

# 수업명 수요의 카페

## 프로젝트 제안서

제출일자: 11/03  
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 균형 지점이라는 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

수요 공급 곡선의 핵심 개념인 가격이 오르면 소비자의 수요가 줄어듦과 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 못하였고 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. 미니 게임, 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 맞추어 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 7일로 제한되어 있습니다.

(1) 세부 기능

- 낮과 밤이 1일이며 밤이 지나면 1일이 지나가게 됩니다.
- 7일이 모두 지나게 되면 엔딩이 나오게 됩니다. 주요 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.

## New

### 1) 기능 17 (애니메이션)

- 설명 : 게임의 소소한 재미를 위해 행동을 진행할 때 적절한 형식의 애니메이션을 추가합니다.

#### (1) 세부 기능

- 텍스트 형식의 "자는중..."이 한글자씩 나오는 등의 애니메이션이 추가됩니다.
- 기회가 된다면 ASCII아트 형식의 애니메이션이 추가 가능성을 고려 중입니다.

## 3. 진척 사항

### 1) 기능 구현

#### (1) 기능 1 플레이어 행동

##### 설명

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

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

를 보여주어 플레이어 행동양식을 선택하여 플레이합니다.

##### 적용된 배운내용

#### 1) 클래스

클래스에 플레이어 행동과 관련된 내용을 담아 놓아 후에 유지 보수가 좀 더 쉽도록 하였습니다.

#### 2) switch-case문

switch-case문을 이용하여 플레이어가 행동 양식을 선택했을 때에 들어가는 선택지를 코드 내에서 알아보기 쉽도록 하였습니다.

#### 3) if 문

낮과 밤을 구분하기 위해 time변수를 활용하여 0일때는 낮의 행동양식을 1일 때는 밤의 행동 양식을 출력하도록 if문을 사용하였습니다.

#### 4) has a

다른 클래스를 상속하여 해당 클래스의 함수를 사용합니다.

##### 코드 스크린샷

```
// 행동 요소와 관련된 클래스 입니다. |
```

```
class Run{  
    int day = 1;  
public:  
    // 메뉴 클래스를 참조합니다.  
    Menu menu;  
    Anim anim;  
    void run();  
};
```

```

// 전체적인 행동 요소를 나타내는 함수입니다.
void Run::run(){
    int menu_select;

    int time = 0;

    while(day < 10){
        menu_select = 0;
        if(time == 0){
            cout << "낮 " << endl; |
        }
        else{
            cout << "밤 " << endl;
        }

        cout << day << "일 차 " << endl;

        if(time == 0){
            // 낮에 할 행동을 보여주고 입력을 통해 확인합니다.
            while(menu_select < 5){
                cout << "=====행동 (낮)===== " << endl;
                cout << "1. 카페 메뉴 만들기 " << endl;
                cout << "2. 디저트 만들기 " << endl;
                cout << "3. 카페 메뉴 확인 " << endl;
                cout << "4. 미니 게임 " << endl;
                cout << "5. 판매 시작 " << endl;
                cin >> menu_select;

                switch(menu_select){
                    case 1:
                        // 참조한 Menu클래스의 함수 make_menu를 불러옵니다.
                        menu.make_menu();
                        break;
                    case 2:
                        menu.make_dessert();
                        break;
                    case 3:
                        menu.show_menu();
                        break;
                    case 4:
                        break;
                    case 5:
                        // 시간을 밤으로 만듭니다.
                        time = 1;
                        break;
                    default:
                        cout << "잘못된 입력입니다 ." << endl;
                        break;
                }
            }
        }
    }
}

```



```

else{
    //밤에 진행할 활동을 선택합니다.
    while(menu_select < 6){
        cout << "=====행 동 (밤 )===== " << endl;
        cout << "1. 메뉴 가격 수정 " << endl;
        cout << "2. 디저트 만들기 " << endl;
        cout << "3. 카페 메뉴 확인 " << endl;
        cout << "4. 정산 " << endl;
        cout << "5. 미니 게임 " << endl;
        cout << "6. 잠자기 " << endl;
        cin >> menu_select;

        switch(menu_select){
            case 1:
                break;
            case 2:
                menu.make_dessert();
                break;
            case 3:
                menu.show_menu();
                break;
            case 4:
                break;
            case 5:
                break;
            case 6:
                // 시간을 낮으로 만듭니다.
                time = 0;
                day++;
                break;
            default:
                cout << "잘 못 된 입력 입니다 ." << endl;
                break;
        }
    }
}
}
}

