**C++프로그래밍**

**프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | *Snake\_Game* |
| 팀 명 | *J3* |
| 문서 제목 | 결과보고서 |

|  |  |
| --- | --- |
| **Version** | 1.1.5 |
| **Date** | 2022-Jun-15 |

|  |  |
| --- | --- |
| **팀원** | 이지형 |
| 이재원 |
| 정호용 |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 C++프로그래밍 수강 학생 중 프로젝트 “Snake-game”를 수행하는 팀 “J3”의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 “J3”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | J3-프로젝트수행결과보고서.doc |
| **원안작성자** | 이지형, 이재원, 정호용 |
| **수정작업자** | 이지형, 이재원, 정호용 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2022-06-15 | 이지형 | 1.0 | 최초 작성 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[1 개요 4](#_Toc43103653)

[2 개발 내용 및 결과물 5](#_Toc43103654)

[2.1 목표 5](#_Toc43103655)

[2.2 개발 내용 및 결과물 6](#_Toc43103656)

[2.2.1 개발 내용 6](#_Toc43103657)

[2.2.2 시스템 구조 및 설계도 6](#_Toc43103658)

[2.2.3 활용/개발된 기술 6](#_Toc43103659)

[2.2.4 현실적 제한 요소 및 그 해결 방안 6](#_Toc43103660)

[2.2.5 결과물 목록 7](#_Toc43103661)

[3 자기평가 8](#_Toc43103662)

[4 참고 문헌 8](#_Toc43103663)

[5 부록 8](#_Toc43103664)

[5.1 사용자 매뉴얼 8](#_Toc43103665)

[5.2 설치 방법 8](#_Toc43103666)

# 개요

**저희 팀은 개발을 진행하기 앞서 개발 진행을 계획을 세웠습니다.**

**첫 번째**로 ncurses라이브러리를 사용하여 기본적인 게임의 화면 예를 들어 각 게임스테이지, 미션창, 점수창 등을 구성과 디자인합니다.

**두 번째**로 뱀이 스테이지에 나타나고 움직이는 것을 구현합니다.

**세 번쨰**로 게임 내 요소인 아이템과 게이트의 등장을 구현합니다.

**네 번째**로 뱀이 아이템을 먹고 게이트를 통과할 때 나타나는 뱀의 변화와 점수를 구현합니다.

**다섯 번째**로 각 스테이지의 미션을 완료 후 다음 스테이지로 넘어가는 것을 구현합니다.

**마지막**으로 기타 오류와 버그를 수정 또는 보완합니다.

개발하는 동안 구글 드라이브를 통해 각자 구현한 파일을 공유하며 협업을 하였습니다. **snakegame1.0**부터 **snakegame1.1.5**까지 버전을 만들어 버전 관리를 하였고 github의 commit처럼 오픈채팅을 통해 각자가 수정하고 구현한 내용 또는 버그나 오류 내용 등을 해당 버전을 드라이브에 업로드함과 동시 실시간으로 공유하였습니다.

snakegame을 개발하면서 사용한 외부 라이브러리는 ~~ncurses입니다. 리눅스 우분투 개발환경에서 사용하였으며 해당 라이브러리의 획득 및 설치 방법은 터미널에~~ **~~「sudo apt-get install libncurses5-dev libncursesw5-dev~~**~~」 입력하여 설치하였습니다. 그 후 main.cpp파일에 #include <ncurses.h>를 작성하여 진행하였습니다. 또한 makefile을 활용하여 「~~**g++ -std=c++11 -o main main.cpp -lncursesw**~~」 을 컴파일 하여 실행하였습니다.~~

snakegame 구조는 map.h, game.h, gate.h의 3개의 헤더 파일과 main.cpp, map.cpp, gate.cpp, snake.cpp의 4개의 소스파일 그리고 makefile로 구성되어 있습니다.

