프로젝트2 보고서

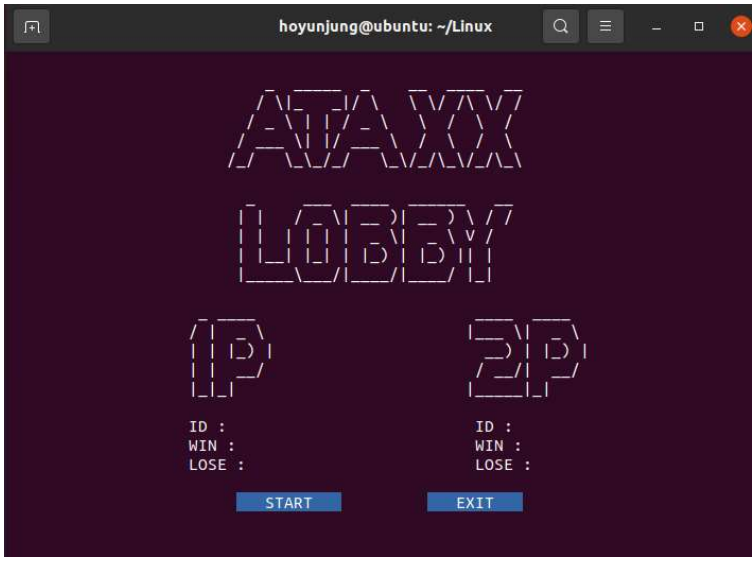
2018203091 정호윤

Step1. '시작화면' 재구성, 로그인 기능 변경

시작 화면을 재구성하였습니다.

1P, 2P 로그인은 기존 JOIN을 통해 이동한 로그인 함수를 각각의 로그인 버튼을 통해 이동하는 방식으로 변경하였으며, 로그인 성공시 입력한 ID의 값을 변수에 저장해 화면에 띄우도록 했습니다.

Step2. '게임 대기 화면' - UI 구현



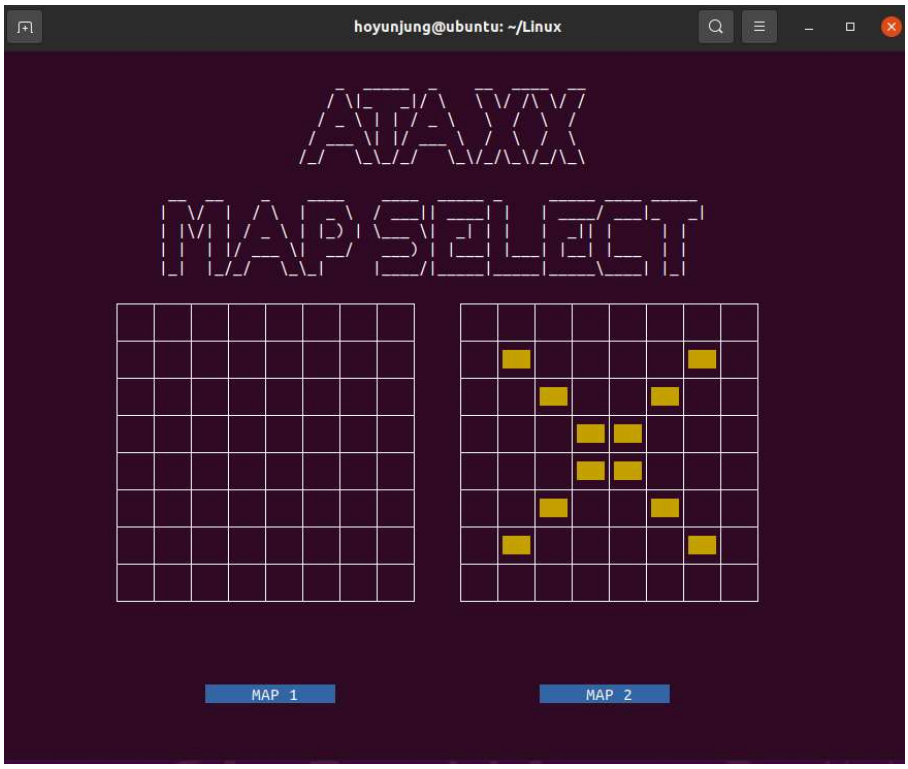
대기 화면 UI입니다.

시작 화면에서 JOIN을 입력하면 lobby_func으로 넘어가 다음과 같은 화면을 출력하게 됩니다.

