**포트폴리오**

클라이언트 부문

한국산업기술대학교

게임공학과

장동필

2018.09.18

* 목차

1. Turtle Bomb
2. Penguin Duo
3. **Turtle Bomb**



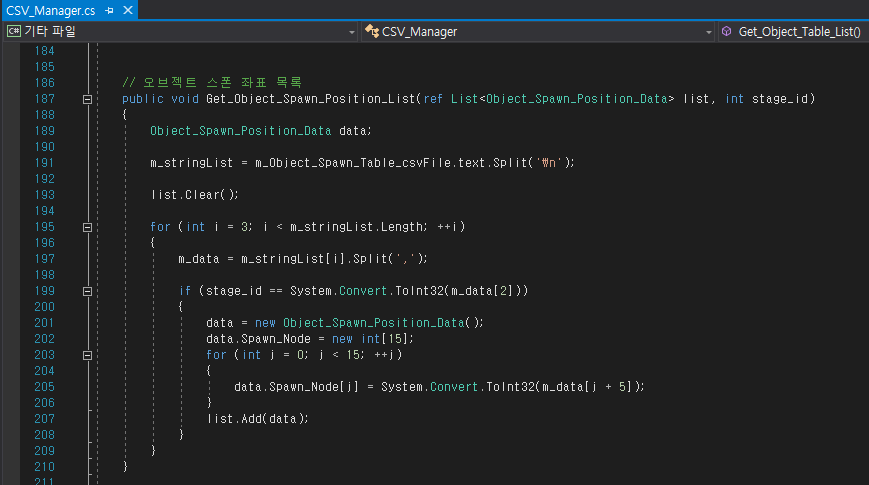
* **장르 :** 모바일 캐주얼 아케이드 (3인칭)
* **게임 소개 :** 거북이 (플레이어)가 폭탄으로 적 (침략자 고블린 무리들)을 물리치는 Bomber Man 류의 게임.
* **제작 환경 :** Unity 2017 engine (C#)
* **제작 기간 :** 약 1년 (2017.09 ~ 2018.09)
* **제작 인원 :** 3인
* **맡은 역할 :** 클라이언트 제작
* **구현 내용 :** ‘앱 시작’ ~ ‘모험 모드’ 까지의 기능 제작. (Player 조작, Monster AI, UI 기능, Effect 편집 및 적용, 아이템, 미션, … 구현)
* **제작 난관 :**

1. **스테이지 수에 따라 증가하는 Scene 개수:** 스테이지에 따라 In-Game Scene을 여러 개 만드는 것 대신, 하나의 Scene에서 오브젝트 배치, 몬스터 Stat 등의 데이터들을 달리 설정하여 띄워주는 것을 구현하기 위해 외부에서 데이터들을 관리할 필요가 생겼다. 이어서 스테이지에 입장할 때 해당 스테이지가 몇 번 스테이지인지 알 수 있도록 스테이지 번호를 관리해야하는 일도 증가했다.
2. **렌더링 최적화 문제:** Scene에서 출력할 Batch 수가 너무 높고, 해상도 조절을 하지 않아 모바일 기기에서 낮은 FPS 성능을 보였다.
3. **많은 양, 서로 얽혀있는 스크립트 문제:** 동일한 (또는 유사한) 기능을 가진 객체들의 스크립트를 제작할 때, 상속을 사용하지 않고 모든 기능을 각각 작성하여 스크립트들의 크기가 커지면서 관리가 힘들어졌다. 또, 스크립트들이 스파게티 식으로 얽혀 버려서 Scene에 특정 오브젝트가 먼저 나오지 않으면 에러가 생기기도 했다.
4. **렌더링 최적화를 진행하기 이전에, 스크립트 상의 문제라고 생각했던 오브젝트 생성/소멸 문제:** 오브젝트들을 전부 인스턴싱으로 생성하고 특정 순간에는 삭제를 하게 되면서, 가비지 컬렉션 때문에 FPS가 급감 하는 것이라고 생각했었다.

