수업명

수요의 카페

프로젝트 제안서

제출일자: 12/01
제출자명: 황윤규
제출자학번: 214968

1. 프로젝트 목표

1) 배경 및 필요성 (14 pt)

고등학생 시절 게임이론을 게임으로 만들어 보여주는 게임을 해본 적이 있습니다. 특정 학문의 이론을 게임으로 접하게 되면서 재미도 있고 글로만 접하는 것이 아니라 더 이해하기 쉬웠던 경험이 있습니다. 이에 착안하여 제가 1학년 시절 경제학개론에서 어려움을 겪었던 기억이 있습니다. 그중에서 가장 핵심이었다고 생각이 드는 수요 공급 곡선과 관련된 내용 재미있게 만들면 좋겠다는 생각이 들어 게임으로 만들고자 합니다. 컨셉으로는 카페를 경영하는 게임으로 수요 공급 곡선에서 가격과 공급자가 원하는 가격인 균형지점이라는 곳이 있습니다. 이때 플레이어는 카페 메뉴를 균형지점에 해당하는 가격을 찾아내어 돈을 가장 많이 버는 것을 목표로 하는 게임입니다. 거기에 다른 추가적인 베블런 효과나 외부효과와 같은 것들을 추가하여 색다른 재미와 동시에 더 추가적인 효과를 기대할 수 있습니다.

2) 프로젝트 목표

수용 공급 곡선의 핵심 개념인 가격이 오르면 소비자의 수요가 줄어들고 가격을 내리면 수요가 증가한다는 개념을 게임을 통해 전하는 것을 목표로 합니다. 그리고 후에 엔딩에서 해당 개념을 설명해줌으로써 더 쉽고 재미있게 전하고자 합니다. 게임의 몰입도는 해치지 않되 게임을 진행함으로써 재미있고 유익한 시간이 되도록 하고 싶습니다.

3) 차별점

기존에 학습을 목표로 한 게임들은 대다수 그렇게 인지도를 가지지 못하였다고 알고 있습니다. 제가 느끼기에는 이런 게임들이 큰 반향을 일으키지 못한 까닭은 너무 학습에만 목표를 잡고 교육의 개념을 게임 내에 지속적으로 노출시켜 게임의 몰입감이 떨어졌기 때문이라고 생각합니다. 따라서 재미를 줄 수 있는 게임에 일종에 요소로 개념을 집어넣고 이를 핵심으로 두면 자연스럽게 해당 개념을 접하게되고 원리를 이해시킨 후에 엔딩에 이와 관련된 내용을 알려줌으로써 게임은 게임대로 몰입감을 더하고 마지막에 가르침도 얻을 수 있도록 하는 점이 차별점이라고 생각합니다.

2. 기능 계획

1) 기능 1 (플레이어 행동)

- 설명 플레이어의 행동 요소로 선택하여 행동하는 기능입니다.

(1) 세부 기능

- 플레이어의 행동은 낮과 밤으로 나뉘어지고 행동 요소가 달라집니다.

낮 : 1. 카페 메뉴 만들기 ,2. 디저트 만들기 2. 카페 메뉴 확인, 3. 정산 , 4. 미니게임, 5, 판매 시작

밤 : 1. 메뉴 가격 수정 , 2. 직원 고용 , 3. 카페 메뉴 확인, 4. 정산, 5. 신문 확인하기, 6. 잠자기

==========(낮)==========

1) 기능 2 (카페 메뉴 만들기)

- 설명 : 카페의 메뉴의 이름과 가격을 입력받아 메뉴를 제작합니다.

(1) 세부 기능

- 메뉴의 가격과 이름을 입력받게되면 수요 공급 곡선에 따라서 가격에 맞추에 판매량이 결정 됩니다.
- 후에 플레이어 행동 양식(낮) 5번 판매 시작 시에 정해진 판매량에 따라 판매됩니다.

1) 기능 3 (디저트 만들기)

- 설명 : 디저트의 이름을 입력받아 메뉴를 제작합니다.

(1) 세부 기능

- 디저트는 커피에 절반만큼 팔립니다.

- 하루에 한번 나오는 신문에서 나온 특정 단어를 조합하여 만들면 더 많이 팔립니다.

- 디저트는 한 종류만 제작이 가능합니다.

1) 기능 4 (카페 메뉴 확인)

- 설명 : 현재 존재하는 카페 메뉴를 모두 보여줍니다.

(1) 세부 기능

- 메뉴의 이름과 가격을 보여줍니다.

