

# F 학점 피하기 게임

Dodge the F Grade

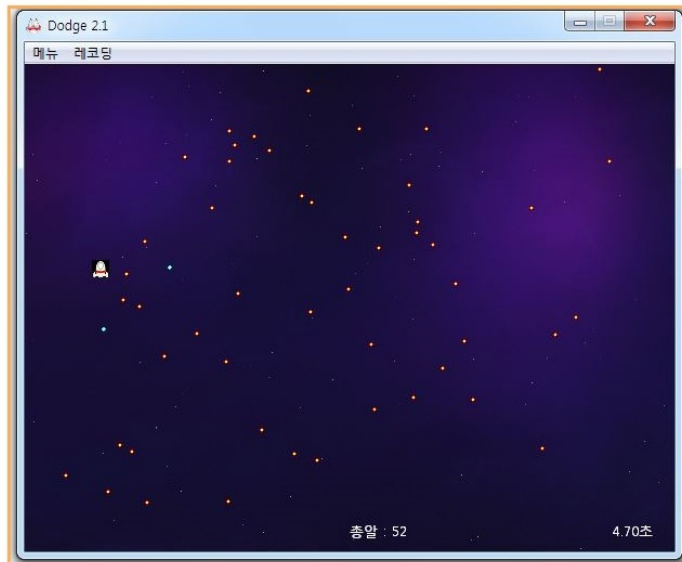
**Sang-Min Lee, Jin-Hyeok Yoo, Soon-Ho An**

Dept. of Computer Science and Engineering  
Kwangwoon University  
Seoul, S. Korea, Republic of  
E-mail : knowledgein@naver.com

Adviser : Prof. Kang-Hoon Lee<sup>a</sup>

## 1. 프로젝트 개요

저희는 이번 학기 프로젝트를 통해 새롭게 배운 프로그래밍 언어인 C++를 이용해서 만들 수 있는 프로그램에 대해 생각해보게 되었습니다. 그러던 중 프로젝트의 난이도를 고려하여 콘솔에만 구현 가능한 수준의 여러 가지 게임들에 대해 알아보았고 흔히 닥지 게임이라고 불리는 피하기 게임을 구현해보기로 하였습니다.



▲Figure 1 플레이어 내장 게임인 ‘Dodge’

저희는 여기에 대학생으로서 공감할 수 있는 내용으로 날아오는 총알이 F 학점이라면 어떨까 생각하게 되었고, 2 가지 모드로 F 학점을 피해 한 학기 동안 뿌려지는 A, B, C, D 학점을 모으는 모드와 무한히 F 만 뿌려지는 모드를 구상했습니다.

## 2. 역할 분담

**2014726037 이상민 (40%)**

전체적인 코드 디자인, Cursor 클래스 디자인, loadGame 함수와 endGame 함수 구현

**2014707040 유진혁 (30%)**

F 클래스와 Grade 클래스 디자인, 중간보고서 작성

**2013610017 안순호 (30%)**

타이틀 함수 및 전체적인 UI 디자인, 제안서 및 최종보고서 작성

### 3. 진행 일정

기본적으로 매주 화요일 중앙도서관 집현실에서 2 시간동안 한 주 동안 짠 프로그램 진행상황을 서로 확인하고 방향을 잡는 회의를 가졌습니다. 아래는 그 동안의 모든 회의 내역입니다.

3.21(화) 14:00~16:00

학기 프로젝트 첫 회의. 작품 구상 및 필요사항, 구현 가능성 및 제안서 내용에 대해 회의. 구현 가능할 게임으로 닳지, 테트리스, 로그라이크 등을 후보군으로 올렸으며 최종적으로는 닳지 게임을 만들기로 함.

3.28(화) 19:00~21:00

닳지 게임의 세부 구현사항 회의. 게임의 2 가지 모드에 대해 어떤 방식으로 구현하게 될지에 대해 자세히 논의. 제안서 방향 및 내용 정함.

4.1(토) 15:00~17:00

제안서 완성 및 자료조사. 게임을 실질적으로 구현하기 위한 여러 가지 예시 코드 수집 및 공유.

