

소프트웨어입문설계 과제#4

~ 축제기간 특집: 간단함을 위한 복잡함 ~

도입부

어느덧 우리 수업도 마지막 한 달을 남겨 두고 있고, 이전에 배웠던 다양한 개념들, 요소들을 모으면 그럭저럭 복잡한 프로그램을 만드는 것도 가능할 정도가 되었습니다. 그러니, 이제는 더 이상 '옛날보다 더 어려워보이는 거 만들 거야' 하는 마음을 갖지 않아도 그럴싸 한 프로그래밍 연습을 얼마든지 할 수 있습니다.

축제기간을 맞이하여, 이번 과제는 이전보다 조금 '덜 어려워보이는' 목표를 여러분에게 제시합니다. 심지어 실습시간에 본 재미지옥 탈출 프로그램만큼 직관적인 규칙을 가지고 있어요. 뭐, 이미 된통 속아서 '내 눈에 쉽다고 내 손에도 쉬운 건 아니다'는 것을 알아버린 만큼, 이번에는 조금 더 여유를 두고 '사실은 복잡한' 이 프로그램을 직접 만들어 보도록 합시다.

과제 수행 방법

1. 먼저 HY-in 과제 페이지에서 hw4_base.zip을 다운로드해서 압축을 풀어 주세요.
2. 압축 푼 폴더 안에 있는 hw4_base.sln 파일을 더블클릭해서 이 '솔루션'을 VS로 열어 주세요.
(주의: '압축 푼 폴더'인지, .sln 파일을 더블클릭했는지 한 번 더 확인해 주세요)
3. 일단 Ctrl + F5를 눌러 실행해 보세요. 이번 과제는 축제기간 특집인 만큼 '일단 바로 강사 버전을 실행할 수 있도록' 코드를 담아 놓았어요. 몇 번 실행해 보면 뭐 하는 프로그램인지 바로 알 수 있을 거예요.
(주의: main_yeshi.c나 main.c를 먼저 열어보면 마음에 상처를 받을 수 있어요!)
4. 이 프로그램은 여러 '상태(state, 또는 뭐 태세stance라 불러도 무방할 듯)'들을 가지고, 각 상태들 사이를 전환해 가며 동작하고 있어요. 다음 페이지에 있는 그림을 보면서, 이 프로그램의 실행이 실제로 이 그림대로 되고 있다는 사실을 납득해 보세요:

