**프로젝트 보고서: C언어를 활용한 우주선 슈팅 게임**

**박성원**

**1. 프로젝트 개요**

C 언어를 활용하여 개발한 우주선 슈팅 게임 프로젝트이며 이 프로젝트는 시작화면, 플레이어 및 적의 움직임, 총알 발사, 충돌 감지, 점수 표시 등의 요소가 포함되어 있습니다.

프로젝트 목표: C 언어를 사용하여 우주선 슈팅 게임 구현

포함된 기능: 시작화면, 플레이어 및 적의 움직임, 총알 발사, 충돌 감지, 점수 표시 등

**2. 관련 연구**

\* Windows.h/ stdbool.h 함수에 대한 공부

\* 위치 이동 함수 'gotoxy' 구현

\* 'kbhit()' 대신 'GetAsyncKeyState()' 함수 사용법

\* 랜덤 출현 구현 방법

\* 캐릭터 잔상, 무한 생성 문제 해결

\* 메모리 최적화

**3. 프로젝트 설계**

**- 구현된 기능 설명**

\*시작화면: 게임 실행, 도움말 보기, 게임 종료 등의 옵션 제공, 아스키 아트로 시작화면 꾸미기

\*게임 실행

-플레이어 조작, 속도 조절, 총알 발사 가능

-적의 움직임: 랜덤 위치에서 생성하여 일정 경로 이동, 화면 경계에 닿으면 방향 변경

-충돌 감지 및 점수: 플레이어 및 적의 총알 간 충돌 감지, 점수 표시

\*게임 종료: 플레이어 생명 모두 소진 시 종료, 최종 점수 표시 후 게임 종료

**- 추후 업데이트 할 기능**

\* 아이템 및 업그레이드

게임 플레이 중 얻을 수 있는 아이템(추가 생명, 강화된 무기 등)을 도입하여 흥미도를 높임.

\*레벨 난이도 조절

레벨별로 난이도 조절을 만들어 게임의 진행이 점차 어려워지거나 다양한 장애물이 추가되는 등의 요소를 추가.

\*사운드 효과

게임 플레이 시에 음악 또는 효과음을 추가하여 게임의 즐거움을 증가

\* 저장 및 랭킹 시스템: 플레이어의 최고 점수를 저장하고 랭킹으로 보여주는 시스템을 추가하여 경쟁 요소를 부여할 수 있습니다.

**4. 프로그램 구현**

**- 구현한 내용**

**1. 헤더 파일**

\* player\_functions.h

텍스트 색상 설정, 커서 숨기기 및 콘솔 화면 상의 커서 이동과 관련된 함수들의 선언이 포함되어 있습니다.

\* game\_functions.h

게임 메뉴, 도움말 화면 표시, 플레이어 아이콘 그리기, 플레이어 아이콘 지우기 등과 관련된 함수들의 선언이 포함되어 있습니다.

**2. 구현 파일**

\*player\_functions.c

player\_functions.h에 선언된 textcolor(), eraseCursor(), gotoxy() 등의 함수들을 Windows API 및 　　콘솔 함수를 사용하여 정의하고 있습니다.

\*game\_functions.c

game\_functions.h에 선언된 playerdraw(), erasePlayer(), help(), startmenu() 등의 함수들을 구현하여 게임 요소 표시 및 사용자 인터페이스를 처리하고 있습니다.

**3. 메인 프로그램 (Shooting\_game)**

주된 게임 루프가 구현되어 있으며, 플레이어 이동, 총알 발사, 적 생성, 충돌 감지 등 게임의 상태를 업데이트하는 부분이 구현되어 있습니다.

＊　사용자 입력 처리

GetAsyncKeyState()를 사용하여 사용자 입력을 처리하며, 플레이어 이동 및 총알 발사 등을 처리하고 있습니다.

이 코드는 콘솔을 이용하여 간단한 게임을 구현한 것으로, 플레이어는 키보드를 이용하여 이동하고 총알을 발사하여 적을 제거하는 게임입니다. 사용자가 메뉴를 선택하여 게임을 시작하거나 도움말을 확인할 수 있습니다. 게임이 시작되면 적이 나타나고, 총을 쏘면서 적을 공격하고 충돌을 감지하여 게임 상태를 업데이트합니다. 게임은 플레이어의 목숨이 모두 소멸될 때까지 계속됩니다.

