1. 개요

- 개발 동기 및 필요성

한 학기동안 배운 java를 이용한 객체지향 프로그래밍을 더 잘 이해하고자 기존에 있는 보드게임 중 하나를 java로 구현해 본다.

- 목표

자바를 이용하여 객체지향 프로그래밍 패러다임에 의거해 디자인된
보드게임 "Laser Maze"를 제작 한다.

2. 일정

11-23 3시간 구현을 위한 기능 공부

11-24 5시간 기존에 실습시간때 했던 움직이는 sprite수업코드를 기반으로 LazerMaze 클래스 생성. 자바 JFrame의 버튼 구성 및 reader를 통한 10개 스테이지 토큰정보 텍스트파일 읽기.

11-30 3시간 텍스트 파일에서 읽어들인 토큰 정보를 tokenManager를 통해 토큰화. lazerMazePreview JFrame을 만들어 미리보기 기능 제공 및 OK버튼 클릭 시 헤당 위치에 토큰 생성(토큰의 방향, 움직임 가능 여부 아직 구현 X).

12-05 8시간 토큰 더블클릭 시 회전 기능 구현. 토큰 드래그앤 드롭 기능 구현. 충돌검사 및 Lazer 클래스 생성 후 레이저가 토큰에 닿을 때 토큰에서 수행하는 메소드 doSomthingToLazer 구현.
Solution, Init, Lazer버튼 구현.

12-06 5시간 콘솔창을 통한 doSomethingToLazer 디버그 및 레이저 출력.
코드 단순화를 위해 텍스트 파일 읽는 방식 변경.
여러 경우의 수를 테스트 하여 발견되는 버그 수정.

12-07 2시간 스테이지 20단계로 확장 및 디버그

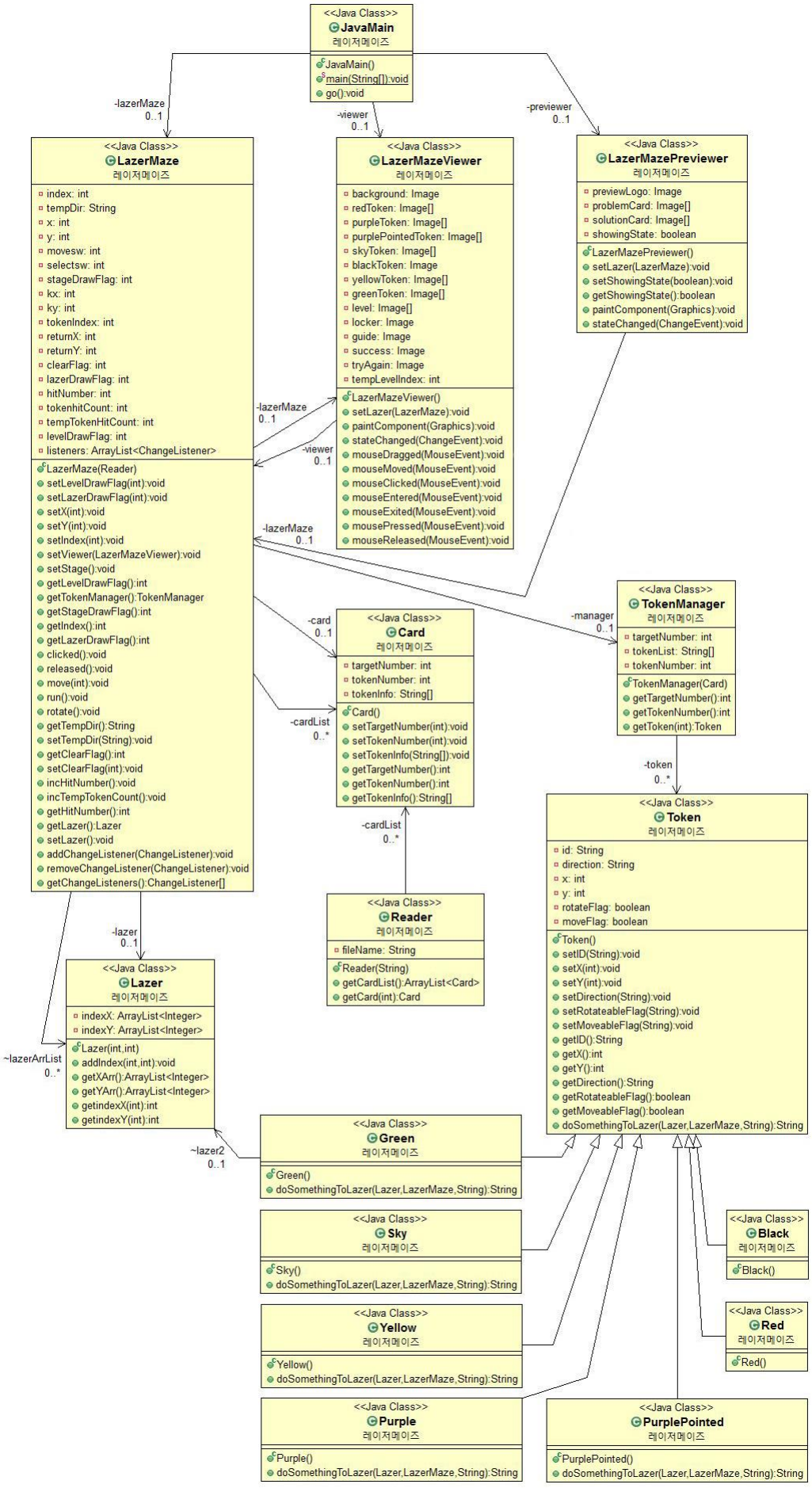
12-08 2시간 시퀀스 다이어그램, 클래스 다이어그램 이미지 파일로 제작 및 보고서 작성

