

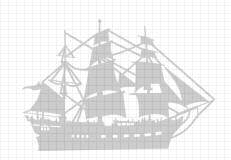
객체지향 개념을 적용한 프로젝트 보고

- 4조
- 1760031 서장일
- 1960025 양승현
- 2160009 명승훈
- 2260006 김상주

CONTENTS

- Team 4. Project voyage 소개
- · Project voyage 란?
- 프로그램 소개
- 브레인 스토밍
- 알고리즘
- 팀원별 역할 분담
- 변화된 점
- 소감





TEAM 4. PROJECT VOYAGE











· Project voyage 란?

- 프로젝트 명칭
- Project_Voyage 대항해시대 미니케임
- 프로젝트 목적
- Project Voyage는 C언어를 사용한 UnchartedWaters_by_SJ를 C++ 언어를 사용하여 트랜스 파일링하고, 객체의 개념을 사용하여 프로그램을 한층 더 고급화함에 목적이 있습니다.
- 프로그램 설명
- 주어진 재화로 상품을 구매, 판매하며 재화를 불려 나가는 간단한 미니게임입니다.
- 난이도를 선택할 수 있으며, 엔딩 조건이 여러 개로 나뉘어 있어 각자의 방식으로 엔딩을 만들 수 있는 게임입니다.



• 프로그램 소개

- 프로그램 소개
- 대항해시대란 15~17세기 유럽인들이 발달한 항해술로 전 세계로 진출하여 항로를 발견하고 세계 일주를 하는 등의 시대를 뜻합니다.
- 이 시대는 유럽 국가들이 새로운/대륙을 발견하고 식민지를 확장하는 등의 활동을 펼친 시대입니다.
- 이 프로그램은 이 대항해시대를 배경으로 한 일본 코에이(KOEI) 사의 비디오 게임 "Uncharted Waters, 大航海時代" 시리즈에서 착안하여 제작된 무역 미니게임을 객체지향의 개념을 적용하여 업그레이드한 프로그램입니다.



• 프로그램 소개

• 프로그램 세계관

• 이국적인 향신료인 후추를 사용한 음식들이 유럽 상류사회에 유행으로 퍼져나가고 있다. 그러나 후추를 판매하고 있는 아랍 중계상들은 인도의 후추를 많이 운반하지도 싸게 팔 생각도 없어 보인다. 이로 인하여 유럽의 후추 가격은 기하급수적으로 상승하게 되었다. 이러한 상황을 타개하기 위해 몇몇 국가들은 인도를 찾기 위한 신항로 개척에 나서게 되는데…

• 프로그램 스토리

• 신항로 개척의 임무를 맡은 당신은 인도를 찾기 위한 여정을 떠나게 됩니다. 고국을 떠나 낮선 장소, 적대적 상황, 당신은 이러한 상황을 극복하고 부를 손에 넣으세요.

• 1차 브레인 스토밍

- 물자보급
- - 랜덤이벤트
- - 과금(재화관련, 선박관련, 명성증가)
- - 입항허가에 따른 영역 확장
- - 마지막 랭킹
- - 물자 보급 및 선원고용
- - 질병 발생 및 치료기능
- - 선박 운용에 필요한 최소 인원보다 선원이 부족하면 항해속도 감소 디버프
- - 항해중 움직임 UI
- #엔딩 이벤트(종료 조건)
- - 세계일주 (명성이 높을 경우 활성화되는 특별 엔딩)
- - 보스와의 싸움(영역이 확장됨에 따라 특정 위치에서 발생하는 보스전)



1차 브레인 스토밍

- # 랜덤 이벤트(항해중 발생)
- - 해적을 만난다.
- - 해적에게 상납금을 준다. -> 풀려난다.
- - 해적에게 상납금을 주지 않는다. -> 사망한다.(Bad)
- - - 맞서싸운다. -> 이겼다 -> 전리품 획득
- - 도망친다. -〉 과정에서 사상자가 발생 -> 선원감소 -> 조건에 따라 디버프
- 목품을 만난다.
- 침몰 -> 무인도로 떠밀려 간다. -> 행방불명(Bad)
- - 폭풍을 만난다. -> 선원이 쓸려갔다. -> 선원 감소
- - 선상반란
- 제압 -> 선원감소 및 명성+1
- - 세압실패 ->∕사망(Bad)