- 만일 메뉴가 이미 한번 판매를 한 전적이 있다면 판매량또한 보여줍니다.

- 이는 후에 플레이어 행동 양식(밤)에 1번 메뉴 가격 수정시에 용이하게 사용됩니다.

1) 기능 5 (정산)

- 설명 : 지금까지 판매한 메뉴들의 총 판매량을 모두 합산하여 보여줍니다.

(1) 세부 기능

- 현재까지 판매한 판매량을 종합하여 확인할 수 있도록 합니다.

1) 기능 6 (미니게임)

- 설명 : 미니게임을 실행합니다.

(1) 세부 기능

- 미니게임을 진행하여 성공 시에 플레이에 도움이 되는 아이템을 지급합니다.

1) 기능 7 (판매 시작)

- 설명 : 현재 게시된 메뉴들로 판매를 시작합니다.

(1) 세부 기능

- 현재 게시된 메뉴들로 판매를 시작하되 판매량은 가격에 따라 정해진 양만 판매되어집니다.

- 판매량은 후에 플레이어 행동 양식(낮/밤) 카페 메뉴 확인을 통해 확인이 가능합니다.

==========(밤)==========

1) 기능 8 (메뉴 가격 수정)

- 설명 : 메뉴의 가격을 수정합니다.

(1) 세부 기능

- 메뉴의 가격을 다시 수정할 수 있습니다. 커피는 가격에 따른 판매량도 재조정됩니다.
- 메뉴의 수정은 메뉴별로 한 번만 가능합니다.

1) 기능 9 (직원 고용)

- 설명 : 돈을 일정 소모하여 직원을 고용합니다.

(1) 세부 기능

- 직원 고용 시에 카페 메뉴를 추가로 더 제작이 가능해집니다.
- 직원 고용 시 3명의 직원을 보여주며 고용 비용은 무작위로 나옵니다.

1) 기능 10 (신문 확인하기)

- 설명 : 하루에 한번 신문이 나오게 됩니다.

(1) 세부 기능

- 신문에는 그날 유행하는 단어가 나오게 됩니다.
- 랜덤하게 하나의 신문을 제공하여 줍니다.
- "오늘은 사과가 맛있습니다."와 같은 신문이 나오면 디저트에 사과타르트와 같은 메뉴를 제작하면 더 많이 팔리게 됩니다.

1) 기능 11 (낮과 밤)

- 설명 : 플레이어는 낮과 밤에 따라 행동양식이 달라집니다.

(1) 세부 기능

- 플레이어의 행동은 낮과 밤으로 나뉘어지고 행동 요소가 달라집니다.
- 낮에는 플레이어 행동 양식(낮) 5번 판매 시작을 통해 판매가 끝나면 밤으로 이동합니다.
- 밤에는 플레이어 행동 양식(밤) 5번 잠자기를 통해 낮으로 이동합니다.

1) 기능 12 (시간)

- 설명 : 플레이어가 행동할 수 있는 시간은 7일으로 제한되어 있습니다.

(1) 세부 기능

- 낮과 밤이 1일이며 밤이 지나면 1일이 지나가게 됩니다.
- 7일이 모두 지나게 되면 엔딩이 나오게 됩니다. 수요 공급 곡선에 관련된 내용을 소개해 줍니다.
- 제작자가 정해 놓은 가격을 달성시에 최종 엔딩이 나오게 됩니다.

1) 기능 13 (고객 피드백 시스템)

- 설명 : 판매 이후에 고객들이 메뉴에 대해 피드백을 남깁니다.

(1) 세부 기능

- 긍정 혹은 부정 피드백을 남깁니다.
- 긍정 피드백의 경우에는 균형 가격에 가까울수록 많아집니다. 반대로 멀수록 부정 피드백이 증가합니다.

1) 기능 14 (균형 가격 설정)

- 설명 : 새로운 메뉴를 만들수록 더 비싼 메뉴를 제작하여 균형 가격이 각기 다릅니다.

(1) 세부 기능

- 새로 만든 메뉴일수록 균형가격이 조금 더 비싸게 조정됩니다.

1) 기능 15 (베블런 효과)

- 설명 : 특별한 효과로 특정 메뉴를 가격을 높게 설정했을 때 판매량이 증가합니다.

(1) 세부 기능

- 미니게임에서 특정 이름의 메뉴의 힌트를 제공하며 해당 이름의 메뉴의 경우에는 가격을 높게 설정할수록 판매량이 증가합니다.

