

[오픈소스SW개론] 과제3

Snake Game OSS 개발 계획서

-

2019.03.26 ~ 2019.05.03

소프트웨어학과 17011654 김경남 (kkyy0126@naver.com)
소프트웨어학과 17011670 진수빈 (subinjin22@gmail.com)

목차

1. 프로젝트 개요	2
1-1 프로젝트 주제	
1-2 주제 선정 이유	
1-3 프로젝트 목표	
1-4 프로젝트 범위	
1-5 프로젝트 환경	
2. Snake Game OSS	6
2-1 선택한 Snake Game 오픈소스 소개	
2-2 다운로드 및 출처	
2-3 라이선스	
3. 프로젝트 수행 내용	10
3-1 코드 작성방법 및 버전 기준 정하기	
3-2 찾은 오픈소스 코드를 가독성 기준으로 clean 코드로 개선하기	
3-3 찾은 오픈소스 코드의 오류 개선하기	
3-4 개선하는 작업을 Git과 GitHub를 통해 수행하기	
3-5 활용할 수 있는 GitHub 기능	
4. 역할 분담	17
4-1 코드 분석, 개선, 기능추가 역할 분담	
5. 추진 방법 및 수행 일정	18
6. 회의록	20
7. 참고문헌 및 사이트	26

1. 프로젝트 개요

1-1. 프로젝트 주제

해당 프로젝트는 오픈소스를 활용한 프로젝트로, 하나의 오픈소스를 복제 및 수정하여 2차파생물을 일정 규칙에 준수하여 도출해 내는 데에 의의가 있다. 프로젝트의 진행을 위해 선별된 주제는 Snake Game으로, 대표적인 고전 아케이드 게임이다. Snake Game은 화면에 나타나는 뱀(사용자의 캐릭터)을 방향키로 움직여 랜덤위치에 생성된 먹이를 먹으며 뱀의 크기를 늘려나간다는 방식을 기본으로 한다.



1-2. 주제 선정 이유

다양한 게임들 중 Snake Game을 주제로 선택한 이유는, 남녀노소 누구나 이해하기 쉬운 룰로 간단하고 빠른 플레이가 가능하고, 이로 인해 다양한 변형이 가능하기 때문이다. 또한 오래된 고전 게임이기 때문에 그동안 축적된 여러 Snake Game의 오픈소스들을 참고하여 효과적인 구현 방식이나 적합한 자료구조 등을 적용시켜 볼 수 있다는 점에서 Snake Game을 주제로 선정하게 되었다.

1-3. 프로젝트 목표

해당 프로젝트의 목표는 코드 실행에 있어 오류를 최소화하여 게임의 버그를 줄이는 것과 새로운 기능을 추가하여 사용자가 Snake Game을 좀 더 재미있고 신선하게 플레이 하도록 하는 것이다. 또한 코드를 가독성의 측면에서 재검토 및 수정하여 재사용성이 높은 소스 코드로 완성시키는 것 역시 목표로 삼고있다.

1-4. 프로젝트 범위

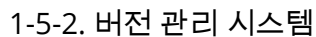
소스코드를 가독성의 측면에서 수정하고, 이식성 향상을 위한 코드로 변경하여 코드의 재사용성을 높이는 것까지 진행한다. 또한 해당 프로젝트는 오픈소스 기반이므로 GitHub를 통해 재배포하는 것 역시 프로젝트의 범위에 해당한다.