4.6(목) 17:00~19:00

중간고사 전 마지막 회의. 기본적인 게임판 및 사용자 입력을 받아 콘솔에서 움직이는 오브젝트 구현. 종료 함수 및 스코어 구현 방식에 대해 논의.

4.25(화) 18:00~20:00

전날 수업 시간에 교수님께서 보여주신 다른 조의 제안서들에 대해 논의하며 프로젝트 방향에 대해 회의. 자동으로 움직이는 오브젝트(F)를 구현하려 했으나 실패함.

5.2(화) 18:00~20:00

게임 판을 배열로 전환. 수업시간에 새로 배운 내용인 클래스를 활용하여 오브젝트를 다시 구현.

5.5(금) 18:00~20:00

움직이는 오브젝트를 구현하였으나 여러 가지 오류가 발생하고 입력과 충돌하는 문제 발생.

5.11(목) 19:00~21:00

F 를 구현하는 방식에 대해 계속 논의. 구현이 잘 되지 않아 프로젝트의 난이도를 조정하는 방향도 논의 되었으나 닳게 다음날 한 조원이 구현 성공. 그 후 토요일 중간보고서 제출.

5.17(수) 수업시간

교수님께서 질의 응답 시간을 가져주셔서 짜고 있는 코드에 대한 조언을 받음. thread 함수의 위험성과 움직이는 오브젝트에 대해 결정적인 조언을 얻음.

5.23(화) 18:00~20:00

오브젝트를 교수님의 조언에 따라 다시 구성. 전체적인 프로그램을 다시 처음부터 만들어나감.

5.27(토) 09:00~13:00

대규모 업데이트. 학점 모으는 모드를 거의 구현해냄. F class 와 Grade class, Cursor class 의 대부분 내용 구현.

6.3(토) 09:00~13:00

무제한 모드 추가, 학점 계산방법을 도입했으나 몇 가지 오류 발생.

6.5(월) 수업시간

교수님께 피드백 받음. 마지막 회의 및 프로그램 추가 수정사항 논의. 이후 업데이트는 각자 구현하여 온라인으로 공유하고 논의함.

## 4. 구현 세부사항

아래는 코드에 자세히 각주로 표현한 내용을 전체적으로 정리해 놓은 글입니다.

게임 플레이 공간은 배열로 지정하였습니다. 배열의 element 값 상수를 바꿔가며 해당 값에 따라 draw 함수에서 화면을 출력하는 방식입니다. draw 함수의 performance 증가를 위해 이전 값과 비교해서 달라진 값만 출력하는 기법을 이용하였습니다.

게임은 두 가지 모드로 나누어져 있으며 loadGame 함수의 parameter 로 이를 결정합니다.

코드는 main.cpp 를 비롯해 4 개의 cpp 파일과 4 개의 헤더 파일로 구성돼 있으며 각각 다음과 같은 내용을 담고 있습니다.

Cursor.cpp, Cursor.h : Cursor 클래스를 지정하여 사용자가 움직이는 객체를 구현하였습니다. 사용자의 입력에 따라 객체를 이동시키는 move 함수와 Grade 를 먹었을 때 학점이 계산되도록 하는 calcGrade 함수, 그리고 학점 평량 평균과 전체학점에 대한 getter 로 이루어져 있습니다.

Grade.cpp, Grade.h : Grade 클래스로 학점 모드에서 맵에 학점을 뿌려주는 역할입니다. Appear 함수를 통해서 학점을 뿌려주게 됩니다. rand 함수를 이용해 랜덤한 위치에 랜덤한 학점이 뿌려지도록 구현했습니다.

F.cpp, F.h : F 클래스로 맵에 저절로 움직이는 오브젝트인 F 를 구현했습니다. F 를 발사 가능한 상태로 만들어주는 set 함수와 F 의 움직임을 제어하는 move 함수와 moveStart 함수 그리고 moveEnd 함수가 있습니다.