map.h파일에서는 각 스테이지의 맵을 0과 1로 이루어진 2차원 배열 형태로 구성하며 각 스테이지를 배열를 통해 관리합니다. game.h 파일에서는 게임이 진행되는 main.cpp파일과 게임 화면을 출력하는 map.cpp파일의 초기 설정값과 변수, 함수들이 선언하여 관리합니다. gate.h는 gate.cpp의 헤더 파일로 뱀의 위치와 이동, 변화를 구현한 snake.cpp과 연결되어 게이트 통과 후 뱀의 위치 설정에 영향을 줍니다.

main.cpp에는 ~~ncurses, vector, time.h, cstdlib같은 라이브러리와 map.h,~~ map.cpp, gate.cpp, snake.cpp 파일 등이 include 됩니다. main.cpp의 game()함수에서는 map.cpp, gate.cpp, snake.cpp의 함수들을 이용하여 게임화면과 스테이지의 세팅, 뱀의 조작, 아이템과 게이트의 발생 등을 설정하고 관리하여 게임을 진행시킵니다. map.cpp파일에서 색깔, 배경, 스테이지, 점수와 미션 창, 초기화 등을 구현합니다. gate.cpp파일에서는 게이트의 발생 위치와 게이트 통과 후 뱀의 위치에 대해 설정합니다. snake.cpp 파일에서는 뱀의 위치, 이동, 아이템, 뱀의 몸 길이 변화 등을 구현하고 관리합니다.

# 개발 내용 및 결과물

## 목표

|  |
| --- |
| **작성요령 (10점)**  **프로젝트의 목표를 기술하세요. 각 단계별 목표를 구체적으로 쓰세요.** |

|  |  |  |
| --- | --- | --- |
| 적용단계 | 내용 | 적용 여부 |
| 1단계 | Map의 구현 | 적용 |
| 2단계 | Snake 표현 및 조작 | 적용 |
| 3단계 | Item 요소의 구현 | 적용 |
| 4단계 | Gate 요소의 구현 | 적용 |
| 5단계 | 점수 요소의 구현 | 적용 |

**프로젝트의 각 적용 단계별 구현 목표를 명확하게 제시한다. 제시한 권고안의 내용을 포함하여, 변경된 부분 등을 구체적으로 단계별 구현 목표를 작성해야 한다.**

**1 단계. Map의 구현**

**2 단계. Snake 표현 및 조작**

**3 단계. Item 요소의 구현**

**4 단계. Gate 요소의 구현**

**5 단계. 점수 요소의 구현**

## 개발 내용 및 결과물

### 개발 내용

|  |
| --- |
| **작성요령 (10점)**  **프로젝트의 수행의 내용을 구체적으로 기술한다. 세부 목표별로 어떤 결과를 어떤 방법으로 달성하였는지를 자세히 기술한다.** |

### 시스템 구조 및 설계도

|  |
| --- |
| **작성요령 (30점)**  **프로젝트의 각 세부 목표의 주요 기능(알고리즘 등)에 대해서 기술한다. 세부 목표별로 수정한 프로그램 소스 파일을 나열하고, 해당 파일에서 세부 목표를 달성하기 위해 작성한 클래스/함수에 대해 나열하고, 각 요소에 대해 간략한 설명을 작성한다. 또한 각 요소의 개발자를 명시한다.** |

### 활용/개발된 기술

|  |
| --- |
| **작성요령 (10점)**  **프로젝트 수행에 사용한 외부 기술/라이브러리를 나열하여 작성한다. 각각 기술을 이 프로젝트에 적용할 때, 도움 받거나 해결하고자 하는 기능에 대해 상세히 설명한다.**  **NCURSES / STL 라이브러리 등을 포함하여 설명한다.**  **또한, 이 프로젝트를 수행하면서, 새롭게 고안한 알고리즘 등이 있다면 설명한다.** |

### 현실적 제한 요소 및 그 해결 방안

|  |
| --- |
| **작성요령 (5점)**  **제안된 프로젝트의 단계 별 수행에 있어, 제한 요소를 찾아 작성한다. 해당 제한 요소를 해결하기 위해서 어떤 방법으로 해결하였는지 작성한다.** |

