

## 숙명 모두의마블 프로그램 만들기

### 1. 문제 정의

- 모두의마블 보드게임의 형태로 숙명인들이 즐길 수 있는 프로그램 만들기
  - 모두의마블은 정해진 보드판 상에서 주사위를 따라 이동하게 되고, 머무르는 칸에 맞는 동작을 적절하게 수행하는 게임임
  - 숙명 모두의마블 게임은 각 플레이어들이 자신의 에너지를 관리하면서 전공수업을 적절하게 듣고 졸업 학점 (credit)을 채우는 것을 최종 목표로 함
  - 모두의마블 특성 상 보드나 카드의 세부 구성을 자유롭게 구성할 수 있도록 코딩

- 보드 구성은 marbleBoardConfig.txt 입력 파일 내용에 따라 구성해야 함
  - marbleBoardConfig.txt 파일내 각 줄이 보드 한칸(노드)에 대한 설정을 가지며, 각줄은 아래 정보를 포함

(노드 이름) (노드 유형) (학점) (소요/보충 에너지)

- 노드 유형은 아래와 같음

번호	유형	설명
0	강의	전공과목을 들을 수 있는 노드
1	식당	에너지를 보충할 수 있는 노드
2	실험실	실험을 하는 노드
3	집	시작점이자 종료점 (지날때마다 에너지가 보충됨)
4	실험	실험을 하러 가는 노드 (실험실로 이동)
5	보충찬스	랜덤으로 음식카드를 골라 에너지를 보충할 수 있는 노드
6	축제	랜덤으로 미션을 수행하는 노드

- 노드의 개수는 가변이며 입력 파일의 내용에 따라 칸 수를 인식해야 함
- 첫번째 노드 (입력 파일의 첫줄)는 항상 유형이 집이며, 마지막 노드의 다음은 첫번째 노드로 연결되는 구조로 구성해야 함
- 학점은 강의 노드에서만 유의미하며, 강의 수강 후 플레이어가 얻게 되는 학점 수임
- 소요/보충 에너지는 강의 수강 혹은 실험실에서의 실험 시 소요되는 에너지이거나 식당 혹은 집에서 충전되는 에너지임

- 음식카드 구성은 marbleFoodConfig.txt 입력 파일 내용에 따라 구성해야 함
  - 파일 내 각 줄이 음식카드 하나에 대한 설정을 가지며, 각줄은 아래 정보를 포함

(음식 이름) (보충 에너지)

- 축제카드 구성은 marbleFestConfig.txt 입력 파일 내용에 따라 구성해야 함
  - 파일 내 각 줄이 축제카드 하나에 대한 내용을 담고 있으며, 띄어쓰기 없는 문자열로 구성됨
- 각 노드에 머무르면 플레이어는 다음을 수행해야 함
  - 강의 노드 : 현재 에너지가 소요에너지 이상 있고 이전에 듣지 않은 강의이면 수강 가능하며, 수강 혹은 드랍을 선택할 수 있음. (수강하면 성적이 A+, A0, A-, B+, B0, B-, C+, C0, C- 중 하나가 랜덤으로 나옴.)