1-5. 프로젝트 환경

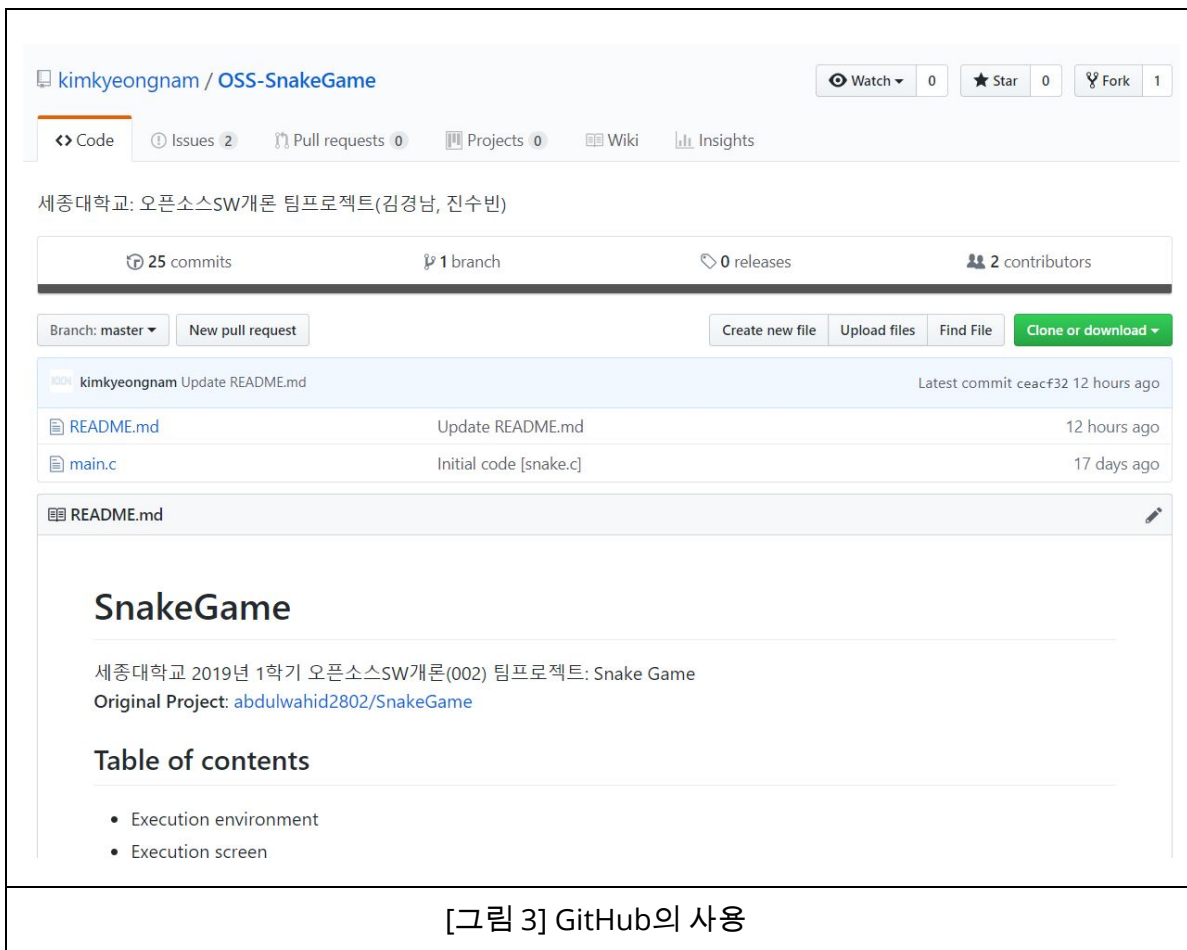
1-5-1. 로컬개발환경

해당 프로젝트는 Windows 10 환경에서 진행되었고, 언어는 C, 개발환경들은 Microsoft Visual Studio 2017을 사용하였다. Visual Studio 2017에서 코드를 컴파일 할 때, 컴파일

Warning Level은 '속성 > C/C++ > 일반 > 경고 수준' 에서 수정이 가능하다.



4



[그림 3] GitHub의 사용

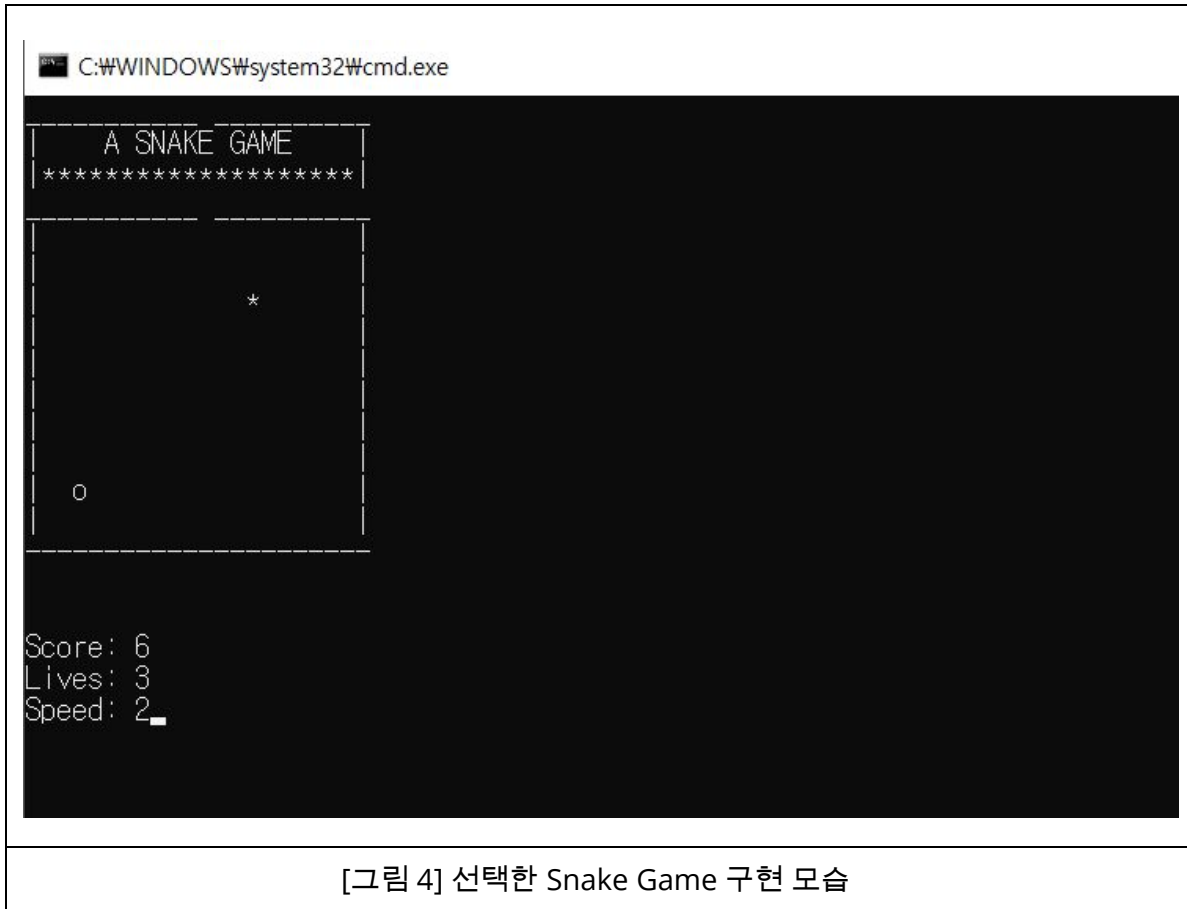
1-5-3. 라이선스

라이선스란 OSS개발자와 이용자 간에 사용방법 및 조건의 범위를 명시한 계약으로 OSS를 사용하기 위해서는 OSS개발자가 만들어 놓은 사용방법 및 조건의 범위에 따라야 한다. 우리가 사용할 Snake Game 오픈소스는 MIT라이선스를 따르고 있으며, MIT라이선스는 실행환경에서 반드시 준수해야하는 규칙은 따로 없고, 소스코드 재배포 시, 라이선스와 저작권 관련 명시를 지켜주면 된다.

2. Snake Game OSS

2-1. 선택한 Snake Game 오픈소스 소개

GitHub를 통해 Snake Game 오픈소스를 가져오게 되었고, 선택한 Snake Game은 C언어로 작성되어 [그림 4]와 같이 cmd창에서 구현된다.



실행과 동시에 뱀을 나타내는 '*'이 움직이며 사용자는 방향키를 눌러 먹이쪽으로 뱀('*')의 방향을 움직인다. 점수가 올라갈수록 뱀이 움직이는 속도(Speed)는 증가한다. 총 목숨 수는 3으로, 벽에 3번 이상 부딪히면 게임은 종료된다.