**-개발 일지**

2023-10-08

시작화면 완성

2023-10-31

시작화면 아스키 아트 수정 및 색추가, 도움말 화면 완성

gamestart 부분 수정중에 kbhit은 작동 안하는 컴파일러가 있다.

-> GetAsyncKeyState로 키를 누르고 있을때 작동하로고 설정

방향키 이동 구현 (이동 구간 끝까지 이동시 벽에 튕겨 반대로 가는 것을 추가로 구현 하였다.)

방향키 이동할때 캐릭터가 무한생성 오류 발생

2023-11-02

q,e 속도조절 키워드 추가

space 총알 구현

2023-11-15 캐릭터 잔상 조금 해결

적 랜덤생성 구현

적 생성 후 잔상문제 해결

2023-11-20 콘솔 크기 변경으로 캐릭터, 적 잔상 해결

적이 움직이면서 랜덤으로 총알 발사

적이 쏜 총알에 맞으면 목숨 감소 구현

2023-11-25 점수 구현

해결 해야함 문제: 적에게 총알을 맞췄을 때 예외 발생

해결완료

적을 맞췄을 시 +10 표시

2023-12-03 적 생성 위치 수정 완료

적을 맞추면 +10 표시 제거 (잔상 문제)

적을 맞췄을 때 남은 잔상 제거

사용자 움직임 공간 줄임

총알 출력 방식 변경

2023-12-10 시작화면 스타일 변경

최종 점검 완료

오류 검사 완료

2023-12-13

시작화면 알파벳에서 “●”모양으로 변경

태두리 추가

**5. 프로그램 동작과 결과**

**-주요 코드 리뷰**

해당 코드는 여러 가지 기능을 구현하기 위해 함수로 모듈화되어 있으며, 각 함수는 주석을 통해 기능과 역할에 대해 자세히 설명되어 있습니다. 변수명 및 함수명은 명확하게 지어져 가독성을 높이고 있습니다. 아래는 출력 결과와 설명입니다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

※ 2023-12-13일 추가: 알파벳 모양에서 문자 “●”로 수정 태두리 추가

1. 시작화면

시작화면은 먼저 “SHOOTING GAME”을 아스키 아트로 제작하여서 “game\_function.h”해더 파일안에 함수인 “startmanu()“함수에 저장 하였습니다. 1번을 클릭하게 되면 게임이 시작되고 2번을 누르면 조작법이 적혀있는 help창으로 넘어가게 됩니다. 마지막으로 3번을 입력하게되면 게임이 종료됩니다.

2. 도움말 창

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

도움말 창에서는 조작법을 설명하고 있습니다. 번호 입력 부분에 2번을 입력하면 도움말 창이 뜨고, Windows.h의 함수인 system(“pause”)를 사용하여 Windows 환경에서 콘솔 창이 바로 닫히는 것을 막기 위해 사용됩니다. 아무키 입력시 다시 시작화면으로 돌아갑니다.

3. 게임 플레이

스크린샷, 우주, 천문학이(가) 표시된 사진

자동 생성된 설명

방향키를 입력하여 사용자를 움직이고 적의 총알 “☆”을 피하며 스페이스바를 입력하여 적을 맞춰야 합니다. 적을 맞추면 점수 10점을 얻게 됩니다.

이때 블로그에서는 “conio.h” 파일의 함수 “kbhit()” 함수를 사용하였지만, 코드의 변활를 주고 싶어서 찾아보다가 저는“Windows.h“ 파일의 함수 ”GetAsyncKeyState()“을 사용 하였습니다.

// 방향키 입력 처리void processDirectionInput(unsigned char\* ch) { if (GetAsyncKeyState(VK\_UP) & 0x8000) { \*ch = UP; } else if (GetAsyncKeyState(VK\_DOWN) & 0x8000) { \*ch = DOWN; } else if (GetAsyncKeyState(VK\_LEFT) & 0x8000) { \*ch = LEFT; } else if (GetAsyncKeyState(VK\_RIGHT) & 0x8000) { \*ch = RIGHT; }}

GetAsyncKeyState() 함수를 사용하여 화살표 키 (위, 아래, 왼쪽, 오른쪽) 중 어떤 키가 눌렸는지를 감지하고, 해당 방향을 나타내는 값을 변수 ch에 설정하는 역할을 합니다. ->GetAsyncKeyState() 함수를 사용하면 프로그램이 특정 키의 눌림 여부를 계속적으로 확인할 수 있습니다. 여러 키를 동시에 누르거나 키가 눌리고 있는 동안 다른 작업을 수행할 수 있게 해주므로, 게임이나 대화형 프로그램 등에서 사용자 입력을 처리할 때 유용하게 사용

4. 플레이어 채력

플레이어 체력은 총 3개로 설정 되어있습니다. 적의 총알에 충돌하게 되면 하트가 감소합니다. 하트 3개가 모두 소진될시 게임이 종료되며 “THE END”라는 문구와 함께 밑에 총 점수가 출력됩니다.