- 식당 : 보충 에너지만큼 플레이어의 현재 에너지가 더해짐
  - 실험실 : 실험중 상태면 주사위를 굴려서 사전에 지정된 실험 성공 기준값 이상이 나오면 실험이 종료되고, 그렇지 않으면 이동하지 못하고 실험중 상태로 머무름. (단순히 실험실 노드에 머무른다고 실험중 상태가 되지 않으며, 실험 노드에 머물러야 실험중 상태가 됨)  
실험 시도마다 소요 에너지만큼 소요하며, 이로 인해 에너지가 음수가 될 수 있음
  - 집 : 지나가는 순간 지정된 보충 에너지만큼 현재 에너지에 더해짐
  - 실험 : 실험중 상태로 전환되면서 실험실로 이동 (주사위 눈 범위에서 실험 성공 기준값을 랜덤으로 지정.)
  - 보충 찬스 : 음식카드를 한장 랜덤으로 고르고 명시된 보충 에너지를 현재 에너지에 더함
  - 축제 : 축제카드를 한장 랜덤으로 골라서 명시된 미션을 수행
- 플레이어 설정 및 동작은 다음을 만족해야 함
- 플레이어 수는 게임 시작시 입력 받으며, 각 플레이어의 이름을 입력받아서 활용될 수 있어야 함
  - 시작 시 플레이어는 현재 에너지가 집 노드의 보충 에너지로 초기화되어야 함
  - 플레이어가 이동할 때 거치는 노드 이름을 모두 출력해야 함
  - 입력된 순으로 플레이어가 돌아가면서 주사위를 던지고 이동하며, 턴 시작 시 모든 플레이어의 상태 (위치, 실험중 상태, 누적 학점, 현재 에너지)를 출력해야 함
- 게임 종료 조건과 종료 시 동작은 다음과 같음
- 플레이어 중 한명이 GRADUATE\_CREDIT 이상의 학점을 이수하고 집으로 이동하면 게임이 즉시 종료됨
  - 게임 종료와 함께 졸업한 플레이어가 수강한 강의의 이름, 학점, 성적을 각각 출력
- 주어진 base code의 구조 하에서 구현하며, 각 소스코드파일은 아래와 같은 역할을 가짐
- `smm_object.c` : 게임에 필요한 객체(노드, 음식카드, 축제카드, 수강 이력) 정보 저장
  - `smm_database.c` : 보드의 노드, 음식카드, 축제카드, 각 플레이어의 수강 강의 이력을 저장하는 데이터베이스
  - `main.c` : 플레이어 관리 및 게임의 주요 동작 흐름 실현
- 노드, 음식카드, 축제카드, 수강 이력 등은 한 형태의 구조체를 활용해서 표현
- Data encapsulation 형태로 구현
- 소스코드파일 간 전역 변수 공유 사용 금지 (즉, `extern` 문법 사용 금지)
  - 구조체 정의는 `smm_object.c`에서 이루어지며, 이 파일 내에서만 사용 가능
  - linked list는 `smm_database.c` 내에서만 접근 가능하며 다른 파일에서는 `smm_database.h` 내 `prototyping`된 함수를 통해서만 접근 가능
  - 표준 입출력은 `main.c`에서만 사용 (즉, 키보드 키보드 및 모니터 출력은 `main.c`에서만 다루어야 함)
  - `main.c`에서만 `smm_database.h` 및 `smm_object.h`를 `include`할 수 있음  
(`smm_database.c`에서 `smm_object.h`를 `include`하거나 `smm_object.c`에서 `smm_database.h`를 `include`하면 안 됨)

## 2. 코딩 방향

- Base code로 주어진 `main.c` 및 `smm_object.c` 파일을 채우는 형태로 완성
  - `smm_object.c` 내에 각종 객체 정보를 담는 구조체를 정의하고 객체를 다루는 각종 함수 구현
  - `main.c` 내에 파일 입출력을 통해 객체 구조체를 생성하고 linked list에 저장하는 코드 구현 (`smm_database.h` 함수 활용)
  - `main.c` 내에 플레이어를 설정하는 코드 구현
  - `main.c` 내에 게임 흐름에 맞는 코드 구현
- `main.c` 내에 게임 흐름과 관련된 모듈 함수에 대한 `prototyping`을 힌트로 제공하였으며, 필요시 사용 가능
- Base code로 주어진 `smm_database.c`는 있는 그대로 활용하며, 제공된 함수를 header에서 참고하여 활용
- 함수/변수에 대한 이름을 잘 정의하고, 각종 정의 및 코드 흐름에 대해 다른 사람들이 알아볼 수 있도록 주석 삽입

- 코드를 보기 쉽게 들여쓰기를 적절히 삽입
- 기능 혹은 일부 코드를 구현하거나 디버깅 과정에서 코드를 수정할 때마다 Github에 올려서 변화를 추적할 수 있도록 함
- 코딩에 필요한 요소들은 추후 관련 이론을 배우는 주치의 실습 시간에 다룰 예정임

---

### 3. 채점 기준

- 프로그램 동작 점수 : 120점
  - 문제 정의에 명시된 기능들이 충실하게 잘 동작하는 지 여부
  - 정상적으로 동작하면 만점, 부분적으로 동작하면 구현 코드 및 표면적인 동작 상태에 따라 부분점수 부여
  - 배점 : 보드/카드 구성 - 20점, 이동 - 20점, 노드 동작 60점, 성적 관리 - 10점, 게임 종료 동작 - 10점
- 코드 관리 점수 : 50점
  - Data encapsulation 개념으로 잘 구현되어있는가
  - 변수 및 함수 이름을 잘 지정했는가
  - 주석을 충실하게 달았는가 (함수 및 변수 정의에 대한 설명 및 코드 흐름에 대한 설명)
  - 들여쓰기를 잘했는가
  - Github에 코드를 단계적으로 구현한 이력이 남아있는가

**- copy 여부에 대해 집중적으로 볼 예정이며, 적발 시 관련자 모두 0점으로 처리**

---

### 4. 결과물 제출 방법

- Github 계정에 별도의 repository로 올림
  - SMMarble라는 이름으로 repository 생성
  - 제출 기한 시간 직전에 올라간 소스코드 파일들을 기준으로 채점
  - Github에 SMMarble라는 repository가 없으면 제출하지 않은 것으로 간주할 예정이므로, 제출 후 웹 페이지 상에서 해당 repository가 보이는지 반드시 확인 요망
  - 가급적 미리 올리거나 지속적으로 여러번 submit을 하는 형태로 진행 요망

**- 제출 기한 : 12월 21일 목요일 23:59까지**