• 1차 브레인 스토밍

- 쿠데타 -> 시세변환
- - 괴혈병 발생
- 라임쥬스가 있다 -> 치료 && 명성+1
- 라임쥬스가 없다 -> 선원감소 && 명성-1
- 선장이 걸린다 -> 사망(bad)
- # 엔딩 마지막에 랭킹
- - id, 보유금액, 명성, 과금한 아이템 수

• 2차 브레인 스토밍

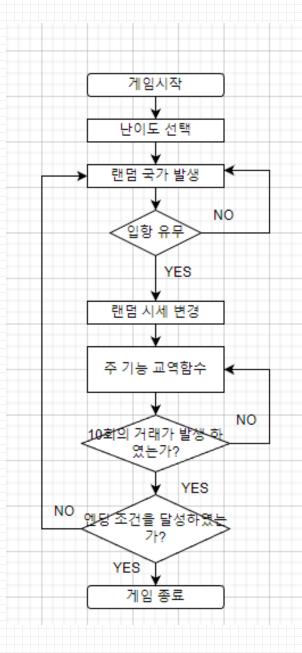
- - 로그인 or 저장 및 불러오기 시간문제로 실패
- - 파일처리를 통하여 특정 플레이어가 얻은 재화(변수)를 저장하고 불러오기 위함
- - 스테이지(입항허가)
- - 입항허가에 따른 영역 확장
- - 특정 조건(변수)가 만족되면 추가되는 스테이지로 구현
- - 예시 : 국가(무역항) 3개가 게임 시작시 기본으로 주어진다. 이 때, 플레이어가 축적한 재화(변수)로 입항허가서(변수)를 구매하게 된다면, 다음 스테이지를 불러오고 다음 스테이지에는 기존 국가(무역항) 3개와 추가되는 무역항(n개)가 존재하게 된다.
- - 엔딩 스테이지(엔딩 이벤트)
- - n번의 스테이지가 지나고 등장하는 가장 마지막 스테이지이며 직전 스테이지에서 재화(변수)를 사용하여 엔딩 스테이지를 구매하게 되면 불러오는 스테이지이다.
- - 엔딩 스테이지는 보수(해적)과의 싸움이 있으며, 이 스테이지는 통상적인 몬스터 처치 게임처럼 작동하도록 구현한다.



• 1차 브레인 스토밍

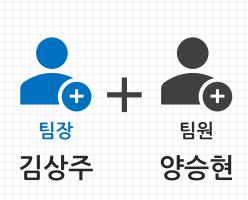
- 쿠데타 -> 시세변환
- - 괴혈병 발생
- 라임쥬스가 있다 -> 치료 && 명성+1
- 라임쥬스가 없다 -> 선원감소 && 명성-1
- 선장이 걸린다 -> 사망(bad)
- # 엔딩 마지막에 랭킹
- - id, 보유금액, 명성, 과금한 아이템 수

• 알고리즘





- 1. 국가(Country) Class와 객체 생성
- - 객체 내용
- 국가 명칭
- 기역 설명
- - 1번 상품 명칭
- 2번 상품 명칭
- 1번 상품 설명
- 2번 상품 설명
- 1번 상품 가격
- 2번 상품 가격



• 2. f_Commerce 함수의 틀 구성

- 국가명, 지역설명 출력
- 교역 정보 출력
- ' 상품명, 상품 설명, 상품 가격
- - 보유 골드
- - 메뉴 선택
- - switch-case 사용
- 구매 함수
- - 판매 함수
- - 보유 물건 확인 함수
- - 다음 항구 함수
- - 저장 및 종료 함수