게임이 종료된 후 endGame 함수는 게임의 모드와 종료 상태에 따른 parameter 값을 따라 종료 문구를

출력합니다. 이후 restart 변수 값을 입력 받아 재시작 여부를 결정하는 방식으로 구현했습니다.

## 구현 시 특히 어려웠던 부분들

### 움직이는 오브젝트 (F)의 구현

게임의 핵심이라 할 수 있는 움직이는 오브젝트의 구현에 꽤 어려움을 겪었습니다. 스레드를 이용하여 커서의 움직임을 제어하려 해봤으나 계속 사용자의 입력을 받아 움직이는 플레이어 커서 오브젝트 (게임에서 ◇에 해당) 와 커서 제어의 충돌이 일어나서 엉뚱한 출력이 반복되는 등 구현에 애로사항이 가장 많았던 부분입니다. 5 월 17 일 수업시간에 교수님께서 질의 응답을 받는 시간을 만들어주셔서 코드에 대한 조언을 받았는데, 여기서 thread 함수를 포기하고 F 를 한칸 씩 움직이게 하는 함수를 반복문으로 제어해보라는 결정적인 조언을 얻게 되었습니다. 이후 5 월 27 일 토요일 4 시간에 걸쳐 완벽하게 오브젝트 구현에 성공했습니다.

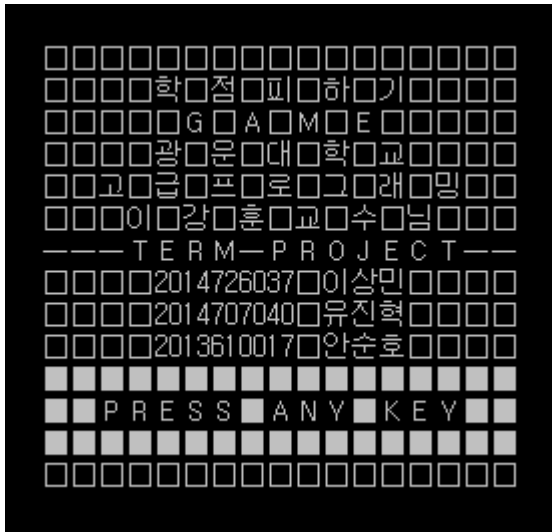
### 학점 계산 결과 출력

학점 모드에서 플레이어가 학점을 먹을 경우 변수를 조정하여 학점을 계산하는 함수를 만들었으나, 출력 과정에서 이상하게도 계산 오류가 발생했습니다. 이에 대한 원인을 알아보기 위해 여러 가지 방법을 사용했으나 결국 원인을 찾지는 못했습니다. 하지만 해결 방안으로 학기초 수업 시간에 배운 magic formula 로 소숫점을 지정해주자 문제가 사라졌습니다.

```
cout.setf( ios::fixed);  
cout.setf( ios::showpoint);  
cout.precision(2);
```

▲Figure 2. 학기 초에 배운 Magic Formula 를 적용한 모습.

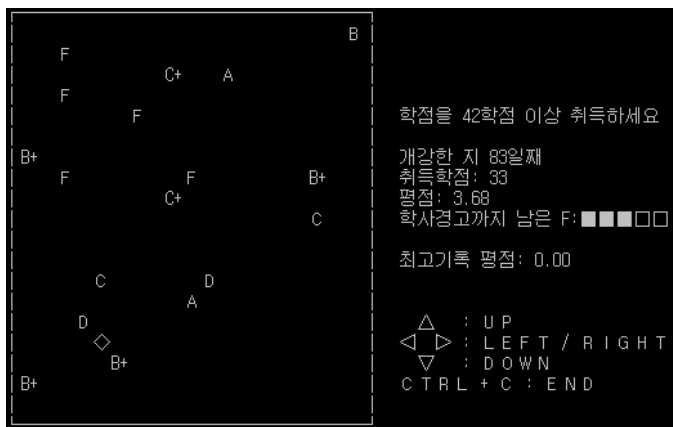
## 5. 최종 결과



▲Figure 3. 시작 화면



▲Figure 4. 무제한 모드 플레이 화면



▲Figure 5. 학점 모드 플레이 화면