### 결과물 목록

**1) 게임구조**

|  |
| --- |
| ~~snakegame1.1.5~~  ~~├── Makefile~~  ~~├── map.h~~  ~~├── game.h~~  ~~├── gate.h~~  ~~├── main.cpp~~  ~~├── map.cpp~~  ~~├── snake..cpp~~  ~~└── gate.cpp~~ |

**2) 파일 설명**

|  |  |
| --- | --- |
| **~~snakegame1.1.5~~** | |
| **파일** | **설명** |
| **~~Makefile~~** | make명령어를 통해 main 오브젝트 생성 |
| **~~map.h~~** | stage1,2,3의 맵 |
| **~~game.h~~** | map.cpp의 초기 설정 및 함수선언, snake 객체 구조체 선언 |
| **~~gate.h~~** | gate.cpp 헤더 파일 |
| **~~main.cpp~~** | 게임이 실행되는 메인 파일 |
| **~~map.cpp~~** | 게임 맵 및 화면 설정 및 출력 |
| **~~gate.cpp~~** | 게이트 위치 및 통과 관리 |
| **~~snake.cpp~~** | 뱀의 위치, 이동 및 길이, 아이템 관리 |

**3) 게임 진행 화면**

**(1) 게임 맵**

|  |  |  |
| --- | --- | --- |
| **스테이지1** | **스테이지2** | **스테이지3** |
|  |  |  |

**(2) 안내 화면**

|  |  |
| --- | --- |
| **스테이지 클리어** | **게임오버** |
|  |  |

# 자기평가

|  |
| --- |
| **작성요령 (5점)**  **프로젝트를 수행한 자기 평가를 서술한다. 팀원 개개인의 자기 평가가 포함되어야 하며, 본인의 역할, 프로젝트 수행 시 어려운 점, 도움이 되었던 점, 이 프로젝트 운영에 개선이 필요하다고 생각하는 점을 충분히 서술한다.** |

**이지형:**

**이재원:**

**정호용:**

# 참고 문헌

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 번호 | 종류 | 제목 | 출처 | 발행년도 | 저자 | 기타 |
|  | 서적 |  |  |  |  |  |
|  | 기사 |  |  |  |  |  |

# 부록

## 사용자 매뉴얼

**1) 실행: 터미널에서 snakegame1.1.5 파일로 이동후 ./main 명령어 실행합니다.**

**2) 실행과 동시에 스테이지 1 시작합니다.**

**3) 뱀의 키보드 방향키로 움직입니다.**

**3) 초록색 아이템을 먹으면 + 1개 획득하고 뱀의 길이는 1증가합니다.**

**4) 빨간색 아이템을 먹으면 - 1개 획득하고 뱀의 길이는 1 감소합니다.**

**5) 분홍색 게이트가 게임 시작 10초 후 생성 이후 10초마다 위치 갱신, 분홍색 게이트 통과 후 G 1개 획득합니다.**

**6) 분홍색 게이트는 한 쌍으로 한 쪽 게이트로 들어가면 다른 쪽 게이트로 나옵니다.**

**7) Mission 화면의 조건을 달성하면 다음 스테이지로 이동합니다. 스테이지는 총 3개가 있고 모두 클리어시 승리**

**8) Score 항목 설명 B: 뱀의 길이, +: 초록색 아이템 점수, -:빨간색 아이템 점수, G:게이트 통과 점수입니다.**

**9) 뱀의 진행방향과 반대방향 입력 시, 뱀의 머리가 벽에 닿을 시, 미션 충족을 못 할 시 게임오버됩니다.**

**10) f1을 누르면 게임을 중간에 종료할 수 있습니다.**

## 설치 방법

**터미널에서 snakegame1.1.5 파일로 이동 후 make 명령어 입력**

**g++ -std=c++11 -o main main.cpp -lncursesw 실행 후 main파일 생성**