1) 기능 16 (외부 효과)

- 설명 : 하루에 한 번 신문이 나오며 그 신문에 단어에 대한 힌트가 제공됩니다.

(1) 세부 기능

- 신문에는 사과가 제철과 같은 단어가 나오면 사과가 들어간 디저트류는 더 많이 팔립니다.

3. 진척 사항

1) 기능 구현

(1) 기능 5 정산

설명

- 설명 : 지금까지 판매한 메뉴들의 총 판매량을 모두 합산하여 보여줍니다.

(1) 세부 기능

- 현재까지 판매한 판매량을 종합하여 확인할 수 있도록 합니다.

적용된 배운내용

1) 입출력 함수

입력과 출력을 합니다.

2) 클래스

클래스 내의 맴버에 대한 접근 지정자를 통하여 접근이 가능하게 합니다.

코드 스크린샷

```
//정산하여 총 가격을 보여줍니다.
void Menu::Total(int total_staff){
    cout << "====================| 정산 |======================" << endl;
    // 커피의 총 판매량을 모두 계산하여 보여줍니다.
    cout << "===| 커피 |===" << endl;
    cout << total_sell << endl;
    cout << "\n";

    // 디저트의 총 판매량을 모두 계산하여 보여줍니다.
    cout << "===| 디저트 |===" << endl;
    cout << dessert_total_sell << endl;
    cout << "\n";

    // 직원 월급의 총 판매량을 모두 계산하여 보여줍니다.
    cout << "===| 직원 월급 |===" << endl;
    cout << total_staff << endl;
    cout << "\n";

    // 커피 + 디저트 - 직원월급을 하여 보여줍니다.
    cout << "===| 정산 |===" << endl;
    cout << total_sell + dessert_total_sell - total_staff << endl;
}
```

(1) 기능 7 판매 시작

설명

- 설명 : 현재 게시된 메뉴들로 판매를 시작합니다.

(1) 세부 기능

- 현재 게시된 메뉴들로 판매를 시작하되 판매량은 가격에 따라 정해진 양만 판매되어집니다.

- 판매량은 후에 플레이어 행동 양식(낮/밤) 카페 메뉴 확인을 통해 확인이 가능합니다.

- 판매 가격에 맞추어 판매 개수를 결정합니다.

핵심 기능으로써 여러가지 공급 곡선을 추가하였습니다.

- 메뉴 제작시에 랜덤으로 수요 공급 곡선을 정하도록 수정하였습니다.

적용된 배운내용

1)for문

for문을 활용하여 만들어진 메뉴들의 수요 공급 곡선을 정하고 이를 판매량으로 결정합니다.

2) 함수

수요 공급 곡선을 정해주는 함수를 호출하여 수요 공급곡선에 맞추어 판매량을 결정하여 줍니다.

3) min 함수

min 함수를 활용하여 수요자의 수요와 공급자의 공급중에 적은 값을 골라 가격이 너무 싸다면 공급자의 공급이 줄어 판매가 줄어들고, 가격이 너무 비싸다면 수요가 줄어 판매가 줄어듭니다.

4) 난수 생성

1~3사이의 난수를 생성하고, 그 난수를 저장하여 3가지의 수요공급곡선을 선택하는데에 사용하게 됩니다.

코드 스크린샷

```
void Menu::SellMenu(){
    // 디저트의 판매량은 커피의 절반만큼 팔립니다.
    // 다만 커피의 판매량은 가격에 따라 바뀔 수 있음으로 0으로 초기화하고 다시 받아줍니다.
    dessert_sell = 0;

    // 메뉴를 전부 돌면서 정해진
    for(int i = 0; i < menu_price.size(); i++){
        SupplyDemand(i);
    }

    // 커피의 모든 판매량의 절반만큼 판매되어 저장합니다.
    dessert_total_sell += 5000 * dessert_sell;
}
```

(1) 기능 8 메뉴 가격 수정

설명

- 설명 : 메뉴의 가격을 수정합니다.

(1) 세부 기능

- 메뉴의 가격을 다시 수정할 수 있습니다. 커피는 가격에 따른 판매량도 재조정됩니다.