- 3. 메인 함수 내 do-while 틀 구성
- do
- · {
- - 변수 초기화
- - 입항 유/무 선택 함수
- - 랜덤 시세 변경(이전 프로그램과 다르게 입항후, 시세 변경이 되지 않도록 while 밖 위치)
- while(조건)
- {
- - 교역 기능 함수
- - 다음 항구로 넘어가는 조건
- - 종료 이벤트 조건
- }
- }while(조건);





- 1. C to C++ 트랜스 파일링 및 엔딩 추가
- - 엔딩 조건 추가
- · 게임 클리어 조건 추가
- - ex) 특정 물품 개수 이상 종료
- - ex) 특정 금액 이상 종료 등
- - 게임 종료 함수 변경
- 게임 개요 출력 변경
- 난이도 선택 함수 변경





• 1. 교역 정보 출력

- - 현재 국가
- · 기본적인 교역 정보 출력
- 국가, 보유 골드, 지역 설명
- 교역품 1, 2번에 대한 가격과 설명
- 구매 할 수 있는 물품 나누기(1번 구매, 2번 판매)





- 2. 랜덤 시세 변경
- · 현재 국가의 객체 가져오기
- - 상품 1, 2번 (+/-)랜덤 변경, while 밖에 존재
- 3. 구매한 교역품 보기
- 현재 가지고 있는 교역품 출력
- 4. 항구 이동
- 변수를 변경하여 반복문을 탈출



• 변화된 점

• 구조체 사용

• 객체 사용

```
ClassCountryInfo(
   string CountryName = "국가 명칭",
   string RegionDepiction = "지역 설명",
   string FirstProductName = "1번 상품명칭",
   string SecondProductName = "2번 상품명칭",
   string FirstProductDepiction = "1번 상품설명",
   string SecondProductDepiction = "2번 상품설명",
    int FirstProductPrice = 0.
    int SecondProductPrice = 0) :
   strCountryName(CountryName),
   strRegionDepiction(RegionDepiction),
   strFirstProductName(FirstProductName),
   strSecondProductName(SecondProductName),
   strFirstProductDepiction(FirstProductDepiction),
   strSecondProductDepiction(SecondProductDepiction),
   nFirstProductPrice(FirstProductPrice),
   nSecondProductPrice(SecondProductPrice) {}
~ClassCountryInfo() {}
```

PROJECT

• 변화된 점

포인터 매개변수 사용

```
country0->goods1_price = country->goods1_price;
country0->goods2_price = country->goods2_price;
country0->goods3_price = country->goods3_price;
country0->goods3_price = country->goods3_price;
strcpy(country0->country_name, country->country_name);
strcpy(country0->goods1_name, country->goods1_name);
strcpy(country0->goods2_name, country->goods2_name);
strcpy(country0->goods3_name, country->goods3_name);
```

• 상속 개념 사용



• 변화된 점

문자열 대입

객체화

```
ClassCountryInfo GameData;
ClassPlayVoyage GamePlay;
ClassUIVoyage GameUI;

ClassCountryInfo C1

(
 "에스파냐",
 "유럽의 이베리아 반도와 북아프리카에 위치한 국가로, 무적함대로 대표되는 강력한 해양 국가이자 대영제국 이전의 해가 지지 않는 나라로까지 여겨졌던 세계적인 열강",
 "쉐리",
 "샤프란",
 "에스파냐 남동부의 헤레스도라 지방에서 만들어지는 감미포도주이다.",
 "샤프란의 암술을 말린 굉장히 고가인 항신료이다.",
 1000, 1000
);
GamePlay.f_AddCountry(C1);
```