기존의 Snake Game은 먹이를 먹으면 뱀의 몸통 길이가 늘어나는게 눈에 보이지만, 선택한 Snake Game 오픈소스는 눈에 보이는 뱀의 길이는 늘어나지 않고, Score점수만 증가하여 먹이를 먹었다는 것을 표현한다.

2-2. 다운로드 및 출처

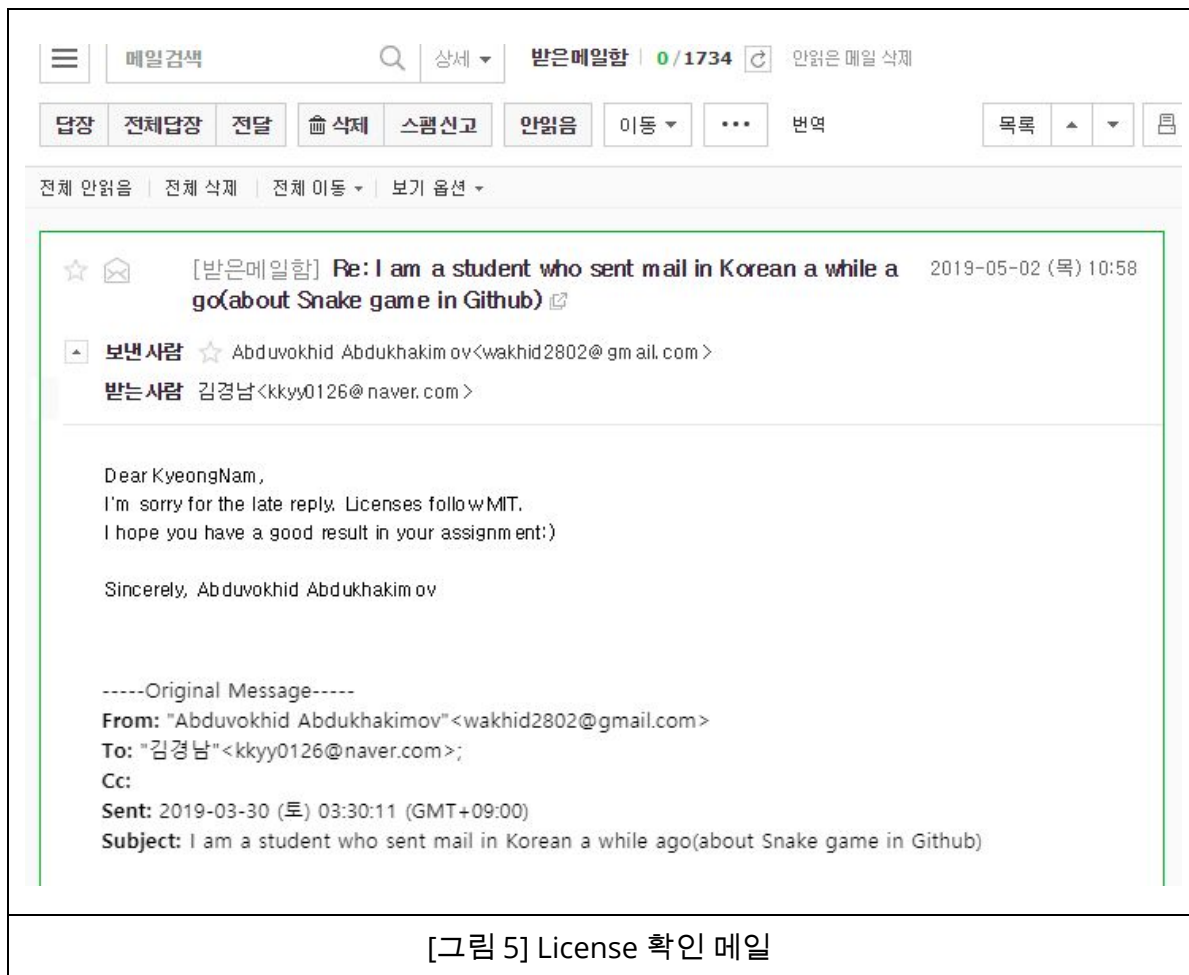
프로젝트를 위해 선택한 Snake Game 오픈소스의 원작자는 Abduvokhid Abdukhakimov (GitHub ID : abdulwahid2802)로, 원작자의 GitHub에서 Snake Game 오픈소스를 가져오게 되었다. 오픈소스를 가져와 코드의 복제 및 수정을 위해 원작자의 Snake Game 리포지토리를 포크(Fork) 한 후, Clone하여 팀장의 GitHub에 새 리포지토리를 생성하였다. 오픈소스 원본 출처와 생성된 새 리포지토리의 출처는 아래와 같다.

Snake Game 원작자(원본) 출처 : <https://github.com/abdulwahid2802/SnakeGame>

개발을 위한 새 리포지토리 출처 : <https://github.com/kimkyeongnam/OSS-SnakeGame>

2-3. 라이선스

원작자의 GitHub를 방문했을 때, Snake Game의 라이선스가 표시되어 있지 않아 원작자에게 메일을 보내 확인 해본 결과, 해당 Snake Game 오픈소스는 MIT라이선스를 따른다는 것을 알 수 있었다.



[그림 5] License 확인 메일

MIT 라이선스(MIT License)는 미국 매사추세츠 공과대학교(MIT)에서 해당 대학의 소프트웨어 공학도들을 돕기 위해 개발된 라이선스이다. MIT 라이선스는 SW의 소스코드를 공개하지 않아도 되는 대표적인 오픈소스SW 라이선스이다. 즉, MIT 라이선스를 따르는 소프트웨어를 개조한 제품을 반드시 오픈소스로 배포해야 한다는 규정이 없어 다른 엄격한 라이선스들에 비해 자유로운 편이다. MIT 라이선스의 준수사항은 다음과 같다.