- 메뉴의 수정은 메뉴별로 한 번만 가능합니다.

```
void Menu::SupplyDemand(int i){
    int price, demand, supply;
    // 3개의 균형지점 중에 랜덤으로 결정하여 줍니다.
    if(supply_demand_num[i] == 1){
        price = menu_price[i] / 100;
        demand = 100 - price;
        supply = 10 + 2 * price;
    }
    else if(supply_demand_num[i] == 2){
        price = menu_price[i] / 100;
        demand = 150 - price;
        supply = 30 + 2 * price;
    }
    else{
        price = menu_price[i] / 100;
        demand = 80 - price;
        supply = 20 + price;
    }

    // 수요와 공급 중에서 작은 값을 판매량으로 결정합니다.
    int insert = min(demand, supply);

    // 몇개가 판매되는지 판매량을 넣습니다.
    menu_sell[i] = insert;

    // 디저트의 판매량은 커피의 절반만큼 팔립니다.
    dessert_sell += menu_sell[i] / 2;

    // 판매된 총 가격을 입력합니다. (정산에서 이용됩니다.)
    total_sell += menu_price[i] * menu_sell[i];
}
```

적용된 배운내용

1) string

수정할 이름을 받습니다.

2) while문

수정사항을 bool 변수인 menu_check를 통해 수정이 올바르게 종료될 경우에 반복을 종료합니다.

3) for문

입력받은 이름이 메뉴에 존재한다면 가격을 입력받고 그 가격으로 수정하여 줍니다.

4) if문

입력받은 이름이 메뉴에 존재하는지를 판별합니다.

코드 스크린샷

```
void Menu::ModifyMenu(){
    string name;
    int price;
    int modify = -1;
    bool menu_check = true;

    while(menu_check){
        // 제작할 메뉴를 입력받습니다.
        cout << "수정할 메뉴명를 입력해주세요 : " << endl;
        cin >> name;

        for(int i = 0; i < menu_name.size(); i++){
            // 이미 존재하는 메뉴라면 menu_check를 true로 수정이 가능하도록 합니다.
            if(menu_name[i] == name){
                modify = i;
                menu_check = false;
            }
        }

        //수정이 가능합니다.
        if(!menu_check){
            cout << "수정할 가격을 입력해주세요 : " << endl;
            cin >> price;
            menu_name[modify] = name;
            menu_price[modify] = price;
        }
        else{
            cout << "메뉴 이름을 다시 한번 확인해주세요." << endl;
        }
    }

}
```

(1) 기능 9 직원 고용

설명

- 설명 : 돈을 일정 소모하여 직원을 고용합니다.

(1) 세부 기능

- 직원 고용 시에 카페 메뉴를 추가로 더 제작이 가능해집니다.
- 직원 고용 시 3명의 직원을 보여주며 고용 비용은 무작위로 나옵니다.

적용된 배운내용

1) class

class를 사용하여 같은 객체를 여러 개 사용하기 위해서 활용하였습니다.

2) friend

접근 지정자와 관계없이 멤버 변수를 공유하여 함수에서도 사용할 수 있도록 하였습니다.

3) 참조자 매개변수

참조자 매개변수로 활용하여 함수가 Employee 객체를 반환하게 하였습니다.

코드 스크린샷

```
//직원 고용 토대 제작 완료
void Hire(Employee& hire){
    srand((unsigned)time(NULL));

    string yn;
    cout << "======================| 주의 사항 |=========================" << endl;
    cout << "직원 고용을 진행시 고용 여부와 상관없이 그날 고용은 끝이납니다" << endl;
    cout << "주의하여 고용을 진행하여 주시길 바랍니다." << endl;
    cout << "============================================================" << endl;

    cout << "직원 고용" << endl;

    //직원을 돌아가면서 총 3번을 보여줍니다.
    for(int i = 0; i < 3; i++){
        // 가격과 고용비용, 이름을 랜덤으로 뽑습니다.
        int employee_money = rand()%10000 + 1000;
        int menu_count_plus = rand()%3 + 1;
        int k = rand()%9;

        // 저장된 직원의 이름을 랜덤으로 보여줍니다.
        cout << "직원이름 : "  << hire.rand_name[k] << endl;

        // 랜덤으로 고용비용을 소개하여 줍니다.
        cout << "고용비용 : " << employee_money << "원" << endl;

        // 메뉴 추가의 개수도 제공하여 줍니다. (추후 확률 조정 예정)
        cout << "메뉴 추가 개수 : " << menu_count_plus << "개" << endl;

        // 한번만 고용이 가능합니다.
        cout << "해당 직원을 고용하시겠습니까?(Y/N) : ";
        cin >> yn;

        if(yn == "Y" || yn == "y" || yn == "Yes" || yn == "yes"){
            cout << "해당직원을 고용하였습니다." << endl;
            // 고용 시에 만들 수 있는 메뉴 개수 추가
            // 최대 메뉴 개수를 제한하여 추가합니다.
            if(make_coffee_count + menu_count_plus < MAX_MENU){
                make_coffee_count += menu_count_plus;
            }
```

1) 기능 수정

(1) 기능 14 균형가격 설정

설명

주어진 가격에 따라서 정해진 함수를 통해 균형 가격을 설정합니다.