```

## (1) 기능 2 카페 메뉴 만들기

### 설명

카페 메뉴를 입력받아 제작합니다. 입력받은 데이터가 이전에 입력된 카페 메뉴의 이름과 동일하지 않다면 vector로 push하여 입력받습니다.

중복 여부는 menu\_check라는 bool변수를 통해 확인합니다.

후에 수요 공급 곡선에 맞추어 판매량을 정하여주는 함수를 추가할 예정입니다.

### 적용된 배운내용

#### 1) while문

while문을 활용하여 만일 메뉴의 이름이 겹치게 된다면 while문의 반복조건을 menu\_check라는 bool변수를 true로 만들어 다시금 반복문의 처음으로 돌아가 다시금 입력을 받도록 코드를 만들었습니다.

## 2)for문

for문을 활용하여 지금까지 존재한 모든 메뉴들과 입력받은 메뉴의 이름을 비교하도록 하였습니다.

## 3)if문

if문을 활용하여 만일 메뉴의 이름이 같다면 메시지를 출력하고 while문의 반복조건인 menu\_check를 true로 만들어 다시금 while문이 작동하도록 하였습니다.

while문의 처음에 menu\_check를 false로 만들어 for문이 끝났을 때 true로 바뀌지 않았다면 입력받은 값을 menu로 push합니다.

## 3) 클래스

플레이어 행동과 마찬가지로 menu와 관련된 내용을 Menu라는 클래스로 묶어놓았습니다.

## 4) vector

menu를 입력받아 중복되지 않은 메뉴라면 vector에 저장합니다.

## 코드 스크린샷

```
// 카페 메뉴와 관련된 클래스입니다.
class Menu{
    // 커피 메뉴에 관련된 내용입니다.
    vector<string> menu_name;
    vector<int> menu_price;
    vector<int> menu_sell;

    bool check_menu_make = false;

    string dessert_name;
    int dessert_price;
    int dessert_sell;

    bool check_dessert_make = false;

    int totalIncome;
public:
    Menu(){
        totalIncome = 0;
    }

    void make_menu();
    void make_dessert();
    void show_menu();
};
```

```

void Menu::make_menu(){
    string name;
    int price;

    // 해당 변수를 통해 메뉴가 정상적으로 입력되었는지 확인합니다.
    bool menu_check = true;

    // 제작할 메뉴를 입력받습니다.
    while(menu_check){
        menu_check = false;

        cout << "메뉴명을 입력해주세요 : " << endl;
        cin >> name;

        cout << "메뉴가격을 입력해주세요 : " << endl;
        cin >> price;

        for(int i = 0; i < menu_name.size(); i++){
            // 이미 존재하는 메뉴라면 menu_check를 true로 두어 다시 실행합니다.
            if(menu_name[i] == name){
                cout << "이미 존재하는 메뉴입니다." << endl;
                menu_check = true;
            }
        }

        if(!menu_check){
            menu_name.push_back(name);
            menu_price.push_back(price);

            menu_sell.push_back(0);
            check_menu_make = true;
        }
    }
}

```

## (1) 기능 3 디저트 만들기

### 설명

디저트는 커피의 판매량의 절반만큼 팔리는데 아직 커피의 판매량이 결정되지 않았으므로 현재 디저트의 이름과 가격을 입력받아 저장합니다.

### 적용된 배운내용

#### 1) 클래스

디저트도 메뉴에 포함되어 클래스에 속하여 있습니다.

#### 2) 입출력

cout를 통해 설명을 출력하고 cin을 통해 입력받아 디저트를 저장합니다.