- SW에 대한 자유로운 사용, 복제, 배포 허용
- SW를 배포하는 경우, 저작권을 반드시 표시
- SW를 배포하는 경우, 라이선스 사본을 반드시 첨부
- SW를 배포하는 경우, 보증의 부인과 책임의 제한에 대한 내용을 반드시 표시

- 수정 프로그램에 대한 소스코드의 공개를 요구하지 않으므로 상용 SW에 무제한 사용가능
- 소스코드 재배포를 위해서는 저작권 표시와 문구, 면책조항을 유지하여야 함
- MIT 라이선스에 따른 프로그램을 바이너리 형태로 재배포 하는 경우에는 소스코드를 배포하는 경우와 마찬가지로 배포판과 함께 제공되는 문서 또는 그 밖의 매체에 저작권 표시와 문구, 면책조항을 담아야 함.
- 공개 범위 : MIT 라이선스는 의무사항만 준수한다면 소스코드를 공개하지 않아도 됨

3. 프로젝트 수행 내용

3-1. 코드 작성방법 및 버전 기준 정하기

3-1-1. 변수 작성 방법

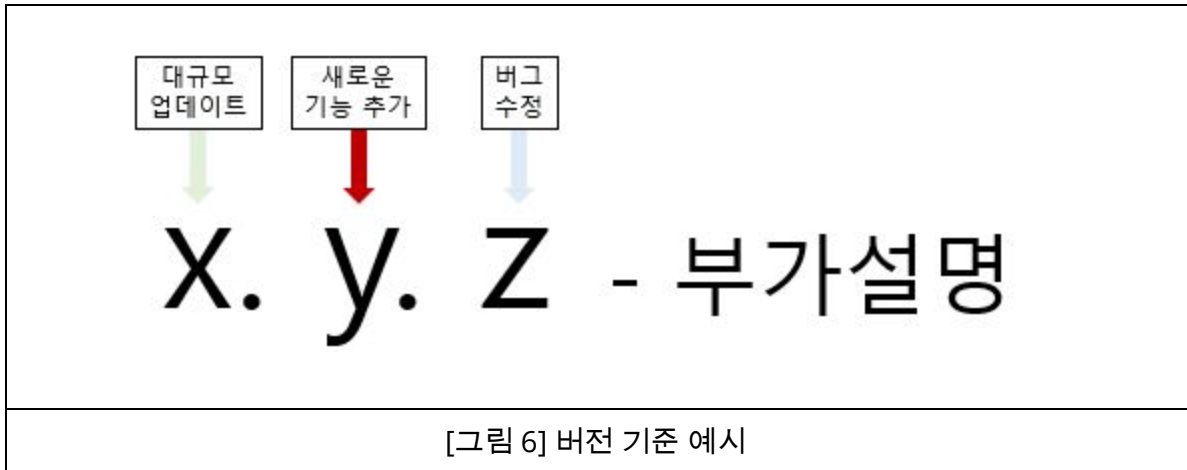
- 변수는 한 목적으로만 사용하기
- 변수는 항상 소문자로 시작하고, 다음 단어는 대문자로 구분하기
- 함수의 매개변수 역시 그 의미를 명확히 표현하는 단어로 작성하기

3-1-2. 함수 작성 방법

- 함수명은 항상 대문자로 시작하고, 다음 단어는 대문자로 구분하기
- 변수명과 함수명을 같게 하지 않기
- 함수의 접두사 공통으로 사용하기

함수명	설명
Avr	평균값
Cnt	데이터의 개수를 얻을 때
Chk	어떤 값을 검증할 때
Get	어떤 값을 얻어낼 때
Set	값을 설정할 때
Is	무엇인가 물을 때
Key	데이터 중 키값만 얻을 때
Max	최대값
Mid	중간값
Min	최소값

3-1-3. 버전 표기 기준



- x: 기존 버전과 호환되지 않게 바뀔 시(대규모 업데이트)
- y: 기존 버전과 호환되면서 새로운 기능을 추가할 때
 - InitMap(): 0
 - DrawMap(): 1
 - GetInput(): 2
 - EatFood(): 3
 - Die(): 4
 - main(): 5

추후 추가되는 함수는 순차적으로 번호 부여

- z: 기존 버전과 호환되면서 버그 수정했을 시
 - 수정하지 않을 시: 0
 - 수정할 시: 1
- 부가설명: 기여한 부분 설명

예) GetInput() 버그수정 및 새로운 기능(Enter를 누를 시 게임 일시정지) 추가.

기존 버전과 호환됨: 0.2.1-enter

3-2. 찾은 오픈소스 코드를 가독성 기준으로 clean코드로 개선하기

3-2-1. 전역변수의 사용을 줄이기

```
static char HEADER[4][25] =  
{  
    "-----",  
    "|   A SNAKE GAME   |",  
    "|*****|",  
    "-----",  
};  
static char FOOTER[25] =  
{  
    "-----",  
};  
static char MAP[15][25] =  
{  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
    " |           |",  
};
```

[그림 7] 전역변수 사용의 예

다음과 같이 하나의 함수 내에서만 쓰이는 전역 변수를 해당 함수 내에 넣어주어 지역변수로 변환하여 준다.

3-2-2. 같은 의미를 갖는 변수들 구조체로 선언하기

```
static int Xpos;  
static int Ypos;  
static int Xfood;  
static int Yfood;  
static int direction = 80;  
static int hits = 0;  
static int wait = 500;  
static int lives = 3;  
static int speed = 1;
```

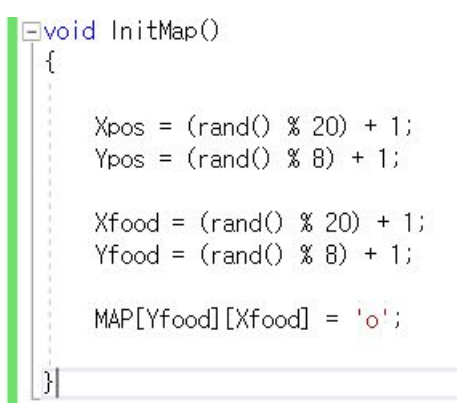
[그림 8] 선언이 많은 변수들의 예

Xpos변수, Ypos변수, direction,lives, speed변수들은 모두 뱀과 관련한 변수들이기 때문에 Snake구조체를 만들어 묶어주고, 마찬가지로 Xfood변수와 Yfood 변수는 Food 구조체로 묶어준다.

3-2-3. 하나의 함수에는 하나의 기능을 하도록 설계하기

main.c 코드의 DrawMap()함수에서는 맵을 그리는 기능과 더불어 사용자의 입력에 따른 위치 변화의 기능까지 포함하고 있기 때문에 각각의 기능을 다른 함수로 표현하여 준다.

