

SE 2022 봄학기 팀프로젝트:

OOAD를 이용한 Bridge 게임 개발

팀규모: 개인 과제이므로 각자 프로젝트를 독립적으로 진행해야 함 (1인 1팀)

제출마감: 6월 10일 금요일 오후 10시

팀프로젝트 배점: 금번학기 총점 중 45% (중간고사를 대체하는 팀 프로젝트이며, 별도의 그룹 프로젝트는 없음)

본 프로젝트는 필요한 기능을 정확히 구현하는 것 뿐 아니라 좋은 설계, 충실한 문서화 측면에서도 평가가 됨을 주목해야함. 또한 주어진 요건 외에도 보다 나은 설계(적합한 아키텍처 설계, 확장성과 유지 보수성 좋은 설계, 단위 테스트 용이한 설계 등) 및 그의 구현을 위해 노력한 점이 있다면, 프로젝트 문서를 통해 텍스트, 다이어그램, 코드 등을 이용해 명확히 설명하고, 문서 및 비디오 클립에서 해당 사항을 강조할 것.

제출방법: eclass 과제 및 평가 탭의 과제 "Project"에 업로드

소프트웨어 구현 언어: Java 혹은 C++

제출물: "성명_학번.zip" 형식의 파일명을 갖는 zip 파일 1개를 제출하며, 해당 zip 파일은 다음을 포함해야 함

- README.txt 파일: 제출된 모든 산출물에 대한 목록 및 요약.
- 모든 소스코드 파일, 실행 파일 (Java의 경우 class 파일, C++의 경우 exe파일), 데이터 등
- 프로젝트 문서 (**PDF 형식 파일로 제출**, 문서에 포함되어야 하는 내용은 하기 설명 참고)
 - 만일 프로그램이 완성되지 못하였다면 어떤 부분들이 부족한지 설명을 해야함
- 비디오 클립:
 - 비디오 클립의 길이는 15분 내외를 추천함. 최대 길이는 25분으로 이보다 더 긴 클립의 경우 25분 이후 내용은 평가에 반영하지 않을 것임
 - 비디오 클립의 시작 부분에는 신분증 제시와 함께 본인의 얼굴을 보여주고 이름과 학번을 말하는 영상과 음성이 포함되어 있어야 함. 이후로는 화면과 음성만 있어도 관계 없음
 - 신분 확인 이후에는 발표에 어떤 내용들이 있는지 목차를 제시하고 설명해야 함
 - 제출된 프로그램을 실행하며 주요 기능들이 잘 동작하는 것을 시연함. 프로그램의 구동부터 각 기능들의 수행을 효과적으로 보여줘야 함.
 - 이후로는 요구 명세 및 분석, 설계 및 구현 등에 대한 설명을 포함하며 특히 주요한 설계 결정들에 대해 언급하고 왜 그와 같은 설계 결정을 했는지, 또한 어떤 장점이 있는지 등을 설명할 것

Project Description:

본 과목의 수강을 통해 그간 배운 OOAD 기법을 사용해서 소프트웨어 개발을 수행한다. 본 프로젝트는 2~4명의 사용자들이 규칙에 따라 Bridge 보드 게임을 개발하는 것이 목표이다. (하기에 충분히 설명되지 않은 조건 및 세부 명세 등에 대해서는 각자 적절한 가정을 세우고 프로젝트 문서에서 해당 가정을 명시할 것)

지도 로드 기능

- 프로그램이 시작되면 default.map 파일을 디폴트로 로드함
- 파일 시스템에 있는 임의의 지도 데이터 파일을 로드할 수 있어야 함

플레이 기능

- 게임 플레이를 선택하면 플레이어의 수를 입력하고 마지막으로 로드된 지도를 이용하여 게임을 진행함
- 각 플레이어의 현재 위치가 지도상에 표시되어야 함
- 각 플레이어가 가진 카드의 개수가 표시되어야 함
- 게임이 종료되면 각 플레이어의 점수가 표현되어야 함
- 각 플레이어의 턴이 되면, 설지(stay) 혹은 주사위를 굴릴지를 결정하도록 함. 쉬기로 결정한 경우 다리 카드 1장을 시스템으로 반납하게 됨.
- 주사위를 굴리는 경우 게임 시스템이 랜덤하게 주사위를 1개 굴려서 1~6까지의 값을 제시함
- 플레이어는 U,D,L,R 혹은 u,d,l,r의 조합을 한줄로 입력하여 (각각 위, 아래, 왼쪽, 오른쪽 이동에 해당), 주사위 값 및 지도에 합당하다면 플레이어의 말(piece)은 지시한대로 진행하게 됨. 단 이동할 수 없는 방향의 정보가 포함되어 있거나 주사위의 값과 일치하지 않으면 재차 입력하도록 함. 예를 들어 dddrru 은 아래로 세번 내려간 후 오른쪽으로 두번, 그리고 위로 한번 이동하는 명령임.