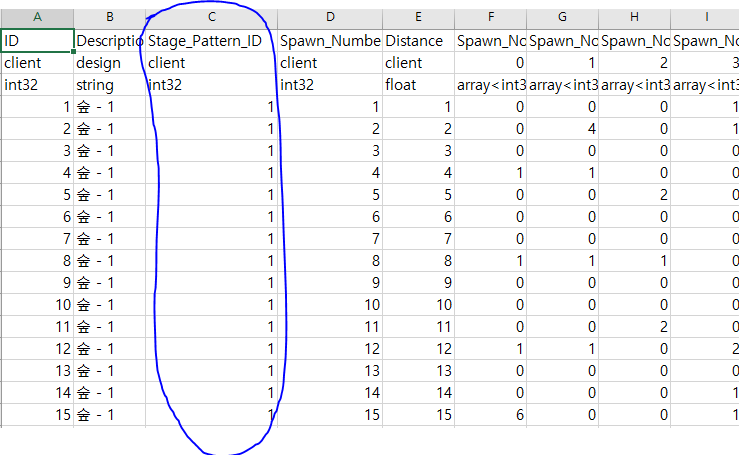
* **난관 극복 :**

1. **스테이지 수에 따라 증가하는 Scene 개수:** 엑셀 파일로 데이터 테이블을 만들고, CSV 파일로 변환하여 사용했다. 각 스테이지에 대해 고유번호를 지정하여, 데이터 테이블이 여러가지 있더라도 선택한 스테이지에 맞는 내용을 각각 읽어올 수 있도록 했다.
2. **렌더링 최적화 문제:** Batch를 위해서는 동일한 속성의 Material을 가져야 한다는 것을 알게 되었고, 렌더링 순서가 연속적이어야 한다는 것도 알게 되었다. 프레임 디버거를 확인하며 Render Queue와 Order in Layer 값을 적절히 조정하는 작업을 했다. 그리고 해상도를 기존 2560x1440 에서 1280x720으로 줄이는 것이 스크립트 안에서 조절 가능하다는 것을 알게 되었다. 이들을 수정한 후 모바일 기기에서 돌려보니 이전에 10~20 FPS가 나오던 것에서 현재 60 FPS가 나오는 효과를 보게 되었다.
3. **많은 양, 서로 얽혀있는 스크립트 문제:** 동일한 (또는 유사한) 기능을 사용해야하는 오브젝트들을 각각 추려내서, 상속을 통해 구현했다. 이렇게 구현하니 사용하기도 간편하고 수정에도 용이해졌다.
4. **렌더링 최적화를 진행하기 이전에, 스크립트 상의 문제라고 생각했던 오브젝트 생성/소멸 문제:** 오브젝트 풀링으로 구현하여 사용했다. 풀링 만을 담당하는 객체를 만들거나, 한번 인스턴싱으로 생성한 객체를 삭제하지 않고, 잘 보관해 두며 사용하도록 했다. 이를 통해 아주 약간의 FPS 성능이 향상되는 효과를 얻었다.

* **수상 및 전시 :** 2018성남 인디게임 공모전 개발지원 부문 우수상 수상 및 전시, 2018 산업기술대전 전시
* **소개 영상 :** <https://www.youtube.com/watch?v=lQ6Y9ieEmNY>
* **GitHub 주소 :** <https://github.com/RRE1012/Turtle_Bomb>
* **핵심 소스 코드 :**