### 코드 스크린샷

```

// 카페 메뉴와 관련된 클래스입니다.
class Menu{
    // 커피 메뉴에 관련된 내용입니다.
    vector<string> menu_name;
    vector<int> menu_price;
    vector<int> menu_sell;

    bool check_menu_make = false;

    string dessert_name;
    int dessert_price;
    int dessert_sell;

    bool check_dessert_make = false;

    int totalIncome;
public:
    Menu(){
        totalIncome = 0;
    }

    void make_menu();
    void make_dessert();
    void show_menu();
};

void Menu::make_dessert(){
    string name;
    int price;

    // 제작할 메뉴를 입력받습니다.
    cout << "디저트를 입력해 주세요 : " << endl;
    cin >> name;

    cout << "디저트 가격을 입력해 주세요 : " << endl;
    cin >> price;

    dessert_name = name;
    dessert_price = price;

    // 디저트 확인 시에 디저트의 여부를 판별
    check_dessert_make = true;
}

```

## (1) 기능 4 카페 메뉴 확인

### 설명

지금까지 제작된 카페의 메뉴를 모두 보여줍니다.

### 적용된 배운내용

#### 1) 클래스

메뉴를 보여주어야 하니 메뉴가 저장되어져 있는 Menu클래스에 속해 접근이 가능하도록 하였습니다.

#### 2)if문

커피와 디저트를 입력받을 때 커피는 check\_meun\_make, 디저트는 check\_dessert\_make 변수를 true로 두어 현재 메뉴가 존재하는 지 아닌지를 판별하여 출력하여 줍니다.

#### 3)for문

현재까지 vector에 입력받아져 있는 카페의 메뉴를 vector의 size만큼 돌려 하나씩 출력하여 보여줍니다.

### 코드 스크린샷

#### 클래스(위와 같음)

```
void Menu::show_menu(){
    if(check_menu_make){
        cout << "=====coffee)===== " << endl;

        for(int i = 0; i < menu_name.size(); i++){
            cout << (i+1) << "번 메뉴 " << endl;
            cout << menu_name[i] << endl;
            cout << menu_price[i] << endl;
            cout << endl;
        }
    }
    else{
        cout << "아직 만들어진 메뉴가 없습니다." << endl;
    }

    if(check_dessert_make){
        cout << "=====dessert)===== " << endl;
        cout << "오늘의 디저트 " << endl;
        cout << dessert_name << endl;
        cout << dessert_price << endl;
        cout << endl;
    }
    else{
        cout << "아직 만들어진 디저트가 없습니다." << endl;
    }
}
```

## (1) 기능 14 균형가격 설정

### 설명

주어진 가격에 따라서 정해진 함수를 통해 균형 가격을 설정합니다.

수요와 공급 곡선에서 가격이 줄어들면 수요는 늘어나지만 공급이 줄어들므로 판매량은 공급된 양만 팔려 공급 곡선에 의해 판매량이 결정되어지고, 가격이 늘어나면 공급은 늘어나지만 수요는 줄어들어 수요 곡선에 의해 판매량이 결정되어집니다.

후에 추가적으로 다른 추가된 메뉴에 한해서 추가적인 조정이 있을 예정입니다.

### 적용된 배운내용

1)for문

현재까지 vector에 입력받아져 있는 카페의 메뉴를 vector의 size만큼 가격을 각각 확인하여 정해진 내용에 따라서 판매량을 결정합니다.

### 코드 스크린샷

```
void Menu::sell_menu(){
    int price;
    for(int i = 0; i < menu_price.size(); i++){
        price = menu_price[i] / 100;
        int demand = 100 - price;
        int supply = 10 + 2 * price;

        int insert = min(demand, supply);

        menu_sell[i] = insert;
    }
}
```

## 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		----->			
기능 8		----->			
업무	11/17	11/24	12/1	12/8	12/15
기능 9			----->		
기능 10			----->		
기능 11			----->		

업무	11/24	12/1	12/8	12/15	12/22
기능 12			(일부)----->		
기능 13			----->		
기능 14			(일부)----->		
업무	11/24	12/1	12/8	12/15	12/22
기능 15				----->	
기능 16				----->	