

---

# 공개SW프로젝트

## 〈프로젝트 제안서〉

---



제출일	2018.04.15.	전공	컴퓨터공학전공
과목	공개SW프로젝트	학번	2010111797
담당교수	김동호 교수님	이름	김강민
담당조교	강희철 조교님	팀명	강양중강양(1조)

# 목 차

## ■ 개요

1. 팀 소개
2. 프로젝트 선정 이유
3. 요구사항 분석
4. 개발환경

## ■ 구현

1. 사용자(Player) 관점
2. 개발자(Developer) 관점
  - ① 인터페이스
  - ② 네트워크
3. 구현 계획
  - ① 프로그램
  - ② 서버
  - ③ 데이터베이스
  - ④ 추가 기능

## ■ 협업

1. 목적 및 기대효과
2. 향후 개발 계획
  - ① 업무 분담
  - ② 타임라인
  - ③ 프로젝트 관리 도구 활용

## ■ 개요

### 1. 팀 소개

‘강양중강양’은 음악에서 4분의 4박자로 가장 안정적이고 기본이 되는 박자이며 누구에게나 친근하고 익숙한 박자이다. 매일 같이 키보드를 두드리는 소리에서 착안하여 리드미컬 하고 역동적인 팀이 되고자 팀 이름을 ‘강양중강양’으로 선정하였다. 또한, 팀 이름에 팀의 정체성을 담기 위해 팀원의 이름을 이용하여 팀명을 선정하게 되었다.

### 2. 프로젝트 선정 이유

테트리스라는 게임은 남녀노소를 불문하고 친숙한 게임이다. 높은 인지도만큼 현재 Github 내에서도 테트리스 관련 오픈 소스가 많이 존재한다. 그 점에서 다양한 오픈 소스를 활용할 수 있을 것이라는 생각이 들었다. 또한, 테트로미노 블록을 이용하여 효율적이고 체계적으로 보드를 구성해 나가는 방법을 고민하는 게임의 방식이 코드를 구성하는 과정이나, 협업을 하는 과정과 가장 유사하지 않을까하는 생각이 프로젝트를 선정하는 것에 큰 영향을 미치게 되었다. 이전에는 단일 콘솔 창 내에서 단일 사용자가 플레이 하는 테트리스만을 구현해 보았기 때문에, 이번 프로젝트를 통해 네트워크를 이용하여, PvP(Player VS Player) 기능을 지원하는 테트리스를 구현해 보자는 의견으로 종합되어 이 프로젝트를 선정하게 되었다.

### 3. 요구사항 분석

본 프로젝트를 수행함에 있어서 가장 중요한 것은 오픈소스를 이용한 개발 능력과 협업 이다. 특히, 오픈소스 중에서도 규모가 큰 코드를 분석하고, 해당 코드가 설계된 구조를 이해하는 것은 물론, 단점을 보완하고 추가적인 기능까지 구현하는 것이다.

또한, 개발자마다 코드를 작성하는 방법과 설계 방식이 다양하기 때문에, 다른 개발자의 생각과 논리를 이해하는 능력과, 추후에 코드를 작성할 때 다른 개발자가 쉽게 논리를 이해할 수 있도록 구현하는 능력, 협업을 통해 전체 코드를 한 명의 개발자가 작성하는 것이 아닌 각각의 모듈에 대해 개발할 수 있는 능력 및 모듈을 종합할 수 있는 능력이 요구된다.

### 4. 개발환경

- 운영체제 : Linux(Ubuntu 16,04 LTS)
- 통합개발환경 : Eclipse
- 개발언어 : Java, MySQL, HTML, PHP
- 서버 : Ubuntu Server
- 데이터베이스 : MySQL

## ■ 구현

### 1. 사용자(Player) 관점

사용자 관점에서 게임 플레이 시 불편하거나, 필요한 추가 기능 구현에 대해 다음과 같은 의견이 종합 되었다.

#### 1) 유저들의 게임 흥미 유발을 위한 랭크 시스템 도입

- 현재는 단순히 게임의 승패만 결정되기 때문에 점수를 통해 순위가 기록되는 시스템을 도입하면 게임에 대한 유저들의 흥미가 높아질 것이다.
- 최고 스코어와 평균 스코어를 이용하여 각 유저들 간의 등급을 분류하여 제공한다.

