|  |
| --- |
| **왕국을 지켜라**  **최종 보고서** |

|  |  |
| --- | --- |
|  | 제출일자:2024. 12. 19  제출자명:한지훈  제출자학번:214620 |

1. **프로젝트 목표**
2. **배경 및 필요성**

대학에서 팀 프로젝트는 필수적인 요소이다. 팀 프로젝트에서 팀장을 맡게 되면 리더십과 팀원 관리 능력을 키우는 것이 중요하다. 이를 위해, 한 나라의 국왕 역 할을 수행하며 이러한 능력을 간접적으로 개발할 수 있는 간단하고 쉬운 Text게임을 제작하고자 한다.

1. **프로젝트 목표**

국왕이 겪을 수 있는 수많은 상황이 주어졌을 때 여러 가지 선택지 중 하나를 선택하여 총 4가지의 포인트를 관리하는 Text게임 제작.

1. **차별점**

기존에 겪어볼 수 없는 역할을 직접 맡아보며 여러 상황에 맞닥트렸을 때 해결 능력을 길러 볼 수 있음. 플레이는 누구나 할 수 있을 정도로 쉽고 재밌게 이러 한 능력을 기를 수 있다는 장점이 있음.

**2.** **기능 계획**

1. **질문, 대답 문자열이 들어있는 2차원 배열 선언**

* N개의 질문, 대답이 들어있는 문자열을 선언하여 while문에서 index에 따른 질 문사용예정

(1) 세부 기능: [N][0]에는 질문, [N][1], [N][2]에는 대답 저장

1. **질문에 따른 포인트 변화량을 담고 있는 2차원 배열 선언**

- 각 질문에 따라 4가지의 포인트의 변화량을 담고 있는 배열을 통해 쉽게 포인트 변화를 주고자 함

1. 세부 기능: 1차원 index는 0~7까지 총 8개이며, 0~3에는 1번 대답에 대한 포 인트 증감량, 4~7번에는 2번 대답에 대한 포인트 증감량을 담는다.
2. **일차와 각 포인트를 출력하는 함수 선언**

- 매 일차마다 몇 일차인지, 포인트가 어떤 지를 알아야 하므로 출력하는 함수를 생성 해야함.

1. **대답 입력 받기**

* cin을 통해 두 개의 선택지 중 하나를 입력 받는 기능

1. **대답에 대한 포인트 증감 및 출력**

- 입력 받은 선택지에 따라 index를 변환하여 포인트 변화량 배열에서 증감량을 가져와 변환.

1. **증감된 포인트 0이면 게임 종료**

- 포인트 4가지를 확인하여 0이면 while문 break 후 종료 문구 출력

1. **특정 일차가 넘어가면 게임 종료**

* N일차를 넘어가면 while문 break 후 종료 문구 출력.

**8)답변에 대한 문구 출력**

- 포인트 증감만 담으면 너무 심심하므로 포인트 증감 이후 포인트 증감 이유를 담은 문구 출력

세부 기능: 질문, 대답이 있는 2차원 배열에 열 두 개 추가하여 각 선택에 따른 문자열 추가. 대답에 대한 문구를 추가 출력해야 하므로 함수 및 기본 구조 변 경 필요 가능성 존재.

**9) 특정 포인트 양에 따라 출력되는 특수 질문 추가**

* 4가지 포인트 양에 따라 생기는 특수 질문 추가

세부 기능: 예를 들어 자산이 10으로 최대라면 관련된 질문이 나오도록 하는 기 능 추가. 특수 질문을 위한 새로운 2차원 배열 생성. 랜덤 변수 생성 시 포인트를 확인하고 그에 따라 나오는 질문 확률 직접 조정.

**10) 이미 나왔던 질문인지 검사**

- 기존의 프로그램을 업그레이드. 나왔던 질문은 다시 하지 않는 기능 추가 (bool 배열 사용)

1. **특정 포인트가 0이 되었을 때 나오는 특수 질문 추가**

* 0이 나오면 바로 죽지 않고 해당 상황에 맞는 특수 문구를 추가.

세부 기능: 기능 6번의 조건을 수정할 필요가 있음. 10번 기능과 비슷하지만 해 당 질문이 나오면 게임이 종료되는 특징이 있으므로 관련 함수 선언 필요해 보임.