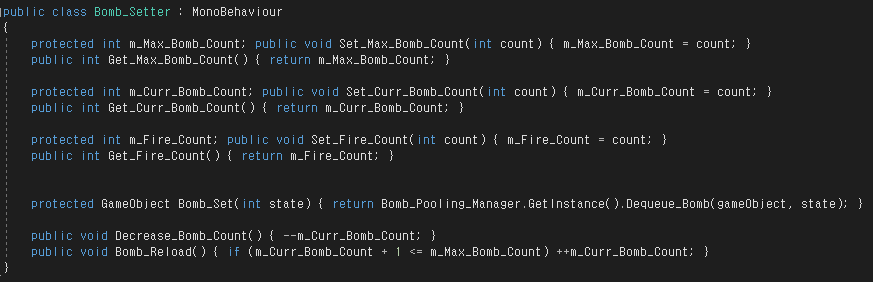
1. **CSV 파일을 읽어와 파싱하여 데이터를 사용하도록 했다. **

**(↑데이터 테이블을 Parsing하는 스크립트의 일부)**

****

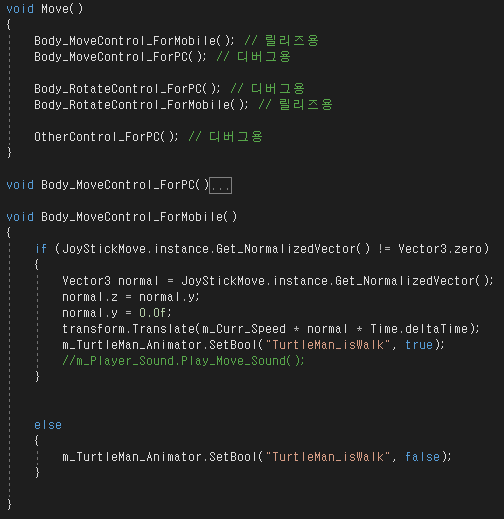
**(↑오브젝트 스폰 좌표 데이터 테이블)**

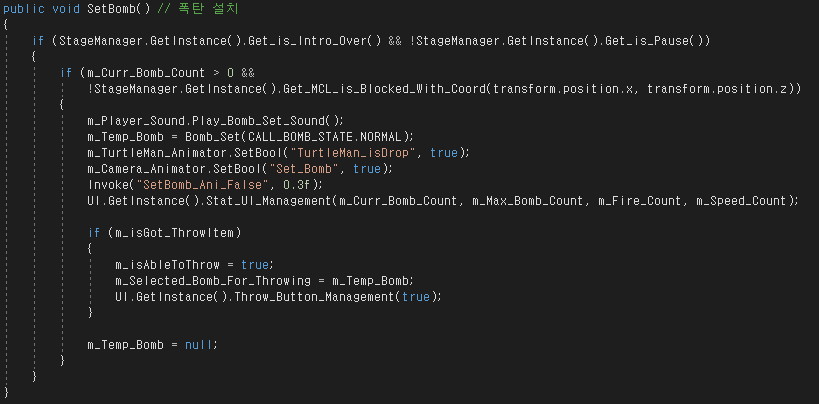
1. **상속**



-> 폭탄을 설치할 수 있는 객체 (캐릭터)를 사용함에 있어서 그 기반이 되도록 하는 클래스인 Bomb\_Setter. 폭탄을 사용할 때 반드시 가져야할 데이터를 가지게 하며 폭탄 설치 시 ‘폭탄 풀링 매니저’를 부르도록 한다. 이후 이 클래스를 상속하여 새로운 캐릭터들을 제작하도록 한다.