#### 2) 게임 플레이에 역동성을 부여하기 위한 아이템 기능 추가

- 일정 점수마다 폭탄의 개수가 누적되고, 필요 시 폭탄을 사용하면 자신의 테트리스 보드 최하단 라인의 블록을 제거한다.
- 임의의 확률로 상대의 보드를 멈출 수 있는 아이템과 상대 화면을 교란하는 아이템을 제공한다.

#### 3) 게임 플레이 시간에 따른 속도조절 및 장애물 기능 추가

- 플레이어의 원활한 진행을 방해하여 유저의 몰입도를 높인다.
- 스코어가 일정 수준 이상으로 올라갈 경우, 본인의 블록이 이동하는 속도를 조정한다.

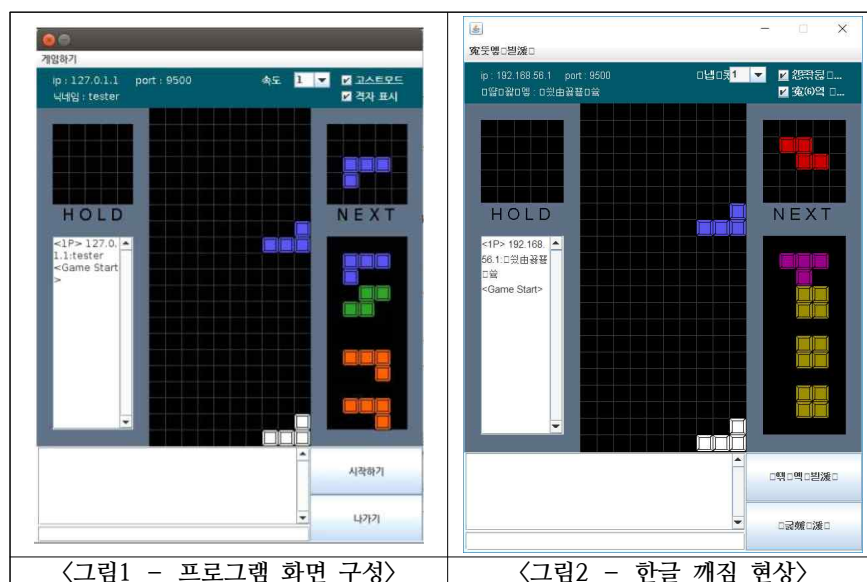
#### 4) 사용자 친화적 인터페이스 개선

- 현재 유저에게 제공되는 인터페이스의 구성이 직관적이지 않으며 게임에 익숙하지 않은 유저들이 쉽게 게임을 익힐 수 있도록 조작방법에 대한 안내와 연습환경을 제공한다.

### 2. 개발자(Developer) 관점

사용자 관점과 달리 실질적인 개발자의 관점에서 구현방향을 인터페이스, 네트워크 두 분류로 나누어 접근하였다.

#### ① 인터페이스



프로그램을 리눅스와 윈도우에서 각각 빌드 하였을 때, 윈도우 환경에서는 한글이 깨지는 현상이 발생하였다. 특정 환경에서만 정상적으로 동작한다는 것은 프로그램의 확장성 및 이식성이 떨어진다는 것을 의미한다. 또한 게임 승패의 결과가 나타나지 않아 게임의 종료를 플레이어가 명확하게 인지 할 수 없다.

## ② 네트워크

포트 번호를 직접 지정해서 매칭 하는 부분에 있어 포트 중복에 대한 처리가 미흡하여 플레이에 제약이 발생할 수 있다. 또한 게임 매칭 후 실제 플레이를 하는 상황에서 준비가 완료되었다는 정보와 상대방에 대한 정보가 주어지지 않는다. 그리고 채팅 기능은 구현되어 있지만 비속어 필터링 기능이 없어 불량 유저에 대한 제재를 가할 수 없다. 거기에 만약 한 플레이어가 접속을 종료하거나, 외부적 요인에 의해 프로그램이 중지되었을 때, 남아있는 플레이어는 계속 진행되는 문제가 있어 개선할 사항이 보인다.