• 메인 함수내 구현

```
do //조건까지 랜덤으로 국가가 변경됨
{
    srand((long)time(NULL));
    random_country = rand() % COUNTRY_NUM + 1;

    while (sailing)
    {
        f_trade(country[random_country], country[0]); //트레이드(교역) 함수호출
    }
    sailing = 1;
    game_off_num++;

} while (game_off_num != 2); //10번의 국가 변경이 발생할 때까지 실행

f_game_clear(); //게임 클리어 함수호출
Sleep(3000);
system("cls");
f_ending(); //엔딩 함수호출
```

• Class 사용

```
do //조건까지 랜덤으로 국가가 변경됨
{
GamePlay.f_ArrayProdcutInit(); //상품 선택을 위한 이중배열 초기화
//srand((unsigned)time(NULL)); //랜덤 seed 변경을 위한 초기화
//GamePlay.nRandomCountry = rand() % MAXCOUNTRY; //랜덤으로 국가 변경
GamePlay.f_EnterPort(); //입항 유/무 선택 함수
GamePlay.f_RandomProductPrice(); //랜덤으로 시세 변경

while (sailing)
{
    //GamePlay.f_RandomProductPrice();
    GamePlay.f_Commerce(); //교역 함수 호출
    nGameClearNumber++;

    if (nGameClearNumber == 10) // 다음 항구로 넘어가는 조건
    {
        GamePlay.f_PrintNextCountry(); // 다음 함수로 이동한다는 구문 출력
        Sleep(3000);
        GamePlay.f_ActNextCountry(); //다음 항구 함수호출
        cout << "다음 항구 함수 호출";
    }
}
sailing = 1;
```

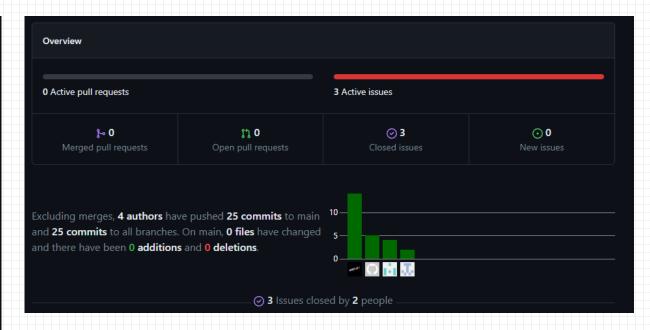
ClassCountryInfo GameData; ClassPlayVoyage GamePlay; ClassUIVoyage GameUI;



• GitHub 사용

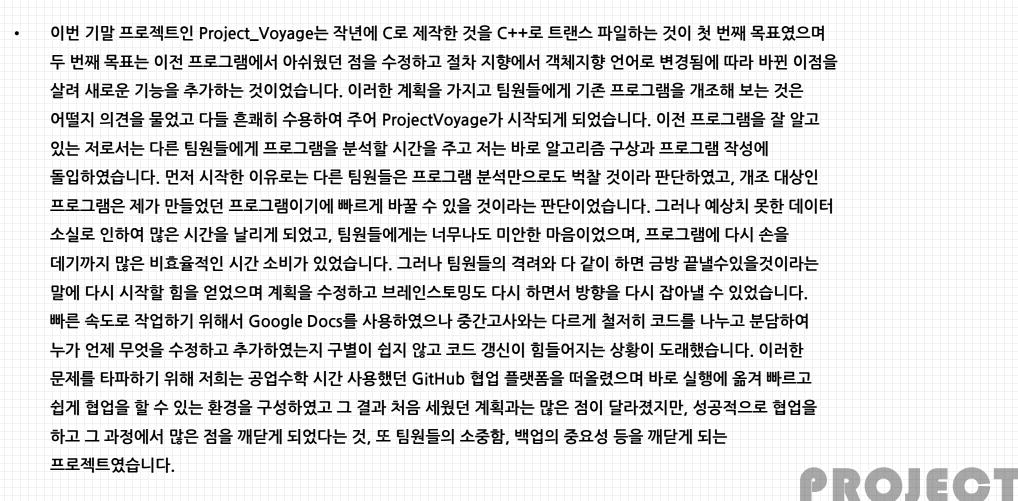


□ ⊙ 0 Open ✓ 3 Closed
□ ⊘ 구매 물품 보기 함수 작동불능 bug help wanted #3 by Hheney was closed 3 days ago
□ ⊘ 랜덤 시세 변경함수 버그발생 bug #2 by Hheney was closed 3 days ago
f_actNextHarbor() Function Inoperative bug good first issue #1 by Hheney was closed 3 days ago