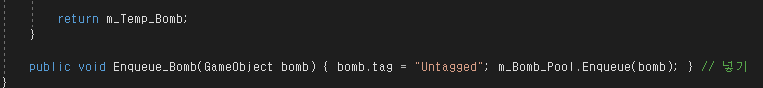
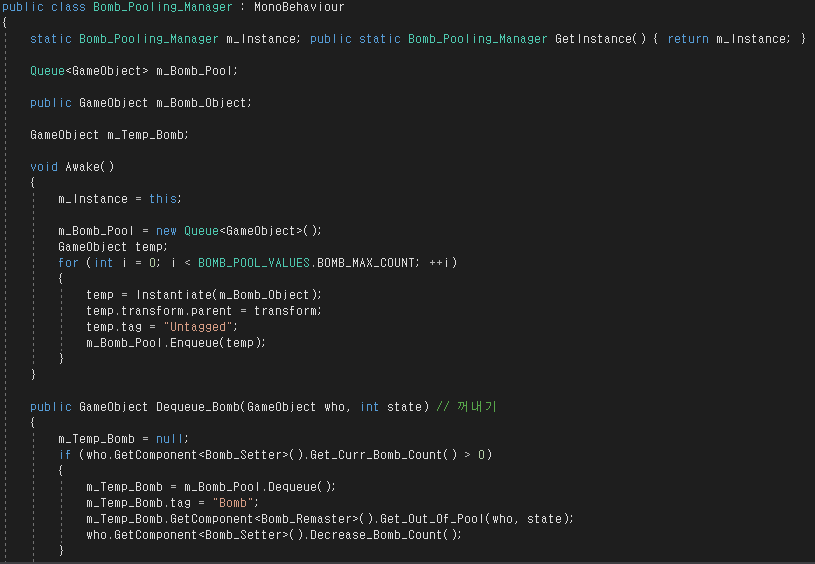
Player_상속 -> Bomb\_Setter를 상속한 Player.



-> 조이스틱을 이용한 이동이라던지,

-> 폭탄을 설치할 때 수행할 것들을 정의하는 등 플레이어에게 필요한 기능을 추가적으로 다룬다.

1. **폭탄 오브젝트 풀링**



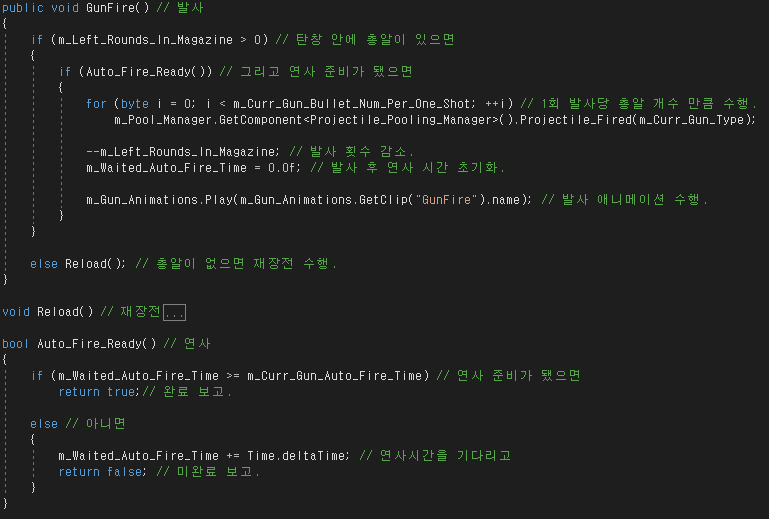
-> 싱글톤으로 구성하여 최초 씬 시작 시 정의한 폭탄 개수만큼 폭탄을 생성하여 큐에 저장한다. 이후 폭탄을 사용할 때 인스턴스를 통해 인큐/디큐 할 수 있도록 구현했다.