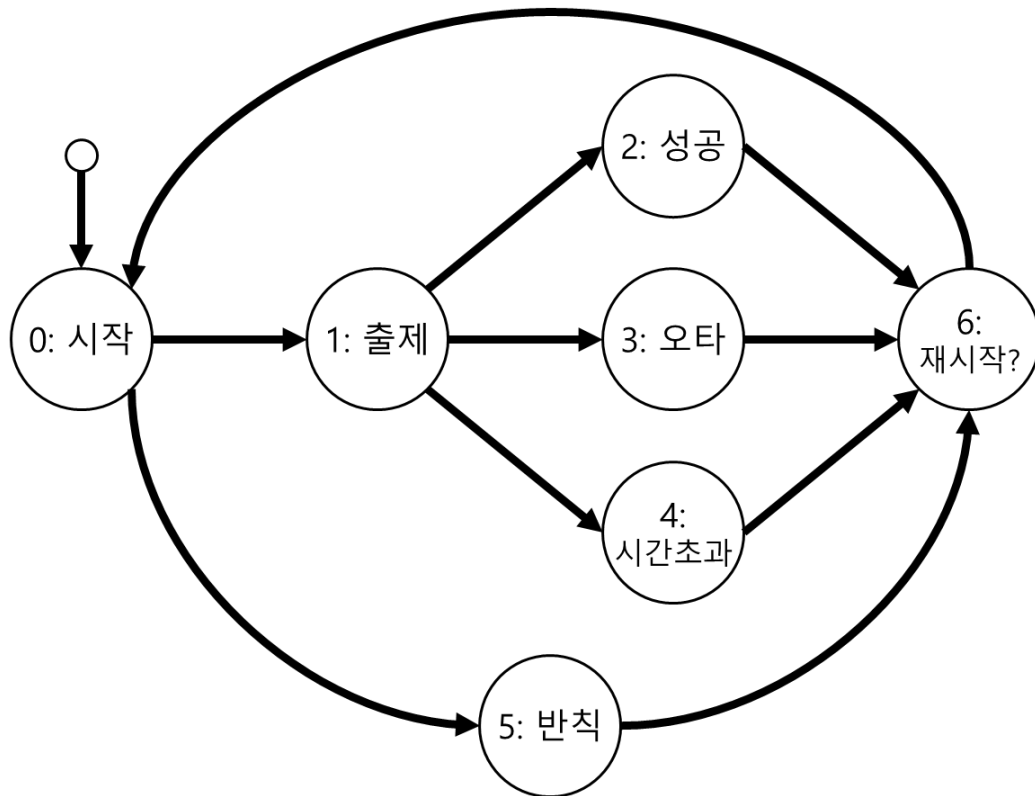


그림 1. 이 프로그램에 대한 상태 전환 그림(보통 이런 그림을 state diagram이라 불러요)

5. 위의 그림을 보며 다음 내용들을 직접 상상해 보세요(몇 번 더 실행해 보면 하기 쉬울 듯):

- 프로그램을 처음 켜면 시작 상태로 전환해요.
- 시작 상태가 되면 다음 문제를 사용자 몰래 준비해요.
몰래 정한 시간(예시에서는 0.5 ~ 5.5초)이 다 되면 출제 상태로 전환해요.
키 누름 입력을 받으면 반칙 상태로 전환해요.
- 출제 상태가 되면 화면에 문제를 print해요.
정해진 시간(예시에서는 0.5초)이 다 되면 시간초과 상태로 전환해요.
키 누름 입력을 받았고, 그 키가 정답이 맞으면 성공 상태로 전환해요.
키 누름 입력을 받았고, 그 키가 정답이 아니면 오타 상태로 전환해요.
- 성공 상태가 되면 정해진 승리 메시지와 '얼마만에 눌렀었는지'를 print해요.
오타 상태가 되면 정해진 패배 메시지와 '뭐 눌러야 하는데 뭐 눌렀었는지'를 print해요.
시간초과, 반칙 상태가 되면 각각 정해진 패배 메시지를 print해요.
정해진 시간(예시에서는 5초)이 다 되면 재시작? 상태로 전환해요.
- 재시작? 상태가 되면 정해진 재시작 방법 메시지를 print해요.
엔터 키 누름 입력을 받으면 시작 상태로 전환해요.
- ...이 흐름을 사용자가 뺄종할 때까지 계속 진행해요.

6. 확인해 보니, 이 그림대로 코드를 작성한다면 대충 각 동그라미마다 아래와 같은 Code들을 정하면 되는 것 같아요:
 - 해당 상태에 처음 진입했을 때 해야 할 일 정하기
 - 해당 상태일 때 키를 누르면 어떻게 처리할 지 정하기
 - 해당 상태일 때 시간이 경과되면 어떻게 처리할 지 정하기
7. 5번, 6번 항목에 적어 둔 내용을 어느 정도 이해했으면, 이제 main_yeshi.c를 열고 Ctrl + A를 눌러 전체 코드를 선택한 다음 Ctrl + E, C를 눌러 해당 내용을 주석으로 만들어 주세요.
(안 되는 친구들은 전체선택 후 편집(E)→고급(V)→선택 영역을 주석으로 처리(M)를 눌러요)
8. 그 다음, main.c를 열고 전체 선택 후 Ctrl + E, U를 눌러 해당 내용의 주석을 풀어 주세요.
(안 되는 친구들은 7번 항목에서 본 메뉴 근처에 있는 제거 메뉴 누르면 돼요)
9. 일단 전체 코드를 구경해 봅시다. 코드 맨 위에는 각 키에 해당하는 코드 값이 #define으로 마련되어 있고, 그 아래에는 '아마도 여러분이 반드시 쓰게 될 변수' 선언을 미리 적어 두었어요. '안 봐도 되는 부분'은 지금은 안 봐도 좋고, main() 함수 정의 아래에 보이는 각 상태별 함수 정의들을 잘 확인해 보세요.
 - 6번 항목에서 착안한 세 가지 작업들을 각각의 **함수 정의**로 만들어 두었어요.
10. 이제 과제 설명서의 그림 및 내용들, 코드에 적어 둔 각 함수별 설명들을 참고해 가며 총 18개의 함수 정의를 조금씩 완성해 보세요.
 - 아마도 변수들을 몇 개 더 선언해야 할 거예요.
파일 맨 위의 '선언하기 적당한 자리'에 주석으로 표시해 두었으니 거기다 ^^하면 됨!
 - 선언 끝내고 다시 내려올 때는 미니맵 스크롤 바를 활용해 봐요.
 - 각 함수별 내용을 어떻게 적을 지 애매하면 main_yeshi.c를 참고하면 될 듯 해요.
 - return값은 '이제 다음에 어떤 상태로 전환해야 하나'를 의미하고, 그림에 적어 둔 숫자를 보며 그대로 활용하면 돼요. 원래 상태 그대로 남아 있어야 한다면 현재 상태 번호를 그냥 return하면 돼요.
11. 전체 프로그램을 꽤 많은 함수들의 집합으로 나누어 표현했기 때문에, 각 함수들 하나하나가 해야 할 일들은 그리 납득하기 어렵지 않았을 거예요.
12. 모든 함수 내용을 다 적었으면 Ctrl + F5를 눌러서 실행해 봐요. 예시 버전에서 봤던 대로, 그림을 보며 상상한 대로 프로그램이 실행되면 과제 성공이에요!

