



이전과 비교하여 달라진 사항을 중점으로 작성하였다.

6 개의 버튼 UI 를 조작하기 위해 btn_6s 라는 배열과 그 배열을 조작하는 함수를 새로 하나 만들었다. 또 2 개의 버튼 UI 를 조작하기 위해 btn_2s 라는 배열과 그것을 조작하는 함수도 새로 하나 만들었다. 버튼 UI 가 추가되고 새로운 화면에 추가됨에 따라서 함수의 버튼 하이라이팅을 담당하는 함수와 일종의 메인함수라고 할 수 있는 부분에 구현 되어있는 방향키 조작을 받아들이는 조건문, 그리고 엔터키 조작을 실행하는 조건문에 추가된 사항을 반영했다. 새로운 화면 같은 경우, 저번 보고서에도 언급했듯 while 문으로 echo 를 몇 번이고 호출해서 화면을 구성했다. 그래서 그런지 저번 보고서 작성 당시에도 화면이 상당히 느리게 떴지만, 새로 추가된 화면은 더더욱 느려졌다. 맵을 그리는 것은 사람 손으로 64 개 배열을 일일이 치기에는 많으니 이런 방식인 것이 어쩔 수 없는 면이 있지만, 화면 구성은 많아봐야 echo 문 4 줄이나 5 줄이면 충분하다. 화면이 뜨는 속도를 보면서, 어떠한 방식을 채택하기 전에는 항상 더 나은 다른 방식은 없는지 생각하는 것이 필요하다는 것을 느낀 부분이다.

처음 실행할 때 메인화면에 1P LOGIN 과 2P LOGIN 을 띄우는 것은 변수 출력에 :=를 사용해서 구현하였다. 그리고 로그인 화면의 디폴트는 달라지지 않도록 구현하는 것이 요구 사항이었으므로 새로운 ID/PW 변수를 만들어서 요구 사항을 만족시켰다. 다만 이런 방식 때문에 ID/PW 변수가 굉장히 많아졌는데, 회원가입과 회원탈퇴에 쓰이는 ID/PW, 1p 로그인만을 위한 ID/PW , 2p 로그인만을 위한 ID, PW, 연제가 구현할 게임 전적을 덮어쓰기 편하기 위해 만든 1p 의 현재 ID/PW , 2p 의 현재 ID/PW 로 종류로만 묶어도 5 가지 종류나 된다.

맵은 64 개의 원소를 가지는 1 차원 배열을 만들어 각 격자 안의 BOX 색깔을 조작했다. 2 차원 배열로 하려고 하였으나, 하술할 이유로 1 차원 배열로 제어하였다. 첫 번째로, 원소의 수 자체가 그렇게 많지 않다. 두 번째로, bash 셸에서도 2 차원 배열을 정식으로 지원하지 않아 사용이 번거롭다. 마지막으로, 이동을 구현하는 데 있어서는 1 차원 배열의 index 로 생각하는 게 원소의 수가 많지 않아 예기치 못한 오류 발생을 막는 데 더 도움이 된다고 판단했기 때문이다. 맵 내부에서의 이동은 맵의 인덱스 값을 방향키로 결정하고, 해당하는 인덱스의 값을 흰색 글자색깔 코드로 치환한 다음, 바뀐 맵의 글자색깔 코드 값을 반영하여 출력하는 방식으로 구현했다. 다만 이렇게만 구현하면 파란 영역이나 노란 영역으로 이동했을 때 그 영역의 색깔 값을 흰색으로 덮어써버린다는 것이 문제였다. 그래서 파란 영역과 노란 영역만을 저장하는 일종의 사본 배열을 만들어서 매번 출력할 때마다 원본 맵 배열이 사본 배열의 값을 받아오고, 그 후에 방향키 조작에 따라 바뀐 인덱스 값에 따라 흰색 색깔 코드를 씌우는 방식으로 파란 영역이나 노란 영역으로 이동하더라도 덮어씌워지지 않도록 했고, enter 키로 돌을 놓을 때는 무조건 사본 배열의 값으로만 판단하고, 돌을 놓은 곳의 인덱스는 사본 배열에서의 해당 인덱스 값만을 바꾸도록 해서 돌을 중복해서 놓으려고 하더라도 카운터가 못 올라가도록 막았다.

다만 맵의 이동을 열추 요구사항과 맞춘 것과는 별개로 맵의 디자인은 요구 사항과 다소 다르다. 요구 사항에서는 흰색 테두리 격자 내부에 BOX 가 있었지만, 본 프로그램에서는 격자 밑변 중간에 검은색 BOX 가 끼어들어서 완벽한 흰색 테두리 격자가 아니다. 요구 사항에 맞게 하려고 노력하였으나, 전혀 감이 오지 않아 비슷한 느낌으로 한 결과, 본 프로그램과 같이 디자인되었다. 맵을 그린 방식은 강의 시간에 들은 팁을 참고했다. 전체 맵을 상단, 중단, 하단 부분으로 나누었다. 상단만 그리는 함수, 중단만 그리는 함수, 하단만 그리는 함수를 구현했고, 그 3 개 함수를 통해 하나의 맵을 그리는 draw 함수를 구현했다. Draw 함수는 상단 함수 1 번, 중단 함수 7 번, 하단 함수 1 번을 호출하도록 구성되었다. 다만 특이하게 상단, 중단, 하단 함수에 엔터문자가 빠져있는데, 이는 맵 선택 화면에서 맵을 그릴 때 이 함수를 활용하기 위해 일부러 엔터 문자를 뺐다. Draw 함수로는 맵 선택 화면처럼 맵을 가로로 배치할 수 없었기 때문이다. 그래서 Draw 함수에서도 보면 엔터문자를 직접 함수 호출 끝부분마다 넣어주고 있다.

이하의 고찰 부분이다.

고생을 많이 한 부분은 맵을 어떻게 그리고 맵의 이동을 어떻게 자연스럽게 구현하는 부분이었다. 하지만 가장 당황스러운 부분은 처음 화면의 구성을 바꾸는 부분이었다. 버튼을 2 개 추가하고 3 개 버튼의 역할을 수정하는 일이라서 겉보기에는 큰 변화가 아니라서 큰 어려움은 없을 줄 알았지만 요구 사항에 맞추려면 생각보다 수정해야하는 부분이 많아서 당황스러웠던 것이다. 이 부분을 해보며 개발자의 고충을 조금 느껴볼 수 있었다. 새로 만드는 게 힘든 일인 것은 당연하지만, 만들어 놓은 것을 조금씩 바꾸는 것도 매우 힘든 일이라는 사실을 깨달은 것이다.

저번 보고서 이후로 오랜만에 프로그램 코드를 열어보니 어떤 식으로 작성했는지 이해하는 시간이 좀 필요했다. 물론 필자가 스스로 짠 프로그램인 만큼 오래 걸리지는 않았지만, 만약 다른 사람과 협업하는 과정이었다면 필자의 프로그램을 이해할 수 있을지 의문이 들었다. 주석도 거의 달지 않았으니 뭐하는 코드인가 싶었을 것이라고 생각한다. 이런 생각을 통해 코드를 짤 때 생각나는대로 코드를 짜면서 그것에 맞춰 역지로 돌아가도록 코드를 추가하는 방식도 시간이 부족하다면 어쩔 수 없지만, 되도록이면 팔짱 끼고 가만히 생각해보고 펜대도 굴려보며 시간을 가지고 코드를 작성하는 시간이 필요하지 않을까 생각해보게 되었다.