12-09 1시간 비쥬얼적 측면 수정 (Look&feel , 산세리프볼드폰트, 레이저 색)

12-10 30분 레이저 애니메이션& 끝에 Arrow 추가

3. 시스템 설계

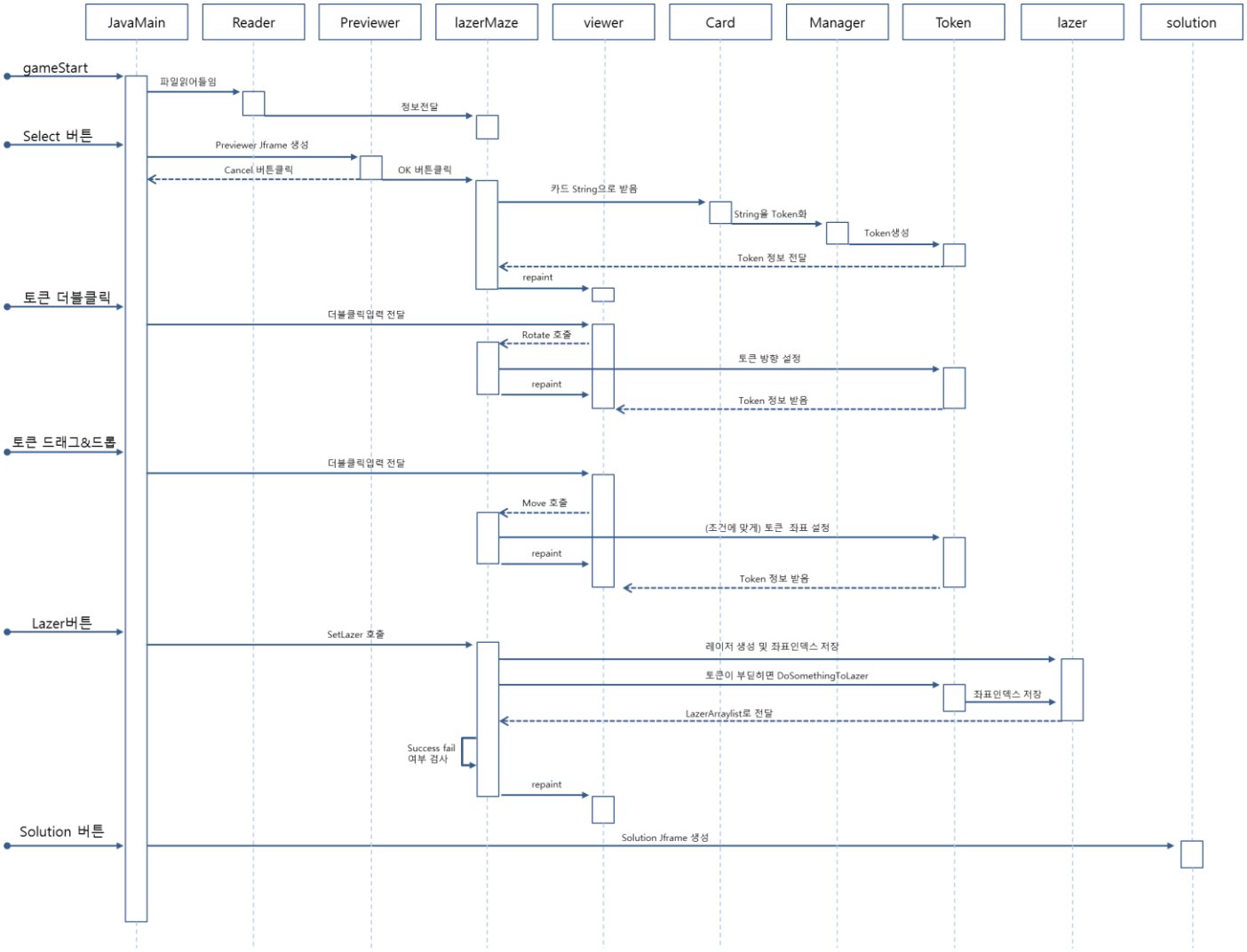
- 클래스 구성도 및 설명 (만약 깨져서 안보이시면 같이 압축한 이미지를 참고하세요)