과제 제출 방법

1. 여러분이 작성한 main.c를 HY-in 과제 페이지에 제출해 주세요:
 - 코드 창 위에 보이는 main.c 탭을 우클릭하고 '상위 폴더 열기'를 누르면 찾을 수 있어요.
2. 제출할 때 느낀점도 함께 적어 주세요!
 - 보통 실습 마무리 단계에 썼던 그 칸에다 적으면 돼요.
3. 과제 제출 기한은 6월 1일 토요일(축제 다음 주일) 23:59:59까지입니다!

과제 평가 규칙

이번 과제의 경우 별도의 평가를 진행하지 않아요. 따라서 여러분이 .c 파일을 기한 안에 제출하면 모두 만점을 받을 수 있어요. 자신의 의도에 따라 느긋하거나 빠르게 과제에 참여할 수 있도록 자유도를 부여해 두었으니, 대학 첫 축제를 의미 있게 보내는 것도 나쁘지 않을 듯 해요!

'어... 저는 뭔가 더 해 보고 싶은데요' 한다면

물론 몇몇 친구들은 여기서 조금 더 많은 경험을 얻고 싶을 거예요. 만약 그렇다면 아래의 내용을 추가로 확인해 보면 좋을 듯 해요:

- 현재 코드 구성은 여러분이 '개발자용 치트 입력'을 얼마든지 탑재할 수 있도록 되어 있어요. Process_Input() 계열 함수들을 약간 고치면(아마 추가 변수 필요) 될 듯!
- 의외로, 이 과제에서 구사하는 main() 구성은 상당히 다양한 분야에서 사용되는 보편적인 '복잡한 프로그램 간단히 만들기' 기법이에요. main()의 각 부분들에 달린 주석들을 참고해 보면서 약간 눈에 익어 두면 좋을 거예요.
- main() 해독에 성공했다면, 여러분은 이제 지금 것과 유사한 state diagram들을 코드로 옮기기 위한 사전 지식을 어느 정도 확보한 게 돼요(여기서의 '유사한'을 판단하려면 조금 더 경험을 쌓아야 하긴 해요). 그러니 일단은, 지난 실습에서 본 개미지옥 탈출 프로그램에 대한 state diagram을 그리고, 각 함수의 내용을 '개미지옥 프로그램에 적합하게' 바꾼 다음 직접 실행해 보면 좋을 듯 해요.
 - 여러분의 새 state diagram에 있는 동그라미 수가 6개보다 적다 해도 안 쓰는 함수 정의를 지울 필요는 없어요(호출 안 할 거니 노상관). 필요한 것만 고쳐서 쓰면 돼요.
 - 6개보다 더 많은 동그라미를 그린 경우에는 코드 파일의 '안 봐도 되는 부분'을 고쳐야 해요. 이 부분을 손대고 싶다면 강사의 도움을 요청해 주세요.