3-2-4. 매직넘버 사용하기



```
void InitMap()
{
    Xpos = (rand() % 20) + 1;
    Ypos = (rand() % 8) + 1;

    Xfood = (rand() % 20) + 1;
    Yfood = (rand() % 8) + 1;

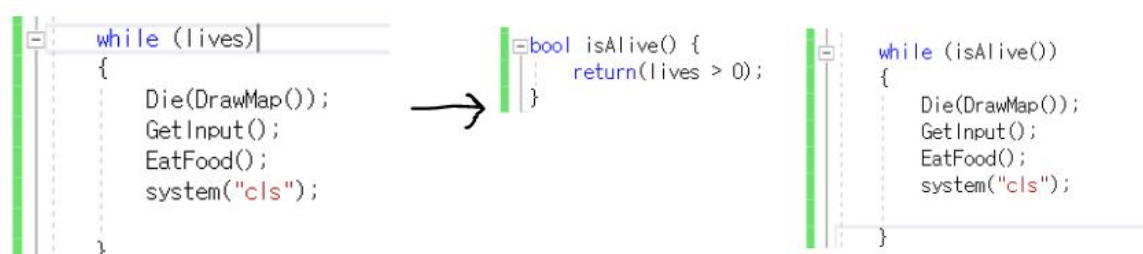
    MAP[Yfood][Xfood] = 'o';
}
```

[그림 9] 상수에 의미가 보이지 않게 사용되는 예

[그림9]를 보면 20과 8이 무슨 수인지를 알 수 없다. 하지만 8은 아래쪽 벽면을, 20은 오른쪽 벽면을 나타내므로 이를 매직넘버화를 통해 상수의 가독성을 높여준다.

3-2-5. 조건식의 함수화

조건식을 변수나 비교연산자로 표현하기 보단 함수로 표현하여 코드의 가독성을 높인다.



```
while (lives)
{
    Die(DrawMap());
    GetInput();
    EatFood();
    system("cls");
}

bool isAlive() {
    return(lives > 0);
}

while (isAlive())
{
    Die(DrawMap());
    GetInput();
    EatFood();
    system("cls");
}
```

[그림 10] 조건식의 함수화 예시

3-3. 찾은 오픈소스 코드의 오류 개선하기

3-3-1. 컴파일 환경 Warning Level 오류 개선

```
clde#10.0.16299.0#wum#winioclt.h(7548): warning C4668: '_WIN32_WINNT_WIN10_RS1'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7553): warning C4668: '_WIN32_WINNT_WIN10_TH2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7557): warning C4668: '_WIN32_WINNT_WIN10_TH2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7564): warning C4668: '_WIN32_WINNT_WIN10_RS2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7574): warning C4668: '_WIN32_WINNT_WIN10_RS2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7578): warning C4668: '_WIN32_WINNT_WIN10_TH2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7583): warning C4668: '_WIN32_WINNT_WIN10_RS3'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(7777): warning C4668: '_WIN32_WINNT_WIN10_RS2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(12232): warning C4668: '_WIN32_WINNT_WIN10_RS2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(12249): warning C4668: '_WIN32_WINNT_WIN10_RS3'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(12266): warning C4668: '_WIN32_WINNT_WIN10_RS3'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(12285): warning C4668: '_WIN32_WINNT_WIN10_RS2'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wum#winioclt.h(12404): warning C4668: '_WIN32_WINNT_WIN10_RS3'은(는) 전처리기 매크로로 정의되어 있지 않음
clde#10.0.16299.0#wucrt#time.h(35): warning C4820: '_timespec64': '4'바이트 채움 문자가 데이터 멤버 'tv_nsec' 뒤에 추가되었음
clde#10.0.16299.0#wucrt#time.h(42): warning C4820: '_timespec': '4'바이트 채움 문자가 데이터 멤버 'tv_nsec' 뒤에 추가되었음
스.c(52): warning C4255: 'InitMap': 함수 프로토타입을 입력하지 않았습니다. '()'에서 '(void)'로 변환됩니다.
스.c(64): warning C4255: 'DrawMap': 함수 프로토타입을 입력하지 않았습니다. '()'에서 '(void)'로 변환됩니다.
스.c(117): warning C4255: 'GetInput': 함수 프로토타입을 입력하지 않았습니다. '()'에서 '(void)'로 변환됩니다.
스.c(126): warning C4255: 'EatFood': 함수 프로토타입을 입력하지 않았습니다. '()'에서 '(void)'로 변환됩니다.
스.c(170): warning C4255: 'main': 함수 프로토타입을 입력하지 않았습니다. '()'에서 '(void)'로 변환됩니다.
스.c(173): warning C4242: '함수': 'time_t'에서 'unsigned int'(으)로 변환하면서 데이터가 손실될 수 있습니다.
clde#10.0.16299.0#wucrt#stdio.h(946): warning C4710: 'int printf(const char *const ,...)': 함수를 인라인화하지 못했습니다.
clde#10.0.16299.0#wucrt#stdio.h(946): note: 'printf' 선언을 참조하십시오.
스.c(114): warning C4715: 'DrawMap': 모든 제어 경로에서 값을 반환하지는 않습니다.
ce#repos#snakeGame#Debug#snakeGame.exe
```

[그림 11] Warning Level 빌드 오류

Warning Level을 최상으로 했을 때의 빌드 오류를 수정하여 준다.

3-3-2. 조건문 else문 오류 개선

조건문을 작성할 때 if문만 사용하고 else문을 써주지 않아 오류처리에 실패할 경우가 있기 때문에 항상 if문을 써주면 else문을 반드시 써서 오류처리를 해준다.

3-3-3. 반복문 return 오류 개선

조건문과 마찬가지로 반복문에서도 항상 return을 잘 활용하여 무한루프에 빠지지 않게 하고, 함수일 경우 return값을 잘 정의해 주어야 한다.