1. **Penguin Duo**

****

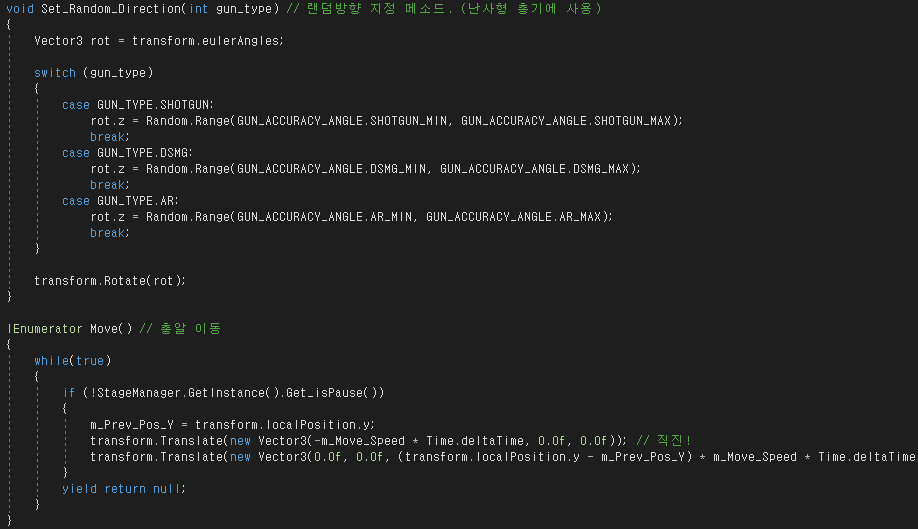
* **장르 :** 2.5D 횡스크롤 레이싱. (2.5D라 표현한 이유는 2D 그래픽 게임이지만 사실상 오브젝트들이 3차원 상에서 움직이고 있기 때문.)
* **게임 소개 :** 녹아버린 남극의 빙산 위에서 포식자들에게 쫓기는 두 마리의 펭귄 형제들이 적을 물리치며 한없이 도망치는 레이싱 게임.
* **제작 환경 :** Unity 2017 engine (C#)
* **제작 기간 :** 약 2주 (2018.7~2018.8)
* **제작 인원 :** 1인
* **맡은 역할 :** 클라이언트 제작 및 리소스 제작.
* **구현 내용 :** 게임에 사용된 모든 비주얼 리소스 그림 (제작), 플레이어 기능 제작 (총기류 발사, 이동, 사망, …), 몬스터 별 AI 제작, 장애물 제작, 아이템 제작, 데이터 테이블 (패턴 테이블) 제작, UI 제작, …
* **핵심 소스 코드 :**

1. **총기**

****

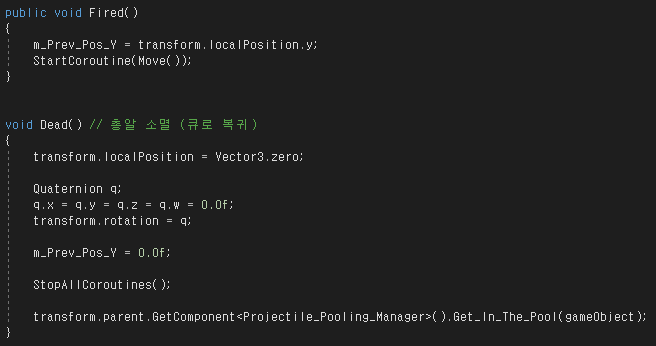
-> 총기의 발사 과정. GunFire() 메소드가 주축이 되며, 이는 Player의 Attack() 메소드에서 호출된다. 발사 조건 충족 시 풀링된 투사체를 발사할 개수 만큼 꺼내 투사체에게 발사 명령을 내린다.

1. **투사체**

****

-> 투사체가 꺼내졌을 때 랜덤 방향으로 이동하도록 하게 해주는 메소드. 각각의 총기의 특성에 따라 랜덤 범위가 달라진다.

지정된 방향에 따라 투사체는 직진을 하게 되며 또한 Y좌표에 비례하여 Z축 이동까지 하게 된다. (다른 오브젝트들 처럼)

****

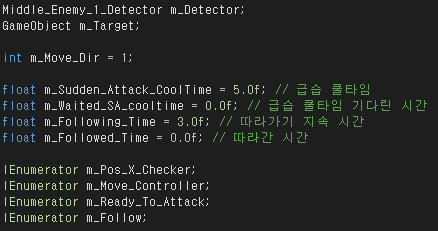
-> 오브젝트에 닿아 소멸 시 초기화 작업 수행 후 풀로 복귀시킨다.