## 3. 구현 계획

### ① 프로그램

현재 프로그램에서 사용하는 GUI 라이브러리는 AWT를 사용하고 있는데 AWT는 운영체제에 의존도가 높아 다양한 플랫폼에서 일관성 있게 사용하기 어렵다는 한계가 있다. 따라서, 플랫폼에 구애받지 않도록 SWING 라이브러리를 사용하여 구현한다.

### ② 서버

- 기본적으로 서버OS는 우분투 서버를 사용하며, 각 사용자가 웹페이지에 접속한다.
- 자신의 OS에 맞는 게임 실행 파일을 다운 받을 수 있게 구현한다.
- 자신의 랭크와 최고 스코어를 확인할 수 있고, 그것을 다른 사람 또는 친구와 비교 가능하게 해준다.

### ③ 데이터베이스

유저들의 정보가 담긴 테이블을 구성하고 MySQL를 통해 데이터를 저장하는 데이터베이스를 구축한다.

### ④ 추가 기능

주기적으로 상대의 스코어를 보여주는 기능을 구현할 예정이다.

## ■ 협업

### 1. 목적 및 기대효과

대규모의 프로젝트를 진행할 때, 한 개발자가 모든 분야의 코드를 작성하는 것은 개발 속도에 있어서 시간이 많이 소요되고, 방대한 분야의 지식을 모두 알아야 한다는 한계가 존재한다. 오늘날의 소프트웨어는 개발 분야가 점점 세분화 되고 전문화 되고 있기 때문에, 한 명의 개발자가 모든 것을 담당하는 것은 거의 불가능하다고 볼 수 있다. 따라서 프로젝트를 수행하기에 앞서 각자의 전문 분야를 담당하고 분할하여 개발한 후, 각각의 개발 사항을 통합하는 방식이 앞선 방식에 비해 월등하게 유리하다. 하지만 확실한 분업을 위해서는 구성된 개개인이 프로젝트 및 세부 내용을 전반적으로 이해할 수 있는 능력을 갖추어야 하며, 코드를 작성하거나 관련 자료를 작성할 때, 본인만 보는 것이 아니기 때문에 본인의 생각이 타인에게 쉽게 이해 될 수 있도록 코드를 작성하는 능력과, 모두가 이해할 수 있는, 어느 정도는 정형화 된 방식으로 코드를 작성하여 가독성도 확보해야 한다. 또한 다양한 프로젝트 관리 도구에 대한 사용 방식을 숙지하고 있어야 원활한 협업이 이루어 질 것이다.

특히, 협업에 있어서는 현재 팀원 모두가 교내 중앙 동아리인 ‘네트워크 사용자 연합(DNA)’에 소속되어 있어 정기적인 회의는 물론, 긴급회의를 진행하기에도 유리하고, 회의 장소나 시간을 선정하는 것도 수월할 것이라 예상된다. 이러한 유리한 조건 속에서 주기적인 대화와 소통 그리고, 협업을 통해 팀의 결속력을 강화하고, 팀원 개개인이 협업의 의미에 대해 생각해보고, 다양한 관리 도구를 이용하여 효율적인 협업을 경험하고, 적용할 수 있는 계기가 될 것이라 기대된다.

### 2. 향후 개발 계획

#### ① 업무 분담 및 계획

개발 업무를 분담하기에 앞서 먼저 각각의 팀원이 사용 가능한 언어에 대한 사항을 종합하는 것이 우선이라 생각하여 조사를 시작하였다. 우선, 모두 C, C++에 대한 기초적인 지식은 보유하고 있었지만, 팀원 별로 MySQL, PHP, Java 등 개인이 지니고 있는 지식이 상이한 부분도 존재하였다. 이를 바탕으로 팀원들의 역할에 대해 조율하기 위해 지속적인 회의를 진행하였고, 최종적으로는 각자가 최대한 선호하는 역할로 분담하게 되었다. 김강민, 양승영 팀원은 ‘인터페이스 개선 및 추가 기능 구현’ 업무를 수행할 예정이며, 김성백 팀원은 ‘데이터 베이스 분할 및 서버’를 담당하고, 임호성 팀원은 ‘웹 페이지 및 서버’를 담당할 예정이다.