수요와 공급 곡선에서 가격이 줄어들면 수요는 늘어나지만 공급이 줄어듬으로 판매량은 공급된 양만 팔려 공급 곡선에 의해 판매량이 결정되어지고, 가격이 늘어나면

```
        if(yn == "Y" || yn == "y" || yn == "Yes" || yn == "yes"){
            cout << "해당직원을 고용하였습니다." << endl;
            // 고용 시에 만들 수 있는 메뉴 개수 추가
            // 최대 메뉴 개수를 제한하여 추가합니다.
            if(make_coffee_count + menu_count_plus < MAX_MENU){
                make_coffee_count += menu_count_plus;
            }
            else{
                // 메뉴의 최대개수로 제한합니다.
                cout << "최대 메뉴 개수에 도달하였습니다.(10개)" << endl;
                make_coffee_count = MAX_MENU;
            }

            // Employee class에 고용한 이름과 고용 비용 vector에 입력
            hire.employee_name = hire.rand_name[k];
            hire.employee_price = employee_money;
            staff_count++;

            // 현재 가용 메뉴 개수를 출력하여 보여줍니다.
            cout << "현재 가용 가능한 메뉴의 개수는 " << make_coffee_count << "개 입니다." <
            cout << "현재 고용한 직원 수는 " << "(" << staff_count << "/3)" << "명 입니다."
            break;
        }
        else if(yn == "N" || yn == "n" || yn == "No" || yn == "no"){
            cout << "다음 직원을 소개합니다." << endl;
            cout << endl;
        }
        else{
            cout << "다시 한번 입력해주세요." << endl;
        }
    }
}
```

공급은 늘어나지만 수요는 줄어들어 수요 곡선에 의해 판매량이 결정되어집니다.

후에 추가적으로 다른 추가된 메뉴에 한해서 추가적인 조정이 있을 예정입니다.

수정

랜덤으로 지정된 값에 따라서 수에 따라서 수요공급곡선을 선택하여 줍니다.

코드 스크린샷

```cpp
void Menu::SupplyDemand(int i){
    int price, demand, supply;
    // 3개의 균형지점 중에 랜덤으로 결정하여 줍니다.
    if(supply_demand_num[i] == 1){
        price = menu_price[i] / 100;
        demand = 100 - price;
        supply = 10 + 2 * price;
    }
    else if(supply_demand_num[i] == 2){
        price = menu_price[i] / 100;
        demand = 150 - price;
        supply = 30 + 2 * price;
    }
    else{
        price = menu_price[i] / 100;
        demand = 80 - price;
        supply = 20 + price;
    }

    // 수요와 공급 중에서 작은 값을 판매량으로 결정합니다.
    int insert = min(demand, supply);

    // 몇개가 판매되는지 판매량을 넣습니다.
    menu_sell[i] = insert;

    // 디저트의 판매량은 커피의 절반만큼 팔립니다.
    dessert_sell += menu_sell[i] / 2;

    // 판매된 총 가격을 입력합니다. (정산에서 이용됩니다.)
    total_sell += menu_price[i] * menu_sell[i];
}
```

3. 프로젝트 일정 (참고: 간트차트)

업무	11/3	11/10	11/17	11/20	11/27
제안서 작성	(완료)				
기능 1		(완료)			
기능 2		(완료)			

업무	11/10	11/17	11/24	12/1	12/8
기능 3		(완료)			

기능 4		(완료)		
기능 5			---------->	

업무	11/17	11/24	12/1	12/8	12/15
기능 6		---------->			
기능 7		(완료)			
기능 8		(완료)			
업무	11/17	11/24	12/1	12/8	12/15
기능 9			(완료)		
기능 10			---------->		
기능 11			---------->		

업무	11/24	12/1	12/8	12/15	12/22
기능 12			(일부)---------->		
기능 13			---------->		
기능 14			(완료)		

업무	11/24	12/1	12/8	12/15	12/22
기능 15				---------->	
기능 16				---------->	