**3. 기능 구현**

**(1) 질문, 대답 문자열이 들어있는 2차원 배열 선언**

- 프로그램 실행에 필요한 메인 변수와 배열을 선언. 질문이 담겨있는 questionsAndAnswers 배열에서 한 행은 각 질문을 나타냄. 각 열에서 0열은 질문, 1, 2에는 질문에 대한 답변이 담겨있음.

- 자료형, 2차원 배열

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**(2) 질문에 따른 포인트 변화량을 담고 있는 2차원 배열 선언**

- 질문에 대한 답변에 따라 4가지 포인트에 변화가 생기는데 이를 2차원 배열로 선언하여 코드를 짜도록 구조를 만듦. 각 행은 각 질문에 대한 것이고 각 열에서 0~3열은 답변 1에 대한 포인트 변화량 (종교, 자산, 군사, 민심 순), 4~7열은 답변 2에 대한 포인트 변화량을 담고 있음.

- 2차원 배열

스크린샷, 텍스트, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**(3) 일차와 각 포인트를 출력하는 함수 선언**

- 시각적 요소를 통해 4가지 종류의 포인트를 보기 쉽게 출력해주는 함수를 선언. 뒤에 있을 while문의 한 이터레이션마다 일수를 추가해주며 UI를 출력해주는 함수를 호출하여 게임 플레이의 기본 바탕을 만듦.

- 함수, 반복문, 조건문, 입출력

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

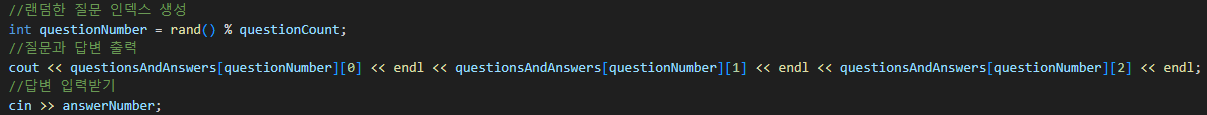
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**(4) 대답 입력 받기**

- 위에서 선언한 랜덤 인덱스를 통해 무작위 질문을 선정하고 그 질문과 답변을 선택한 뒤 대답을 입력 받음.

- 입출력, random함수

**(5) 대답에 대한 포인트 증감 및 출력**

- 질문에 대한 대답이 1, 2인데, 위에서 저장한 pointChange배열을 활용하기 위해 대답이 1일 때 0, 2일 때 4로 초기 인덱스를 설정. 이후 각 포인트에 배열에 있는 증감량 만큼 + -하여 저장. 이후 포인트 증감 출력.

- 배열, 조건문

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**(6) 증감된 포인트 0이면 게임 종료**

- while문 마다 포인트를 확인하여 0 이하로 떨어지면 문구 출력 후 break를 통해 게임 종료.

- while반복, 조건문

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**(7) 특정 일차가 넘어가면 게임 종료**

- while문의 초반에 UI출력 전 dayCount가 특정 수를 달성하면 승리 문구 출력 후 break를 통해 게임 종료.

- 조건문

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

**(8) 특수 질문, 일반 질문을 구분하는 클래스 추가**

- 업캐스팅을 통해 주요 질문 객체 question에 일반, 특수 질문을 구분하여 객체 대입. 중복 코드 및 불필요한 조건문 삭제. 일반, 특수 질문은 질문부터 포인트 변화량까지 모두 다르므로 상속을 통해 중복 제거. 아래 캡처본은 모두 헤더파일에서 선언.

- 클래스, 상속, 다형성, 업캐스팅, 헤더파일 분리

헤더파일 (Question.h)

**텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**main code (project.cpp)**

특수 질문 객체는 일단 다루지 않고 일반 질문 객체로 구성. 아래 코드 외에도 클래스에 맞도록 코드 수정.

****

****

****

**텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명**

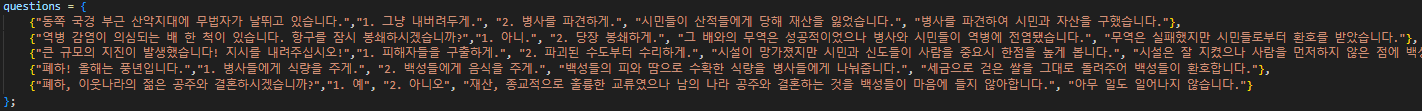
**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**9) 답변에 대한 문구 출력**

- questions 벡터에 대답에 대한 문구를 추가(index 3, 4에 추가). 대답에 대한 인덱스에 따라 해당 문구 출력.