3-3-4. 헤더파일 작성 방식 오류

해당 Snake Game 오픈소스 코드는 헤더파일 작성 방식에 오류가 있다. 헤더파일을 include할 때, 사용자 지정 헤더가 아니므로 ""(큰따옴표)가 아닌 <>로 작성해 주어야 한다.

```
#include "stdlib.h"  
#include "string.h"  
#include "conio.h"  
#include "windows.h"  
#include "time.h"
```

[그림 12] 헤더파일 작성 오류

3-4. 개선하는 작업을 Git과 GitHub를 통해 수행하기

3-4-1. Git

Git은 리눅스 커널 개발자 리누스 토발스(Linus Torvalds)가 직접 만든 버전관리를 위한 오픈소스 툴이다. 특히 프로젝트를 진행하거나 종료 후 유지보수가 필요한 부분에서 소스관리를 체계적으로 할 수 있도록 도와준다.

해당 프로젝트는 Git Bash를 통해 pull, commit, push, branch, tag, merge, rebase 등 Git명령어를 사용하여 코드 개선 및 기능 추가를 할 예정이다.

3-4-2. GitHub

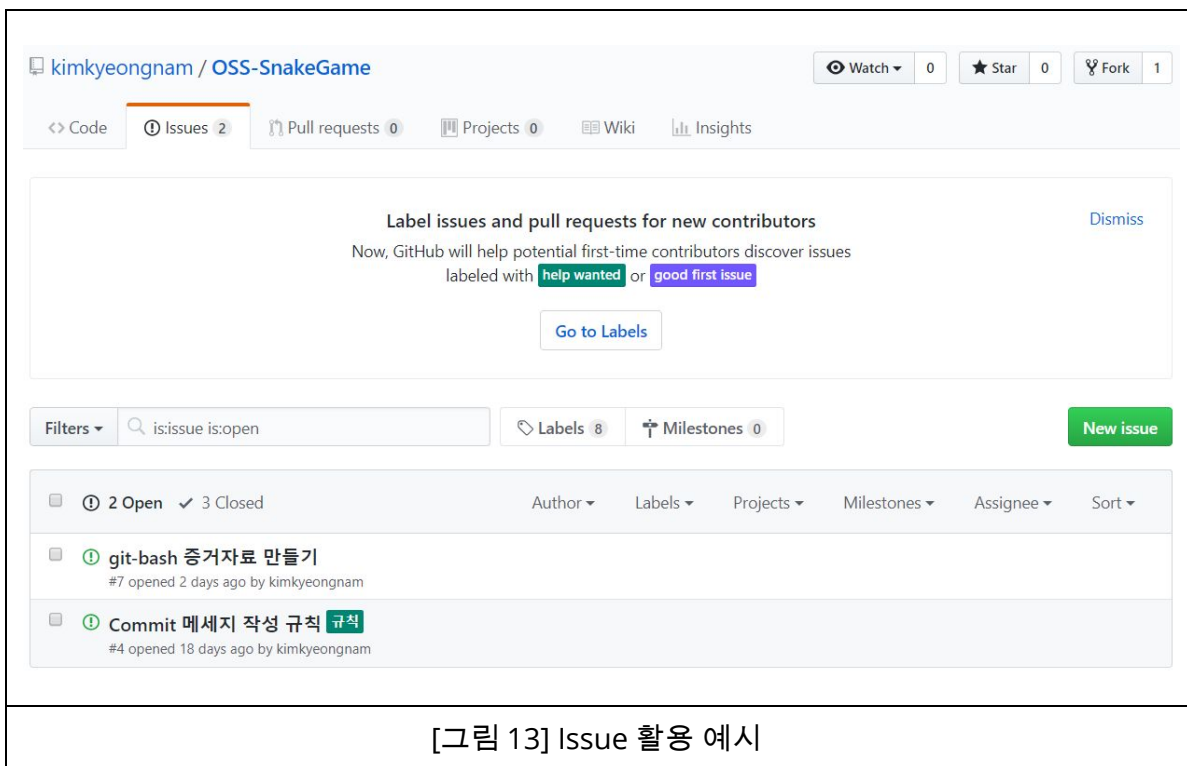
GitHub는 Git 프로그램을 이용해 로컬PC에서 원격 서버에 저장되어 있는 최신 프로젝트 소스를 다운 받거나, 자신이 수정한 소스 내용을 저장할 수 있는, 가장 유명한 웹 기반 Git Solution 중 하나이다.

GitHub를 이용하여 팀장은 찾은 OSS 저장소를 fork 및 자신의 로컬저장소로 clone한다. 이후 clone된 저장소를 새로운 directory를 생성하여 복사하고, GitHub에 새로운 원격저장소 생성 후 새로운 directory와 연결하여 해당 directory 내용을 push한다. 새로운 원격저장소에 팀원들을 추가하고, 각 팀원은 이 저장소를 fork 및 clone을 통해 코드를 관리한다. 또한 각 팀원들은 자신의 저장소에서 수정한 내용을 pull request하여 다른 협업자들이 수정 내용을 확인할 수 있도록 한다.

3-5. 활용할 수 있는 GitHub 기능

3-5-1. Issue 활용하기

Issue를 활용하여 팀원간의 지켜야 할 규칙이나 알아두어야 할 점을 기록한다.



4. 역할 분담

4-1. 코드분석, 개선, 기능추가 역할 분담

이름	코드 분석 (함수)	코드 개선 및 수정	기능 추가	보고서 작성 및 발표 준비
김경남	InitMap() DrawMap() GetInput()		필요한 기능 추가	각각의 목록 보고서 작성 및 발표 준비
진수빈	EatFood() Die() main()			

선택한 Snake Game 코드는 총 6개의 함수(InitMap, DrawMap, GetInput, EatFood, Die, main)로 구성되어 있고, 해당 함수들 중 관계가 있는 함수들로 묶고 둘로 나눠 코드를 분석하였다.

