티의 플레이 버튼을 통해 실행하지 않는한 테스트가 어렵다