- std::vector, 입출력

****

****

**10) 특정 포인트 양에 따라 출력되는 특수 질문 추가**

* **헤더파일 (Question.h)**

4가지 포인트 양에 따라 생기는 특수 질문 추가. 부모 클래스 Question의 필드에 2차원 string 배열 questions와 2차원 int 배열 point\_chagne를 선언하여 GeneralQuestion과 SpecialQuestion클래스에 각자 적용. GeneralQuestion 클래스에서는 포인트에 따른 질문 확률이 같으므로 그대로 적용. SpecialQuestion 클래스의 questions는 현재 포인트에 따른 condition 문자열을 통해 상태 파악 후 출력하기 위해 생성자에 questions 조건 추가 (r+이라면 r+인 질문들만 questions 배열에 들어감)

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**-메인 코드 (project.cpp)**

현재 포인트 상태를 나타내주는 string함수 ConditionCheck 선언. 포인트에 따라 r+, r-, m+ … 등의 문자열 반환. 정상적인 포인트 상태라면 “normal” 반환

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

condition 변수에 ConditionCheck 함수 리턴값 저장. normal이라면 question 객체를 GeneralQuestion 객체로 지정. normal이 아니라면 특정 확률(코드에선 1/4)을 설정하여 question을 SpecialQuestion 객체로 지정. 이때 condition 문자열에 따라 객체의 questions가 정해짐 (r+이라면 r+인 질문들만 questions 배열에 들어감)

- 헤더파일, 함수화, std::vector

**텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**11) 질문 2차원 벡터 텍스트파일 생성 후 불러오기**

- 질문이 매우 많고 길기 때문에 각 questions.txt, special\_questions.txt 파일로 일반 질문과 특수 질문을 |기준으로 인덱스를 나누어 작성. 이후 파일을 불러와 questions 변수에 저장. 저장 과정이 기므로 부모 클래스 Question에 함수 구현.

**텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명**

이후 각 클래스 생성자에서 질문 변수 초기화

****

****

텍스트 파일은 아래와 같이 질문, 답변1, 답변2, 답변 1에 대한 대답, 답변2에 대한 대답으로 작성되어있음.

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

**12) 이미 나왔던 질문인지 검사**

이미 나온 질문은 다시 나올 수 없도록 메인과 헤더파일의 구조 전체를 바꿈.

- 클래스, 다형성, vector, 파일 입출력

- question.h 헤더파일

부모 클래스에 질문 벡터의 중복된 질문을 제거하기 위해 DeleteQuestion 가상 함수 선언.



GeneralQuestion 클래스의 생성자로 questions와 point\_change를 초기화. DeleteQuestion을 오버라이딩하여 매 질문마다 해당 질문에 해당하는 questions, point\_change 행을 삭제하여 저장.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

마찬가지로 SpecialQuestion의 생성자에서 questions와 point\_change\_list 초기화. 뒤에 \_list가 붙은 이유는 GetQuestions에서 questions의 5번 인덱스인 condition에 따른 특수 질문을 반환하고, 그 질문 리스트에 맞는 포인트 변화량 리스트를 저장 해야한다. 메인에서 point\_change라는 변수명으로 접근하고 있으므로 초기화를 SpecialQuestion의 필드인 point\_change\_list로 한다. 삭제 함수 DeleteQuestion은 GeneralQuestion과 같다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

- 메인 코드 (project.cpp)

질문이 나올 때마다 객체를 새로 생성하므로 while문 밖에서 객체 변수를 선언하여 질문 삭제 및 저장을 위한 발판 마련

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

중복 질문이 계속 삭제되므로 condition에 따라 general\_question과 special\_question 객체의 questions가 비어 있는지 유효성 검사. 이후 조건에 맞는 객체를 question 포인터에 저장.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**13) 일반 질문 추가 및 포인트 변화량 파일 입출력화**

일반질문의 개수를 5개 -> 20개로 추가하여 코드 테스트에 더 유리하도록 변경. 질문이 증가함에 따라 point\_change 변수의 길이도 증가하여 questions와 마찬가지로 InitializePointChange 함수를 만들어 text파일로부터 불러와 저장. 일반, 특수 객체 모두 사용하므로 부모 클래스에 선언 및 정의.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명



텍스트 파일은 아래와 같이 |를 구분으로 포인트 변화량이 담겨있음.