• 소감



VOYAGE



• 소감

이번 기말 Term-Proiect를 하면서 굉장히 느낀 점이 많습니다. 처음에는 각자 할당받은 파트를 구현하기 위해 함께 디버깅해가며 수정하는 과정을 거쳤습니다. 그 수정본을 통합하기 위해 저희는 구글 닥이라는 플랫폼을 사용했는데 수정된 내용을 통합하기에 너무 범위가 방대하고 오류 난 부분을 찾아내는 게 너무 부담스러웠습니다. 그렇게 낙담하던 도중에 개발자들이 많이 사용한다던 깃허브라는 플랫폼이 생각났습니다. 실제로 게임 프로그래밍 입문 1이라는 수업에서 조용희 교수님께서 사용하실 때 코드 수정하기 수월해서 대략 일주일이 안 남은 시점에 깃허브를 사용했습니다. 기존에 제작한 코드들을 과감하게 쳐내고 새로 통합하면서 팀장인 김상주가 기반을 다졌고 저희는 각자 맡은 부분을 재수정하고 제작하면서 기말 프로젝트를 구현할 수 있었습니다. 물론 진행하는 과정에서 시행착오도 많았고 난관에도 봉착해서 분위기가 안 좋다고 느껴질 때도 있었지만 항상 교수님께서 강조하시는 '협업'을 생각하며 서로 끌어주고 당겨주며 진행하다 보니 큰 어려움 없이 헤쳐나갔습니다. 저는 이번 프로젝트에서 2가지를 깨달았습니다. 첫 번째는 소통입니다. 본인이 할당받은 코드가 끝났다고 전부인 게 아니라 어느 부분에서 수정했으며 팀원 모두가 코드 한 글자라도 바뀐 걸 알아야 그다음 스텝이 가능합니다. 두 번째로는 백업입니다. 백업을 상시 받아놓는 편이지만 만에 하나 무슨 일이 생길지 모르기 때문에 철저하게 백업해야 한다고 뼈저리게 느꼈습니다. 끝으로 프로그램 제작 과정 중에 제 역량에 한계가 분명하게 느껴져 어려움을 겪을 때도 격려해 주며 여기까지 이끌어준 저희 팀장과 팀원들에게 큰 감사를 표하고 싶습니다.





• 소김

 기말 프로젝트를 하면서 처음엔 이걸 언제 다하지?라는 생각과 내가 할 수 있을지에 고민과 스트레스를 받았지만 막상 시작을 하고 나니 내가 예전보다 성장했다는 걸 느꼈고 또 막히는 부분은 팀장인 상주형이 알려주며 이해하지 못하면 다시 차근차근 설명해 주며 최대한 이해시켜주려는 모습이 듬직했다 또 이번에도 팀원들이 참여를 잘 해주어 편했다 항상 새벽까지 같이 참여를 해준 팀원들과 우리를 이끌어준 팀장님께 감사를 전하고 싶다.





• 소감

• 수업 시간에 배운 내용을 기반으로 좀 더 심화된 문제들을 풀어 나가다 보니 실력 향상이 조금은 되었다고 느꼈고, 진행 과정에 있어서 유능한 팀원들 덕에 보다 수월하게 진행되어서 좋았고 팀원들보다 한참 모자란 실력 탓에 크게 기여한 것은 없지만 잘 이끌어준 팀장, 팀원들에게 감사합니다.





```
int main()
{
    printf("이상 4조의 PROJECT VOYAGE 발표 마치겠습니다.\n");
    printf("감사합니다!");
    return 0;
}
```