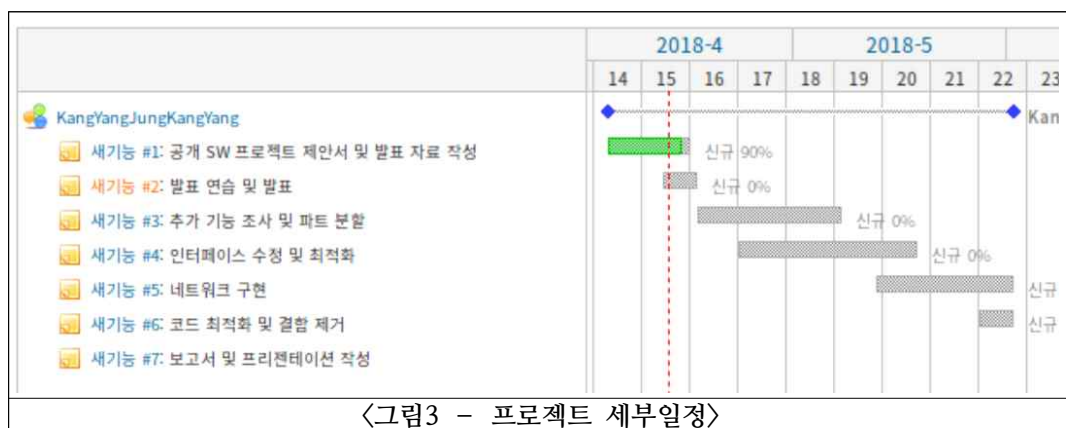
향후 계획으로는 현재 대상 프로젝트가 네트워킹 기능이 구현은 되어 있지만, 양 측의 포트 번호와 IP 주소를 정확하게 알아야 한다는 한계가 분명히 있기 때문에 이를 극복하기 위해, 웹 페이지와 서버를 구축하여 각각의 사용자가 본인의 아이디와 패스워드를 통해 접속하는 방법을 구현하여 사용자가 보다 쉽고 편리하게 게임을 즐길 수 있게 할 것이다. 거기에 유저 인터페이스는 더욱 사용자 친화적이며 보다 직관적으로 구성할 계획이다.

프로젝트 진행에 따른 위험 요소로는 먼저 서버의 안정성 및 보안이 보장이 되어야 한다. 서버에 데이터베이스를 구축하고 개인 정보를 수집하기 때문에, 보안에 대해 분명히 고려하여야 한다. 더욱이 원활한 운영을 위해, 갑작스러운 다중 사용자 접속 즉, 트래픽 처리의 부분과 접속

자들간의 동기화 부분 등을 고려해야 하며, 사용자가 단일 환경에서만 프로그램을 실행할 것이라는 보장이 없으므로 단일 실행 환경에 종속되지 않도록 개발해야 할 것이다. 또 다른 보완점으로는 소켓 프로그래밍과 Java 소켓의 직렬화, 역직렬화를 할 때 문제가 발생할 여지가 있을 것으로 판단되었다.