스크린샷, 폰트, 텍스트, 번호이(가) 표시된 사진

자동 생성된 설명

**14) 의도치 않은 특수 질문 empty로 인한 while문 break 수정**

condition에 따른 특수질문이 없을 때 while문을 break하니, 일반 질문이 많이 남았음에도 게임이 종료되는 현상이 있었음. condition에 해당하는 특수질문이 비어있다면, question을 GeneralQuesiton 객체로 바꿔 갑자기 종료되는 현상 수정.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**15) 특수질문 다양화를 위한 ConditionCheck함수 수정**

만약 두 가지 컨디션이 동시에 발생하면 항상 함수에 선언된 순서대로 return하여 랜덤성이 적어 문제가 있었음. 각 포인트를 체크하고 vector에 넣은 후 난수를 생성하여 주어진 컨디션 중 하나를 랜덤하게 선택하여 return하도록 수정.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**16) 특정 포인트가 0이되면 바로 종료하지 않고 특수 질문 출력 후 종료**

- 게임에 더 흥미를 주기 위해 포인트가 0이되면 재밌는 특수질문을 출력하여 몰입감 증폭을 위함.

먼저 특수 질문에 r--등을 컨디션으로 가지는 질문을 추가하고, ConditionCheck에 r--등의 컨디션을 추가한다. 일반 컨디션과 다르게 바로 리턴한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이후 질문의 객체를 설정하는 부분에서 바로 CheckZeroPoint함수를 통해 포인트가 0이 되었는지 확인 후 SpecialQuesiton 객체로 설정해준다. (포인트가 0이라면 컨디션이 r--, m--등 포인트 0일 때의 컨디션이므로 관련 질문만 나오게 된다.) CheckZeroPoint도 게임 종료 문구가 출력되지 않도록 수정한다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

이후 CheckZeroPoint()를 통해 break하는 부분을 포인트 수정하는 부분 직전에 넣어 포인트가 바뀌기 전에 break한다.



그리고 게임이 끝났을 때 최종 스코어를 출력해줘 게임이 종료된 이유를 알 수도 있게 하였다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**4. 테스트 결과**

**(1) 일차와 각 포인트를 출력하는 함수 선언**

- 시각적 요소를 통해 4가지 종류의 포인트를 보기 쉽게 출력.

게임 시작 문구 출력.

텍스트, 스크린샷, 블랙, 폰트이(가) 표시된 사진

자동 생성된 설명

**(2) 대답 입력받기**

-랜덤 인덱스에 따라 무작위 질문과 그에 대한 답변 출력.

사용자로부터 입력을 받음.

텍스트, 폰트, 타이포그래피, 디자인이(가) 표시된 사진

자동 생성된 설명

**(3) 대답에 대한 포인트 증감 및 출력**

- (2)에서 사용자의 입력이 들어오면 그에 대한 포인트 증감 출력

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

**(4) 증감된 포인트 0이면 게임 종료**

**- 4가지 포인트 중 하나라도 0이되면 관련 문구 출력 후 게임 종료**

**텍스트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명**

**(5) 특정 일차가 넘어가면 게임 종료**

**- 일차가 10일을 넘어가면 승리 문구 출력 후 게임 종료**

**텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명**

**5) 답변에 대한 문구 출력**

**텍스트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명**

**6) 특정 포인트 양에 따라 출력되는 특수 질문 추가**

-각 포인트가 9이상이거나 2이하라면 확률에 따라 특수 질문 등장.

**텍스트, 스크린샷, 폰트, 흑백이(가) 표시된 사진

자동 생성된 설명**

**7) 이미 나왔던 질문인지 검사**

- 이미 나왔던 질문은 다시 나오지 않음.

