

Invader.c

리뷰 및 리팩토링

NVIDIA AI ACADEMY 8기
윤성진

목차

1. 코드 리뷰
2. 문제점 분석
3. 개선 중점
4. 개선 사항

점수 : 400

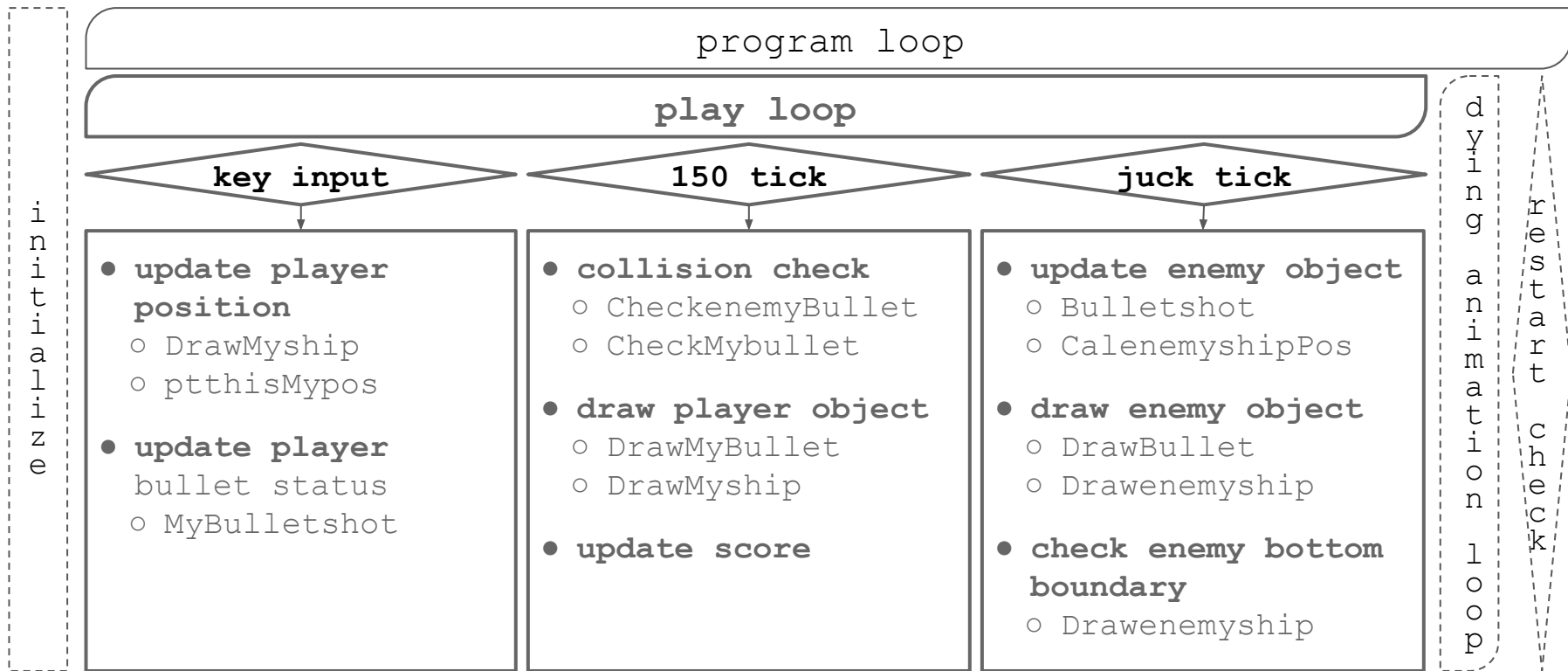
^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^
^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^
^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^
^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^

당신의 비행기는 파괴되었습니다.
다시 할까요? (y/n)

코드 리뷰

- `_kbhit()`과 `_getch()`를 이용한 키보드 입력
- `gotoxy()`와 `printf()` 를 이용한 콘솔 출력
 - 업데이트할 부분을 빈칸문자로 지우고 문자열로 채움
- 윈도우 틱에 의한 프레임 제어
 - 고정 플레이어 틱 마다 플레이어 오브젝트 업데이트
 - 유동 적 틱 마다 대한 적 업데이트

코드 리뷰



문제점 분석

- 버그

- 벽 타고 하강하는 적
- 적 사망 모션 일부 잔류

- 상호작용

- 플레이어 이동 중 과도한 깜빡임
- 경직된 플레이어 움직임

- 아트

- 텍스처 크기 변경 제한
- 플레이어와 적의 투사체 구별 어려움

```
int Calflag()
{
    int add = 1;

    if(enemyship[0].pos.x < 1) flag = TRUE;
    else if(enemyship[0].pos.x > 40-3) flag = FALSE;
    else add = 0;
    return add;
}
```