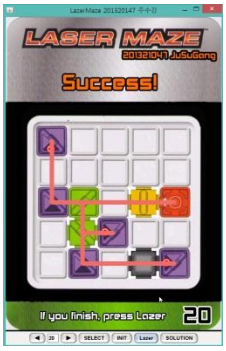
클래스 다이어그램 설명

- JavaMain 에서 LazerMaze, LazerMazeViewer, LazerMazePreviewer 를 인스턴스로 가지고 있다. (셋 다 인스턴스로 굳이 가질 필요는 없어보이지만 혹시 몰라서 인스턴스로 만들어 두었다.
- LazerMaze 가 전반적인 controller 를 담당한다. Select 나 Solution 버튼을 누르면 나타나는 새 창을 위한 LazerMazePreviewer 를 만들고, 실제로 회전되고 이동하고 레이저를 그리는 부분인 LazerMazeViewer 를 만들었다.
- Reader 에서 카드를 읽은 뒤 스트링으로 저장하고, 이를 바로 TokenManager 에서 관리했다면 더 깔끔했을텐데 중간에 LazerMaze 에 ArrayList 로 Card 가 저장되느라 약간 돌아간다.
- Token 에 전반적인 좌표나 회전정보, 움직임 가능정보 등을 TokenManager 를 통해 수정된 정보들로부터 받는다.
- Lazer 는 좌표만 가지고 이 정보를 나중에 LazerMazeViewer 에서 그리게 된다.

◆시퀀스 다이어그램



- 실행 화면 등



4. 시스템 구현

- 구현 언어, 도구, 지원 프로그램 등

JAVA eclips(4.4.0 LUNA)

- 주요 실행 결과 표시

- 프로그램 소스 등

Eclips 파일 별첨

5. 결론

동기 및 개발시스템, 그리고 얻은 결과에 대한 요약을 기술한다.

저번 학기 때 신현준 교수님께 배웠던 게임 프로그래밍1 수업이 정말 큰 도움이 되었다. 그 당시에도 text파일을 이용한 map을 구현했었는데, 그때 한번 File I/O를 다뤄보았기 때문에 이번 레이저메이즈의 출력 실마리를 찾을 수 있었다.

우선, 처음 자바를 이용하여 이 게임을 만들어 낸 것이 뿌듯하고 자랑스럽지만, 객관적으로 보면 50점짜리 결과물이다. 왜냐하면 난 맨 처음에 펜을 들고 클래스다이어그램과 시퀀스다이어그램을 고민하지 않았으며, 그렇기 때문에 내가 짠 코드는 다른 곳에 재사용이 불가능할 정도로 서로 얽혀있다. 아마 실습시간에 조교님이 설명해주신 Board를 이용해 LazerMaze가 순수 컨트롤러 역할만 했다면 좀 더 연결성을 떨어뜨릴 수 있었을 지 모른다. 그래서 만약 또 이렇게 프로그래밍을 할 일이 생긴다면, 클래스를 가장 먼저 구상하면서, 재사용이 가능한가에 대한 고민을 끊임없이 해야 겠다.

맨 처음 화면을 열면 기본 판과 레벨선택을 위한 ◀,텍스트필드,▶, SELECT버튼만 보이고, 나머지 기타 등등 버튼은 사용하지 않기 때문에 setVisible(false)로 해두었다. ◀,▶ 버튼을 이용하여 레벨을 수정하며, 1미만인데 ◀ 를 누르면 1로 고정, 20이상인데 ▶ 를 누르면 20으로 고정하였다. 텍스트필드에 직접 입력할수도 있으며, 이때도 1 미만의 수를 입력하면 1로 20초과의 수를 입력하면 20으로 인식하도록 하였다. SELECT 버튼을 누르면 미리보기 화면을 띄우고 CANCEL을 누르면 단순히 화면만 꺼지고, OK를 누르면 미리보기에 따라 판이 세팅된다. 세팅됨과 동시에 INIT, Lazer, SOLUTION 버튼이 하단 패널에 추가된다. 기본적인 규칙대로 판에 이미 있는 토큰은 움직일 수 없으며, 회전할 수 없는 토큰에는 흰색 격자로 표시를 해 두었다. 놓여있지 않는 토큰은 토큰 밖으로 Drag&Drop하면 이전 자리로 돌아온다. 토큰을 보드 안으로 Drag&Drop하면 격자에 맞게 자석처럼 딱 붙으며, 다른 토큰이 이미 있는 곳으로 Drag&Drop하면 바로 이전에 있던 위치로 되돌아간다. INIT버튼을 누르면 맨 처음 상태로 되돌아간다. 문제를 풀어 토큰을 놓는 대로 Lazer가 발사되며 만약 답을 모르겠으면 SOLUTION을 눌러 답을 확인할 수 있다. Lazer가 token의 규칙에 따라 Success! 와 Try again이 나오고, Yellow토큰의 검사나, 모든 토큰이 지나지 않았을 경우에도 Try again이 발생한다.