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

**8)** **특정 포인트가 0이되면 바로 종료하지 않고 특수 질문 출력 후 종료**

**- 민심포인트가 0이 된 후 특수 질문이 나오는 것을 확인함.**

텍스트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

**5. 계획 대비 변경 사항**

**1) 특수 질문, 일반 질문을 구분하는 클래스 추가**

- 이전: 하나의 소스파일에 모든 질문과 특수질문을 담아 구분하려고 함.

- 변경 후: 클래스를 통해 포인터에 특수, 일반 질문을 담아 코드의 중복을 줄임.

- 특수질문일 경우와 아닐 경우를 매번 조건문을 통해 구분하면 코드가 복잡해지고 유지 보수에 불리할 것 같아 클래스로 구분하여 가독성, 유지보수성 등을 향상시키기 위함.

**2) 이미 나왔던 질문인지를 검사하는 기능을 나중에 구현**

- 이전: 이미 나왔던 질문인지를 검사하는 기능 구현 예정

- 변경 후: 답변에 따른 문구 출력을 먼저 구현

- 두 가지 클래스를 다루므로 한 가지 클래스 경우에 먼저 적용시킨 후 두 번째 클래스를 이에 맞게 구현하기 위함. 나왔던 질문인지를 검사하는 필드도 클래스에 적용시킬 수 있을 것 같아 뒤로 딜레이함.

**6. 느낀점**

처음으로 코딩을 직접 처음부터 끝까지 다 해보며 프로그램을 만들어 보았습니다. 게다가 이렇게 주기적으로 코드를 수정하며 유지 보수하는 과정은 정말로 처음 해보는 작업이었습니다. 유지 보수하는 과정 없이 그냥 단편적인 프로그램을 작성할 때는 ‘그냥 프로그램이 돌아가기만 하면 되지’ 라는 생각으로 뒤 생각 없이 막 갖다 붙여도 아무런 문제가 없었습니다.

하지만 유지 보수라는 코딩에서 매우 중요한 조건이 추가되니 비교도 할 수 없도록 크게 난이도가 올랐다고 느꼈습니다. 그래서 그만큼 클래스와 상속의 중요성을 크게 깨닫게 되었습니다. 처음에 일반 질문에 대해서만 생각하다가 특수 질문을 추가하겠다는 생각을 했을 때는 그저 배열을 두개 만들고, if문을 반복해서 넣으면 뭐 되겠지 라는 안일한 생각으로 코딩을 했는데 중간에 특수 질문과 일반 질문의 조건이 다른 부분이 생각보다 많아서 if문도 그 만큼 많아졌습니다. 그래서 가독성이 떨어졌고, 무엇보다도 프로그램을 업그레이드하기 위해 보수하는 과정에서 깨나 애를 썼습니다.

그러다가 수업 중에 클래스와 상속을 배우게 되었습니다. 물론 1학기 java수업때도 클래스와 상속을 배웠으나, 직접 프로그램을 만들어보지 않았기 때문에 대체 왜 상속을 하는지, 다형성을 구현해야 하는지 등을 이해하지 못했습니다. 하지만 이번 기회에 배운 내용을 토대로 일반 질문과 특수 질문은 중복되는 점이 많으므로 Question이라는 부모 클래스를 상속받도록 프로그램을 제작해보니 눈에 띄게 유지 보수성이 증가했음을 깨달았습니다. 클래스와 상속의 중요성을 크게 깨달았고, 이 부분이 프로젝트를 진행하면서 가장 크게 느낀 점이 아닐까 싶습니다.

실제로 마지막 작업인 0포인트에서만 출력되는 질문을 출력하도록 구현하는 과정에서도 처음에는 머리가 복잡했지만, 결국 이것도 특수 질문이기 때문에 SpecialQuestion객체를 이용할 수 있지 않을까 싶어서 condition을 추가하고 text파일에 추가한 condition에 맞는 질문만 추가했을 뿐인데 쉽게 구현할 수 있었습니다. 이 프로그램은 저 혼자 만든 것이기에 객체, 함수 등을 마음대로 사용했지만 팀 프로젝트라면 또 어떻게 달라질지 궁금합니다. 교수님께서 팀 프로젝트의 중요성을 강조하신 만큼 3학년에도 수업에 참여하여 좋은 경험을 해보고 싶습니다.