유저 인터페이스(User Interface, 이하 UI) 관련 사항

- 기본적인 요건은 GUI(Graphical User Interface) 혹은 콘솔 (텍스트 형식) 인터페이스를 통해 게임이 동작하도록 만드는 것임
- 설계 시 UI와 응용 로직(Logic)을 분리하여 추후 UI가 바뀌더라도 로직 이하의 계층은 변경없이 재사용할 수 있도록 설계 및 구현해야 함. 이와 같은 설계 내용이 프로젝트 문서에도 명확히 나타나야 함
- GUI 방식과 콘솔 방식을 둘다 구현하였으며 이들을 지원하기 위해 적절한 설계가 되어 있다고 평가된다면 가산점을 부여함

프로젝트 문서에 포함되어야 할 내용

[요구 정의 및 분석 산출물]

- 유스케이스 다이어그램, 유스케이스 명세 (3개 이상의 유스케이스에 대해), 도메인 모델, SSD (System Sequence Diagram), Operation Contract 등을 이용하여 요구 정의 및 분석 결과를 표현할 것. 텍스트 형태의 설명도 포함해야 함

[설계 산출물]

- Design Class Diagram, Sequence Diagram, StateChart 등 적절한 UML 다이어그램들을 이용하여 설계 결과를 표현할 것. 텍스트 형태의 설명도 포함해야 함

[구현 산출물]

- 소스코드 및 실행파일
- 프로그램 사용 방법
- 테스트 결과

플레이 규칙

보드의 시작 지점은 START 로 표시되고 다른 위치는 END 로 표시된다 (지도 데이터의 S 와 E 에 해당).

게임 시작시 START 에 모든 플레이어의 말(piece)이 위치한다.

각 플레이어는 자신의 턴(turn)에서 주사위를 굴러 몇 칸을 이동할지 결정한다.

플레이어는 주사위의 값에서 다리(bridge) 카드 개수를 뺀 만큼 이동한다. (단 음수인 경우 0 칸 이동함)

플레이어는 주사위를 한 번 굴려서 나온 값 만큼 앞 혹은 뒤로 이동하도록 선택할 수 있다.

(예를 들어, 주사위가 5 인 경우 플레이어는 앞으로 2 칸 이동한 다음 뒤로 3 칸 이동하여 원래 칸 뒤로 1 칸 이동할 수 있다)

플레이어는 이렇게 하여 진행하는 칸을 희생해서 도구 카드를 집을 가능성을 높일 수 있다.

어떤 플레이어가 END 를 넘으면 보드에 남아 있는 나머지 플레이어들은 더 이상 뒤로는 이동할 수 없게 된다.

가장 처음 END 를 넘은 플레이어는 7 점, 두 번째 플레이어는 3 점, 세 번째 플레이어는 1 점을 얻는다.

플레이어는 각 도구 카드에 대해 다음과 같은 점수를 얻는다.

- Philips Driver (P) : 1점
- Hammer (H) : 2점
- Saw (S): 3점

END 셀에 들어가려면 플레이어는 남은 셀의 수보다 크거나 같은 주사위 숫자를 얻어야 한다.

한명의 플레이어만 보드에 남으면 게임은 끝난다. 그 플레이어는 더 이상 턴을 진행할 수 없으며 각 도구 카드에 대해 점수를 받게 된다.

게임이 끝날 때 가장 많은 점수를 얻은 플레이어가 승리한다.

플레이어는 이동 시 다리를 건너거나 혹은 건너지 않고 갈 것인지 결정할 수 있다.