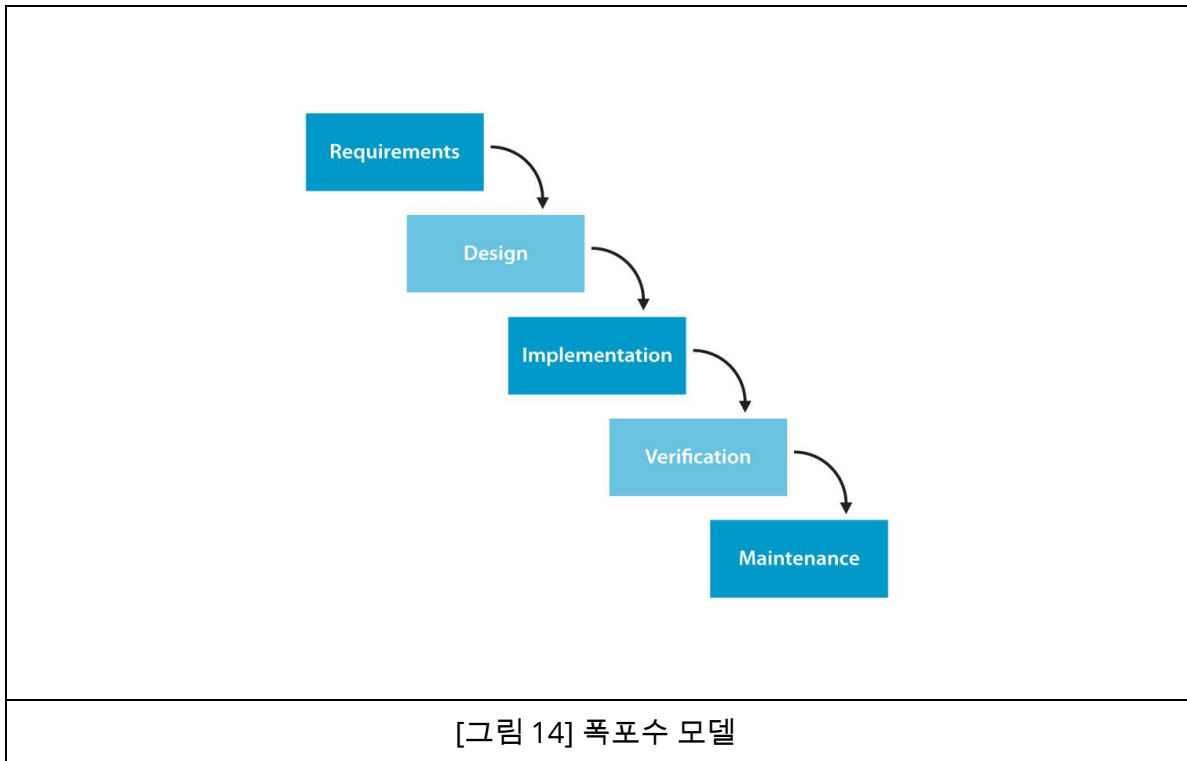
분석 이후 진행될 코드 개선 및 수정 역시 각자 분석했던 코드를 바탕으로 프로젝트 수행 내용에 맞춰 개발하도록 할 것이다.

보고서 및 발표자료 준비는 매주 수요일마다 정기회의를 통해 서로의 진도 및 해야할 목록을 피드백하여 과제 기여도 균형을 유지하도록 하며, 각각의 목차에 대해 서로의 생각을 정리하고 작성하도록 하였다.

5. 추진 방법 및 수행 일정

5-1. 추진 방법

효율적이고 효과적인 프로젝트 결과물 산출을 위해 소프트웨어 개발 생명주기(Software Development Life Cycle, SDLC)모델 중 폭포수 모델(Waterfall Model)을 통해 프로젝트를 추진한다.



폭포수 모델은 분석, 설계, 구현, 확인(테스트), 유지보수의 과정을 순차적으로 진행하는 모델로 Snake Game 오픈소스 개발 단계에 적합하다. 기존의 Snake Game 오픈소스 코드를 분석하고, 새로운 기능이나 수정이 필요한 부분을 다시 설계하여 구현하고 유지보수를 반복하는 것이다.

모델에 맞춰 개발 계획을 미리 세워 놓고 프로젝트를 추진해나갈 것이다. 수행일정은 다음과 같다.

5-2. 수행 일정

날짜	3.25 ~ 3.31	4.1 ~ 4.7	4.8 ~ 4.14	4.15 ~ 4.19	4.22 ~ 4.28	4.29 ~ 5.5	5.6 ~ 5.12	5.13 ~ 5.19	5.20 ~ 5.26	5.27 ~ 5.31
요구 분석					중간 고사					
설계										
구현										
테스트										
유지 보수										

6. 회의록

일시	2019. 03. 27 16:30 ~ 18:00	장소	센B112
작성자	김경남	작성일	2019. 03. 30
참석자	김경남, 진수빈		
안건	오픈소스 게임별 특징 찾기 및 정기 회의 날짜 결정		

■ 회의내용

- 회의내용
 - [오픈소스] 오픈소스 각 주제별 특징 찾기
 - [정기회의] 정기 회의 날짜 결정
- 결정사항
 - [보고서] 과제 3번 양식 작성 완료(3/27)
 - [오픈소스] 다음 회의까지 주제별 게임 특징 정보 수집 및 GitHub 코드 검색
 - [정기회의] 매주 수요일 16:30 - 18:30에 진행하기로 결정
이외의 회의는 상시에 진행
- 향후일정
 - [정기회의] 2019.03.27 센B112에서 정기 회의 진행 예정

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
[오픈소스] Tetris game, Car game, Pacman game, Snake game, Tic Tac Toe game의 게임별 특징 정보 수집 및 GitHub 코드 검색	김경남	~ 04.10	회의록 작성
[오픈소스] 2048 game, Bricks game, Chess for two humans, Mine Sweeper game, Egg game의 게임별 특징 정보 수집 및 GitHub 코드 검색	진수빈	~ 04.10	

일시	2019. 04. 03 16:30 ~ 18:30	장소	센B112
작성자	김경남	작성일	2019. 04. 03
참석자	김경남, 진수빈		
안건	프로젝트 일정 조율, 오픈소스 게임 및 코드 선택		

■ 회의내용

- 회의내용
 - [오픈소스] 각자 조사해 온 게임 정보 교환 및 최종 주제 결정
- 결정사항
 - [일정] 프로젝트 수행 일정 조율
 - [오픈소스] 게임주제: Snake game으로 결정
GitHub코드는 좀 더 조사 후 선택하기로 결정
- 향후일정
 - [정기회의] 2019.04.10 센B112에서 정기 회의 진행 예정
 - [오픈소스] 선택한 Snake game의 코드를 나누어 분석할 예정