\*목숨 3개 충돌 2개로 감소 충돌 1개로 감소

--> --> 

\* 하트 모두 소진시 게임 종료

스크린샷, 컴퓨터, 천문학이(가) 표시된 사진

자동 생성된 설명

**6. 결과 분석 및 토의**

**-결과**

\*시작화면: 아스키 아트도 정확하게 뜨면서, 번호 옵션별로 잘 작동함.

\*플레이 화면: 사용자 움직임, 키 입력처리, 적 출현/이동/제거, 적/사용자 총알, 충돌부분 모두 설계대로 출력되고 있음.

**- 개선 및 추가 구현 사항**

\* 적의 움직임 패턴을 더 다양하게 구현하여 게임의 난이도를 조절하도록 나중에 개인적으로 수정 할 예정입니다.

\* 시각적 효과 업데이트나 사운드 효과를 추가하여 게임의 흥미를 높일 수 있습니다.

\* 추가적으로 4번 설계에 적은 내용대로 나중에 업데이트를 통해 아이템을 생성하여 보너스 점수를 준다던 지, 아니면 총알이 무한대로 발사가 된다던 지, 아니면 안 좋은 아이템으로 방향키가 역으로 바뀐다던 지 등 아이템을 구연했으면 합니다.

\*그 외에도 난이도를 만들고, 사운드를 추가하면 더욱더 흥미도가 올라갈 것 같습니다.

\* 플레이를 하다가 보면 적을 맞췄을 때 잔상 같은 것이 많이 남습니다. 최대한 제거를 해볼려고 했지만 프로젝트 기간 안에 구현하지 못한 것이 너무 아쉽습니다.

\*적이 총알을 여러 방향으로 출력 됐으면 좋았을 태지만 코딩능력 부족으로 아직 구현하지 못한 것이 아쉽습니다.

\*시작화면을 더욱더 꾸몄으면 게임의 완성도를 증가시켰을 텐데 아쉽습니다.

**7. 결론**

해당 프로젝트를 통해 C 언어를 사용하여 슈팅 게임을 구현하는 과정을 배우고 익힐 수 있었습니다. 그리고 익숙하지 않았던 해더 파일들과 함수를 블로그를 통하여 많은 기능들을 배웠고 코드에 구현할 수 있었습니다. 특히나 사용자가 키를 입력했을 때 움직임 구현하는 것이 가장 힘든 순간 이였던 것 같습니다. 그리고 아스키 아트를 구현하는 데에 시간이 많이 걸렸습니다. 1학기라는 짧은 기간동안 게임 하나를 개발했다는 것이 뿌듯하고 기분이 좋았습니다. 그치만 프로그램을 설계할 때 모든 기능들을 추가하지 못하여서 아쉽습니다. 추가적인 기능 구현이나 개선 사항을 고려하여 게임을 추후 업데이트를 진행을 하고 싶습니다.

**8. 참조 문헌**

코드는 여러 블로그 구글링, GPT의 도움을 통해 학습하여서 제작하였습니다.

대구가톨릭대학교 c프로그래밍 (1분반) 이론보고서 참조

윤성우의 열혈 C 프로그래밍(개정판) 오렌지미디어 2010 책 참조

<https://dongkyung.tistory.com/15> 블로그 참조하여 제작

(키 입력 처리 부분 kbhit() 대신 GetAsyncKeyState 함수 사용, 속도조정 함수 추가, 시작화면 아스키 아트 변경, 적 잔상 오류수정 등 주석으로 표시)

<https://cinrueom.tistory.com/27> GetAsyncKeyState 함수 블로그 참조

<https://coding-factory.tistory.com/663> text color 색 정의 블로그 참조

<https://blog.naver.com/noksek0615/221555514567> 커서 설정 블로그 참조

<https://hyomyo.tistory.com/27> gotoxy정의 블로그 참조

<https://ko.wikipedia.org/wiki/Windows.h> windows.h 해더파일의 함수들