1. **중형 몬스터 AI**

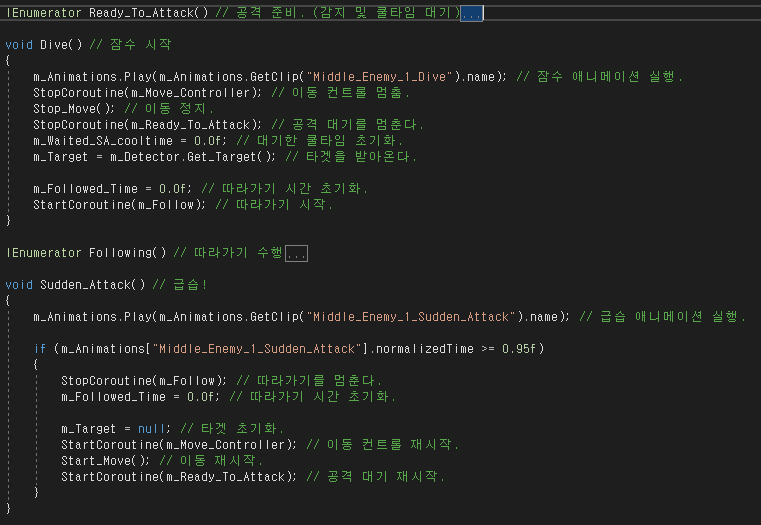
**middle**

-> 몬스터들은 Enemy 클래스를 상속한다.

Enemy 클래스에는 초기 스탯 설정, 시스템 설정, 이동과 피격, 죽음, 충돌과 같은 메소드들을 다룬다.



-> 중형적이 가지게 되는 변수들과 상태들. 각 상태들은 유일하게 동작한다. Pos\_X 체크와 이동 컨트롤러는 각종 연출에 사용된다.



-> Ready\_To\_Attack()과 Following() 상태.

공격 준비는 플레이어를 감지하기만 하는 상태이다. 감지를 하게 되면 Dive()를 호출하여 Following 상태로 전환 시키고, 플레이어를 따라 Y축과 Z축 이동을 수행하다가 일정 시간이 되면 급습을 수행한다. 급습 연출이 끝나면 위의 X좌표 체크 상태와 이동 컨트롤러를 이용하여 제자리로 복귀한 뒤 다시 공격 준비 상태로 돌아간다.

* **제작 난관 :**

1. 제대로 된 설계를 하지 않고 만들어서 기존 기능들을 계속 갈아엎는 일이 발생.
2. 리소스를 모두 직접 그리다 보니 시간이 많이 소모됨.
3. 애니메이션과 효과음, 배경음을 직접 만들 역량이 안됨. (현재 게임 내에 아무런 사운드도 없으며, 스프라이트 애니메이션 또한 없음)
4. 끝이 없는 게임이다 보니 몬스터와 장애물, 아이템들의 종류와 등장 패턴들이 상당히 많이 필요하지만 1인 개발로서 한계를 느낌.

* **난관 극복 :**

1. 만들고자 하는 기능을 명확하게 하고 필요한 기능들을 정리할 시간을 가짐으로써 해결 (설계 진행). 게임 기획의 중요성을 깨달음.
2. 각 필요 리소스 별로 템플릿 리소스를 만들어 그 위에 덧칠 (스케치) 하는 방향으로 제작 하면서 제작 시간 최소화.
3. 현재 해결 불가. 그림 실력과 작곡 능력을 갖출 필요가 있음.
4. 3번과 유사. 그러나 이는 쉽게 해결할 방법이 존재 (타작품 모방) .견문을 넓힐 필요가 있음.

* **소개 영상 :** <https://www.youtube.com/watch?v=cUWlOkwOv8Y>
* **GitHub 주소 :** <https://github.com/endl524/2018_MINIGAME>