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
[오픈소스] GitHub에서 Snake game 코드 조사	김경남	~4월 10일	회의록 작성
[오픈소스] GitHub에서 Snake game 코드 조사	진수빈	~4월 10일	

일시	2019. 04. 10 16:30 ~ 18:30	장소	센B112
작성자	진수빈	작성일	2019. 04. 10
참석자	김경남, 진수빈		
안건	Snake game 코드 결정 및 역할분담		

■ 회의내용

- 회의내용
 - [오픈소스] 조사한 Snake game 코드 공유 및 최종 선택
선택한 코드 분리 및 분석할 부분 선택
- 결정사항
 - [오픈소스] 코드: process/Snake로 결정 (<https://github.com/process/Snake>)
역할분담: 선택한 함수 코드 분석 및 flow-chart를 작성하기로 결정
- 향후일정
 - [정기회의] 2019.04.17 센B112에서 정기 회의 진행 예정
 - [오픈소스] 각자의 flow-chart 확인 및 전체 flow-chart 작성 예정

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
[오픈소스] InitSnake(), UpdateSnake(), SetPos(), UpdateScreen(), FreshDraw() 함수 분석 및 flow-chart 작성	김경남	~4월 17일	
[오픈소스] GameLoop(), GameOver(), PrintCenteredText(), main(), DrawFood() 함수 분석 및 flow-chart 작성	진수빈	~4월 17일	회의록 작성

일시	2019. 04. 13 19:30 ~ 2019. 04. 14 03:30	장소	- (카카오톡을 통한 온라인 회의)
작성자	김경남	작성일	2019. 04. 13
참석자	김경남, 진수빈		
안건	Snake game 코드 재조사, 결정 및 역할분담		

■ 회의내용

※ [긴급회의] 기존의 분석했던 코드가 C++인 것을 깨달아 긴급회의 진행

- 회의내용
 - [오픈소스] 게임주제: Snake game으로 유지할지 회의
 - [일정] 프로젝트 일정 재조율
- 결정사항
 - [오픈소스] 게임주제: Snake game으로 주제를 유지하기로 결정
코드: 새로운 소스 코드 조사 및 최종 결정
(<https://github.com/abdulwahid2802/SnakeGame>)
역할분담: 선택한 함수 코드 분석
 - [일정] 프로젝트 수행 일정 재조율 완료
- 향후일정
 - [정기회의] 2019.04.17 21:00부터 카카오톡을 통해 정기 회의 진행 예정
 - [오픈소스] 각자 맡은 부분의 flow-chart를 작성할 예정

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
[오픈소스] InitMap(), DrawMap(), GetInput() 함수 분석	김경남	~4월 17일	회의록 작성
[오픈소스] EatFood(), Die(), main() 함수 분석	진수빈	~4월 17일	

일시	2019. 04. 17 21:00 ~ 22:30	장소	- (카카오톡을 통한 온라인 회의)
작성자	진수빈	작성일	2019. 04. 18
참석자	김경남, 진수빈		
안건	forest 및 tree level의 flow-chart 제작		

■ 회의내용

- 회의내용
 - [오픈소스] 각자 맡은 코드분석 확인 및 flow-chart 작성
- 결정사항
 - [오픈소스] flow-chart: 각자 맡은 코드를 tree level의 flow-chart로 제작 후, 팀장이 forest level의 flow-chart를 제작하기로 결정
- 향후일정
 - [정기회의] 2019.04.27 센B112에서 회의 진행 예정
(04.24는 중간고사 기간이므로 회의 취소)
 - [오픈소스] 보고서(과제3) 작성에 초점을 맞출 예정

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
[오픈소스] 1. InitMap(), DrawMap(), GetInput()의 코드 주석 및 flow-chart 작성 2. 전체 flow-chart 작성	김경남	1. ~4월 18일 2. ~4월 19일	
[오픈소스] EatFood(), Die(), main()의 코드 주석 및 flow-chart 작성	진수빈	~4월 18일	회의록 작성

일시	2019. 04. 29 16:30 ~ 19:30	장소	학술정보원 4층 창의토론라운지
작성자	김경남	작성일	2019. 04. 27
참석자	김경남, 진수빈		
안건	보고서 작성		

■ 회의내용

- 회의내용
 - [오픈소스] 과제4 피드백 확인
 - [보고서] 목차 세부사항 설정
- 결정사항
 - [보고서] 목차 세부사항 결정
각자 맡은 부분을 작성하기로 결정
- 향후일정
 - [오픈소스] 코드를 클린코드로 개선 및 기능 추가
 - [보고서] 보고서(과제3) 제출

■ 업무분담 담당리스트

업무사항	담당자	기한	비고
보고서 목차 3, 4, 5, 7번 작성	김경남	~5월 3일	회의록 작성
보고서 목차 1, 2, 3, 7번 작성	진수빈	~5월 3일	

7. 참고문헌 및 사이트

- [그림1] 출처 : CoolmathGames, “snake game image”, <https://www.coolmathgames.com/0-snake>
- MIT 라이선스 주요내용: 오픈소스SW 라이선스 종합정보시스템, “MIT 라이선스 주요내용”, <https://www.olis.or.kr/license/Detailselect.do?lId=1006&mapCode=010006>
- [그림14] 출처 : SOFTWARE TESTING TUTORIALS - MANUAL AND AUTOMATION QUESTIONS ANSWERS, “waterfall model”, <http://jobsandnewstoday.blogspot.com/2013/04/what-is-waterfall-model.html>
- 프로젝트 유형에 따른 SDLC 선정(폭포수 모델): 봉봉이 아빠의 관심사, “폭포수 모델 프로젝트 규모”, <https://bongbonge.tistory.com/category/%EC%A0%95%EB%B3%B4%EA%B4%80%EB%A6%AC%EA%B8%B0%EC%88%A0%EC%82%AC/%EB%8F%84%EB%A9%94%EC%9D%B82.%20%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4%20%EA%B3%B5%ED%95%99?page=3>, (2015.07.13)