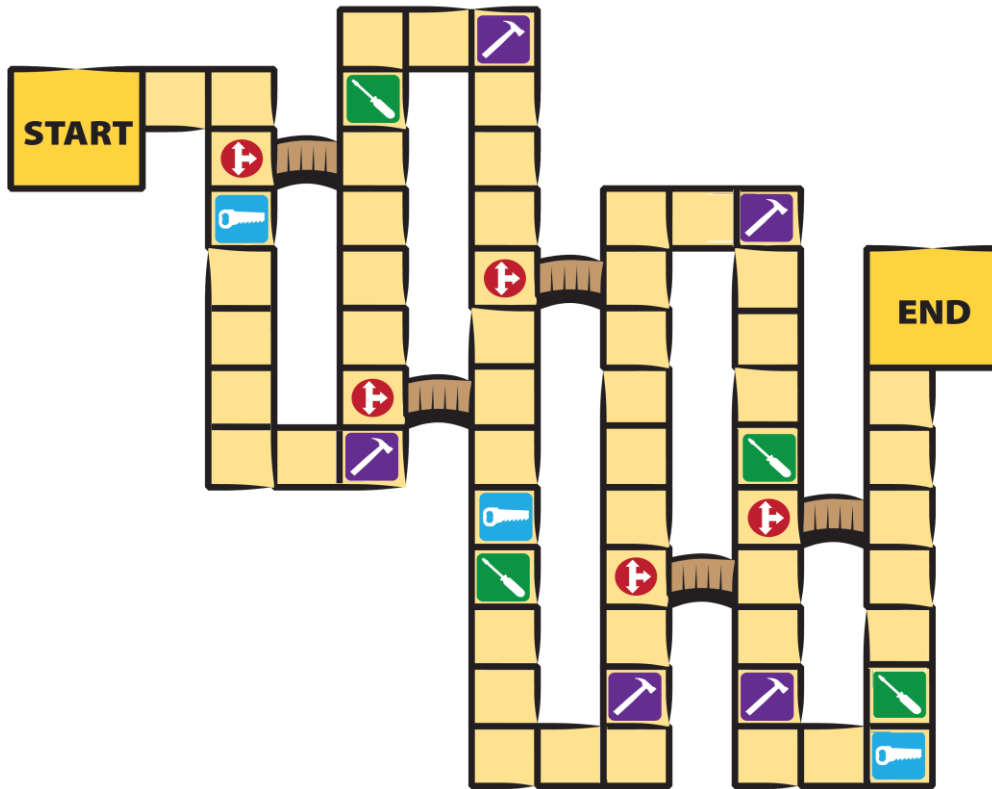
다리는 일반적인 셀과 마찬가지로 하나의 이동으로 계산된다.

다리를 건너 플레이어는 다리 카드를 한 장 받는다.

플레이어는 자신의 턴일때 주사위를 굴리지 않고 쉬어서, 다리 카드 1개를 제거할 수 있다.

이 게임에서 다리는 항상 왼쪽 셀과 오른쪽 셀을 연결한다고 가정한다.

지도 데이터 파일 예시 1



데이터 파일은 셀 종류 정보를 나타내는 한 개의 캐릭터, 그리고 공백으로 구분된 진행 가능 방향 캐릭터들로 구성됨

- 셀 종류
 - S: Start, C: Cell
 - B: Bridge의 시작과 인접한 셀, b: Bridge의 끝과 인접한 셀
 - H: Hammer, S: Saw, P: Philips Driver
- 진행 방향 종류
 - U: Up, D: Down, L: Left, R: Right
 - 단, B의 경우 Bridge가 오른쪽에, b의 경우 Bridge가 왼쪽에 있으므로 해당 방향으로 진행 가능하다는 것을 엄두에 둘 것

위 지도를 나타내는 데이터 파일(**default.map**)은 다음과 같음

S R
C L R
C L D
B U D
S U D
C U D
C U D
C U D

C U R

C L R

H L U

B D U

C D U

C D U

C D U

b D U

P D U

C D R

C L R

H L D

C U D

C U D

C U D

B U D

C U D

b U D

C U D

S U D

P U D

C U D

C U D

C U R

C L R

C L U

H D U

C D U

B D U

C D U

C D U

C D U

C D U

b D U

C D R

C L R

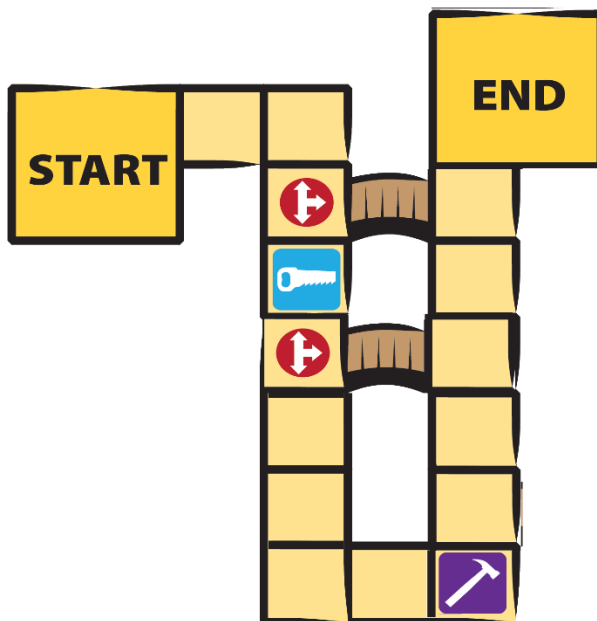
H L D

C U D

C U D

C U D
P U D
B U D
b U D
C U D
H U D
C U R
C L R
S L U
P D U
C D U
C D U
b D U
C D U
C D U
E

지도 데이터 파일 예시 2



위 지도를 나타내는 데이터 파일 (**another.map**)은 다음과 같음

S R
C L R
C L D
B U D

S U D

B U D

C U D

C U D

C U R

C L R

H L U

C D U

C D U

b D U

C D U

b D U

E