문제점 분석

● 버그

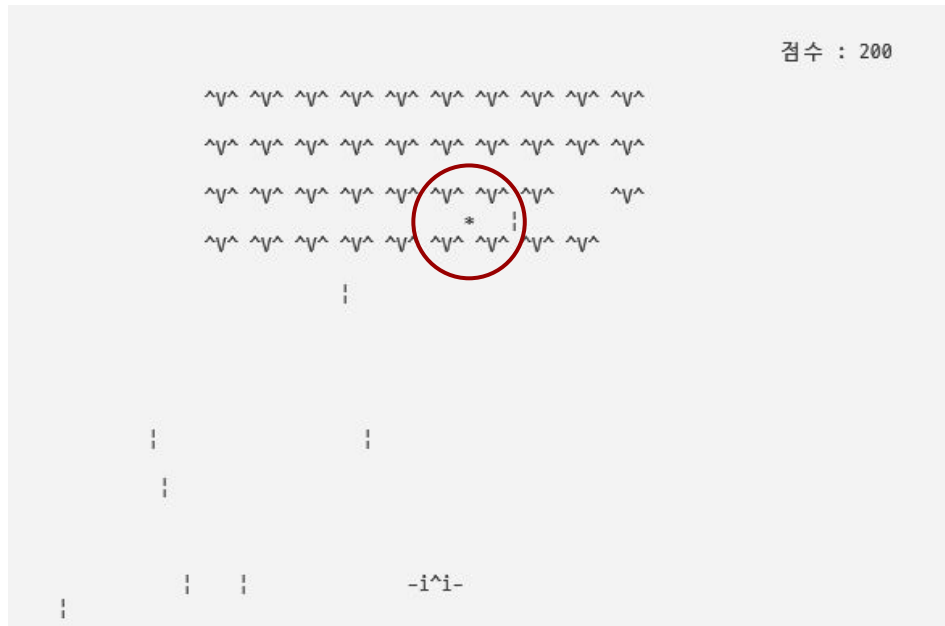
- 벽 타고 하강하는 적
- 적 사망 모션 일부 잔류

● 상호작용

- 플레이어 이동 중 과도한 깜빡임
- 경직된 플레이어 움직임

● 아트

- 텍스처 크기 변경 제한
- 플레이어와 적의 투사체 구별 어려움



문제점 분석

- 버그

- 벽 타고 하강하는 적
- 적 사망 모션 일부 잔류

- 상호작용

- 플레이어 이동 중 과도한 깜빡임
- 경직된 플레이어 움직임

- 아트

- 텍스처 크기 변경 제한
- 플레이어와 적의 투사체 구별 어려움

- gotoxy와 printf 를 이용한 콘솔 출력

- 업데이트할 부분을 빈칸문자로 지우고 문자열로 채움

- 이동 때마다 플레이어가 지워지고 다시 그려지며 깜빡임 (flickering) 발생

문제점 분석

- 버그

- 벽 타고 하강하는 적
- 적 사망 모션 일부 잔류

- 상호작용

- 플레이어 이동 중 과도한 깜빡임
- 경직된 플레이어 움직임

- 아트

- 텍스처 크기 변경 제한
- 플레이어와 적의 투사체 구별 어려움

- `_kbhit`과 `_getch`를 이용한 키보드 입력

- 키보드 연속 입력 제한
- 버퍼로 인식하기에 느린 반응속도

개선 중점

- 부드러운 움직임과 화면 출력
- 커스터마이징이 간편한 게임 세팅과 텍스처
- 완성된 게임 요소 - 타이틀, 프롤로그, 플레이, 엔딩
- 확장성과 이식성을 고려한 설계

개선 사항 - InvaderC.c

- **콘솔 버퍼를 이용한 더블버퍼 출력** 구성
 - 여분의 콘솔버퍼에 한번에 오브젝트를 그린 뒤 버퍼 전환 - **깜빡임 없는 부드러운 출력**
- 위치 업데이트 / 충돌판정 / 텍스처 드로잉 과정 분리
- GetAsyncKeyState()를 이용한 **실시간 키보드 입력 매핑**
- **2D 오브젝트**를 그리기 위한 스프라이트 및 히트맵 구현
 - Object2D 구조체를 통한 2D 오브젝트 관리
 - WriteConsoleA() 이용한 콘솔 직접 출력
 - 불리언 히트맵 기반의 충돌 판정
 - **asset.c** 파일 분리하여 원하는 크기의 스프라이트를 원하는 프레임만큼 미리 정의 가능

개선 사항 - InvaderC.c

- 전역 설정 상수 맵 분리
 - 컴파일 전에 게임 수정 가능
- 구조체 기반 리팩토링
 - 많은 수의 변수 관리 및 확장 시 편의 개선
- 역할에 따른 소스 및 헤더 파일 분리
 - 윈도우 **api** 기반 코드 **console.c**로 분리, 윈도우 타입 어레이싱 - 원활한 플랫폼 변경 가능
 - 플레이어와 적을 오브젝트에서부터 각각 구현하고 관련 함수를 각 소스파일에 분리
- 크로스 플랫폼 빌드 위한 **CMake** 기반 빌드 환경으로 전환

개선 사항 - InvaderC.c

- 개발 경험 개선
 - 콘솔 및 윈도우 조작에 대한 인터페이스를 제공하여 간편한 개발 가능!
- 플레이 경험 개선
 - **WASD** 기반 플레이어 이동, 애로우키 기반 **8방향** 방향 총알 발사
 - 가상의 적 바운더리 오브젝트를 구성하여 바운스 및 하부 경계 조건을 단순화
- 플레이 비주얼 개선
 - 타이틀, 프롤로그 스토리, 사망 원인 메세지 추가
 - 적 모두 격추시 다음 적이 나오는 구간에 애니메이션 추가
 - 플레이어 이동과 총알 방향에 따른 스프라이트 변경

InvaderC.c

감사합니다

NVIDIA AI ACADEMY 8기
윤성진