‘오브젝트 풀링’을 이용한 탄막 게임 메모리 최적화

이정[1], 최문기[2], 성진협[3], 김영종✉
*숭실대학교 소프트웨어학부
✉숭실대학교 소프트웨어학부
leekkzzzz@naver.com, tjdwlsguq@naver.com, cmg3172@naver.com,
✉youngjong@ssu.ac.kr

Memory Optimization in Bullet Hell Game using ‘Object Pooling’

Jung Lee[1], Moongi Choe[2], Jinhyeop Sung[3], Youngjong Kim✉
*School of Software, Soongsil University
✉School of Software, Soongsil University

요 약

컴퓨터 부품의 고성능화로 점점 게임이 요구하는 사양이 높아지는 추세이다. 따라서 게임 개발에 있어 최적화가 필수적으로 요구된다. 본 논문에서는 ‘오브젝트 풀링’이라는 메모리 최적화 기법을 소개한다. 따라서 ‘오브젝트 풀링’을 적용한 탄막 게임을 만들어 메모리 최적화를 직접 구현해보고 연구한다.

1. 서론

기본적으로 게임 개발에 있어 중요하게 다뤄지는 부분은 최적화이다. 그래픽이 좋아질수록 그래픽 최적화가 중요하고, 게임 내 오브젝트 혹은 엔티티가 많을수록 메모리 최적화가 중요하다. 물론 이 둘은 비중의 차이는 있더라도 동시에 이뤄져야 한다.

탄막 게임은 수백에서 수십만 개의 탄환 오브젝트가 생성 혹은 제거되는 게임으로 메모리 최적화가 중요시되는 게임 중 하나이다.

따라서 본 논문에서는 ‘오브젝트 풀링’을 이용해 탄막 게임에서의 메모리 최적화를 구현해보고자 한다.

2. 관련 연구

게임 개발에 요구되는 최적화 사항 중에는 크게 그래픽과 메모리가 있다. 그래픽 최적화 기법에는 ‘오클루전 컬링’과 ‘Low Polygon’ 등이 있고, 메모리 최적화 기법에는 ‘가비지 컬렉션 최적화’와 ‘오브젝트 풀링’ 등 다양한 종류의 최적화 기법이 존재한다. 본 단락에서는 ‘가비지 컬렉션’과 ‘오브젝트 풀링’에 대해 살펴보고자 한다.

2.1 가비지 컬렉션(Garbage Collection)

(그림 1) Unity 로고

Unity의 경우 사용하지 않는 메모리를 찾아서 비우는 프로세스를 가비지 컬렉션이라고 한다. Unity의 가비지 컬렉션은 스크립트가 관리되는 힙에 메모리를 할당할 때 이를 수용할 수 있는 힙 메모리가 충분하지 않으면 가비지 컬렉터를 실행한다. 가비지 콜렉터가 실행되면 힙의 모든 오브젝트를 검사하고 애플리케이션에서 더 이상 레퍼런스가 없는 오브젝트를 삭제하도록 표시한다. 그런 다음 레퍼런스가 없는 오브젝트를 삭제하여 메모리를 확보한다. [1] 하지만 이 과정에서 메모리 해제 시 남은 메모리를 재정렬하지 않아, 공간은 충분하지만 메모리 할당을 못 하는 경우가 발생할 수도 있고 가비지 컬렉션이 실행되는 중에는 다른 동작을 지연시키기 때문에 가비지 컬렉션은 게임 내 성능저하로 직결된다.

2.2 오브젝트 풀링(Object Pooling)

기존 가비지 컬렉션은 메모리 부족으로 인해 기존에 생성된 오브젝트를 삭제하는 반면에, ‘오브젝트 풀

링'은 오브젝트를 삭제하지 않고 이를 재활용하기 위해 풀(Pool)로 다시 가져와 보관하는 기법이다.

해당 기법을 적용할 경우, 게임 내 오브젝트들이 특정 조건(캐릭터 사망, 충돌 판정, 화면 이탈 등등)에 의해 삭제되어야 할 때, 가비지 컬렉션을 통해 메모리에서 삭제되지 않고 비활성화(Disable)하여 풀(Pool)에 저장함으로써 언제든 해당 오브젝트가 필요하면 풀(Pool)에서 다시 가져와 쓸 수 있게 된다. 따라서 이 최적화 기법은 한번 생성된 오브젝트에 대하여 삭제하는 시간도 단축하지만 동시에 생성하는 시간도 단축한다.

3. 탄막 게임

(그림 2) 탄막 게임 예시

탄막 게임은 정해진 패턴으로 공격하는 적 오브젝트의 탄환을 전부 피하며 플레이어 캐릭터가 적 오브젝트를 파괴하는 게임으로, 적으면 수백 개에서, 많으면 수십만 개의 탄환 오브젝트의 생성 혹은 제거가 동시에 이뤄져 메모리 과부하가 잘 일어나는 게임으로 본 논문에서는 '오브젝트 풀링' 메모리 최적화 기법을 활용하여 탄막 게임 내 성능 개선을 구현하고자 한다.

3.1 탄환 오브젝트 풀링

탄막 게임에서 가비지 컬렉션으로 인한 성능저하가 발생하지 않도록 push() 함수와 pull() 함수를 통해 풀(Pool)에 넣고 빼는 기능을 추가해 '오브젝트 풀링'을 적용한다.

```
push(){
    object disable;
    push object to pool;
}
pull(){
    pull object from pool;
    object enable;
}
```

(표 1) push(), pull() 함수

push(), pull() 함수를 이용하여 풀(Pool)에 이미 생성되어있는 오브젝트를 활성화(Enable), 비활성화(Disable)한다.

```
bulletDestroy{
    if bullet collides with player or
                                screen outer space
        push();
    if bullet life time is over
        push();
}
```

(표 2) 탄환을 삭제하는 클래스

bulletDestroy() 함수는 탄환의 수명이 다하거나 화면 바깥의 공간 또는 플레이어와 충돌하면 탄환은 push() 함수를 호출하여 탄환 오브젝트를 풀(Pool)에 넣는다.

```
spawnBullet(){
    for(the number of bullet to spawn){
        pull();
    }
}
```

(표 3) 탄환을 생성하는 함수

spawnBullet() 생성해야 하는 탄환의 개수만큼 pull() 함수를 호출하여 풀(Pool)에서 탄환 오브젝트를 가져온다.

4. 결론

Unity에서 기본으로 제공되는 '가비지 컬렉션'을 통한 탄막 게임과 '오브젝트 풀링' 기법을 적용한 탄막 게임을 동일한 환경, 동일한 조건으로 실행시켰을 때 '오브젝트 풀링'을 적용한 탄막 게임이 성능 면에서 더 좋은 결과를 낼 것이다.

참고문헌

[1]"Unity 사용자 매뉴얼 : 가비지 컬렉터 개요", https://docs.unity3d.com/kr/2021.3/Manual/performance-garbage-collector.html