## ② 타임라인

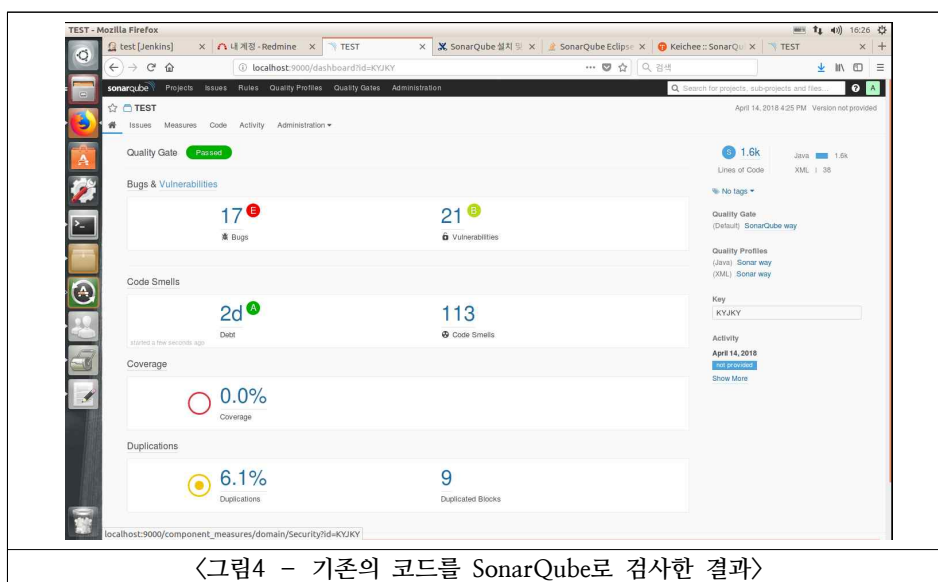
Redmine의 Gantt Chart 기능을 이용하여 1차적으로 전반적인 프로젝트 세부일정을 계획하였다. Redmine을 이용했을 때에는 진행률이나, 담당 팀원을 쉽게 식별할 수 있고, 기존에 차트를 만들었던 방식은 수정 사항이 생겼을 때 즉각적으로 반영하기가 어렵지만 Redmine에서는 즉시 항목을 추가 또는 수정 및 제거가 가능하다는 점에서 좋았다.



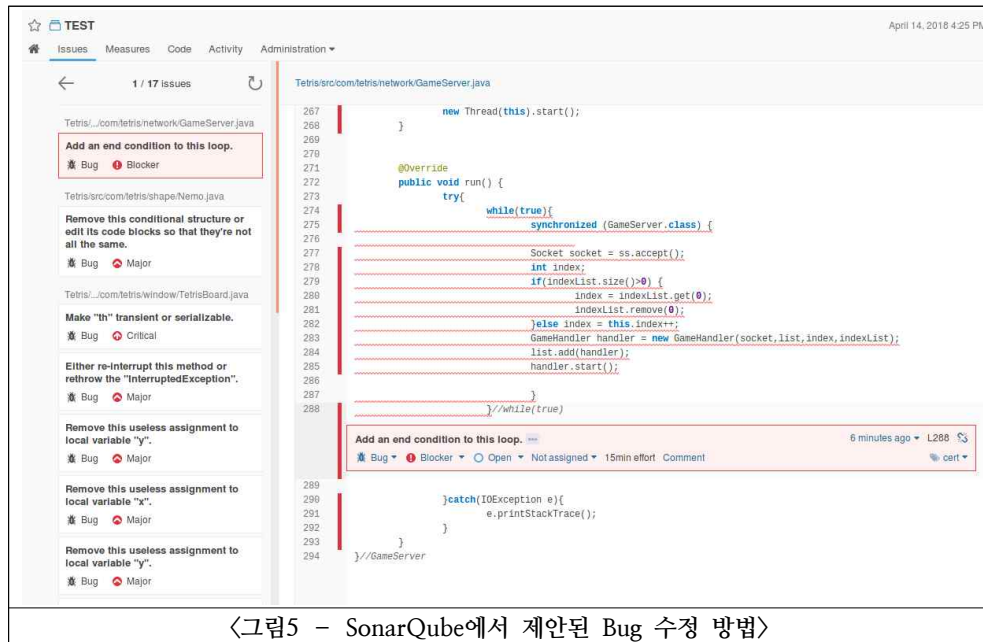
〈그림3 - 프로젝트 세부일정〉

## ③ 프로젝트 관리 도구 활용

- 1) Redmine을 이용한 팀원 간의 프로젝트 일정 관리
- 2) Eclipse와 연동이 되는 SonarQube를 통해 작성한 코드에 대한 품질 향상
- 3) Gerrit과 Jenkins를 활용해 Git 서버 저장소 관리 및 코드 리뷰를 통한 의견 공유



〈그림4 - 기존의 코드를 SonarQube로 검사한 결과〉



〈그림5 - SonarQube에서 제안된 Bug 수정 방법〉