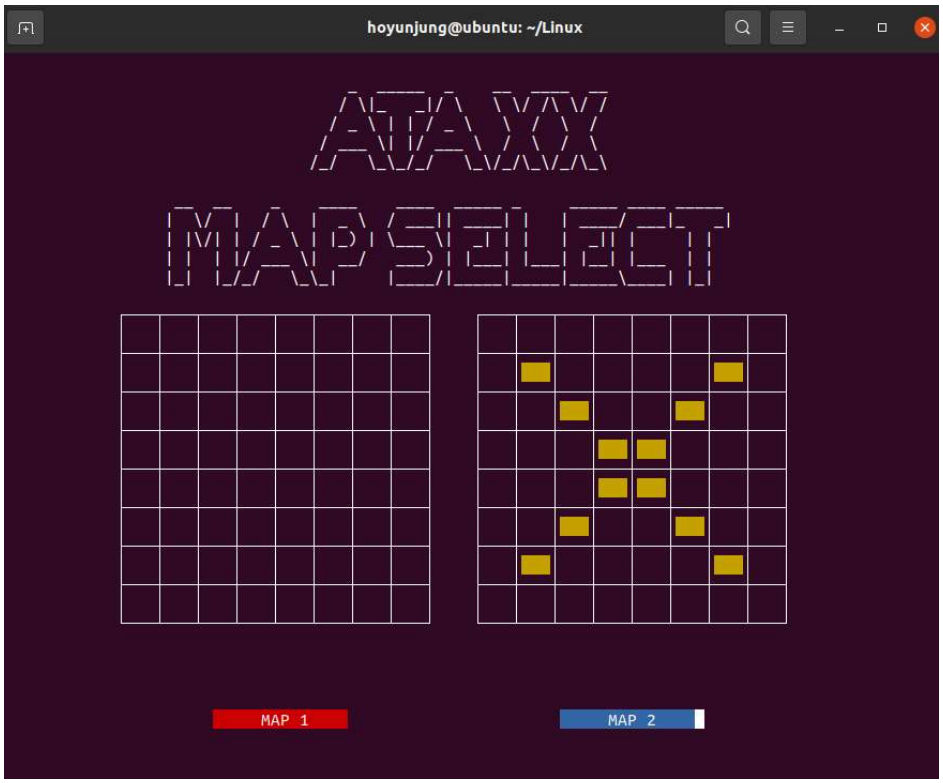
Step3. '게임 대기 화면'- 기능 구현

대기 화면 기능 구현입니다. 로그인 성공시 저장한 ID의 값을 이용해 userdata.txt 파일에서 grep 명령어로 해당 행의 내용을 파악, cut을 사용해 잘라낸 값을 각각 win과 lose의 값에 저장합니다. 그리고 저장한 값을 게임 대기 화면에서 출력할 수 있도록 설정했습니다.

Step4. '맵 선택 화면'- UI 구현 및 하이라이팅 구현

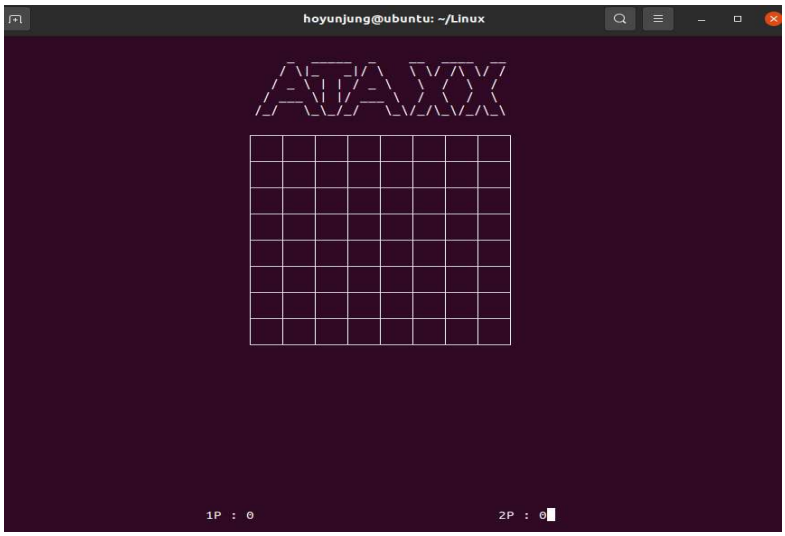


UI 구현입니다. Map은 Map 함수를 만들어 구현했으며, 특수문자를 활용해 다음과 같이 구현했습니다. Map 1과 Map 2는 같은 함수에서 다뤄지며, map 2를 나타내는 index 값이 입력될때만 if문을 사용해 노란색으로 추가 출력이 될 수 있도록 조정했습니다.



하이라이팅 역시 기존과 같은 방법으로 구현했습니다.

Step5. '맵 1'- UI 구현

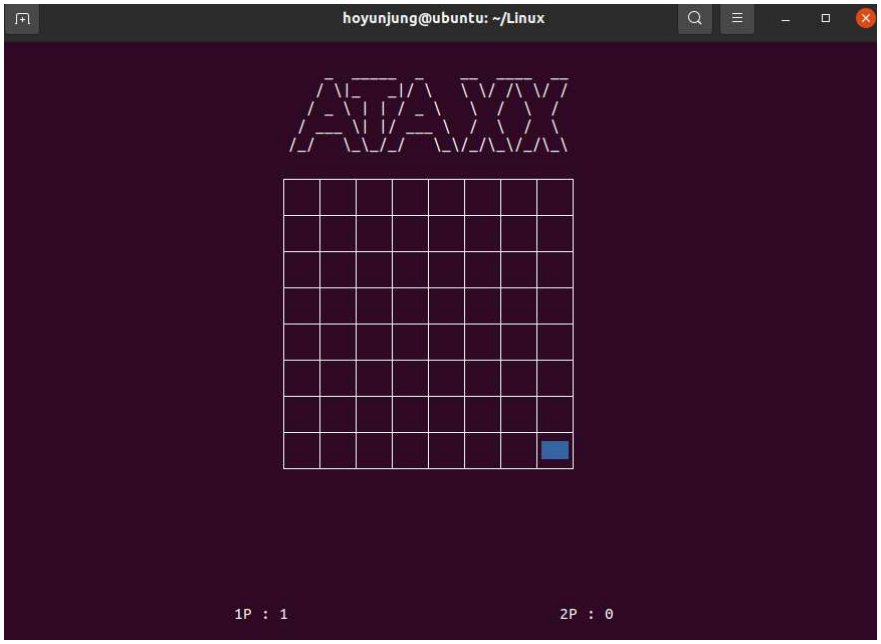


맵 UI 구현입니다.
맵을 출력하는 방식은 Step4의 함수와 같은 함수를 사용했습니다.

Step6. '맵 1'- 방향키 이동 구현

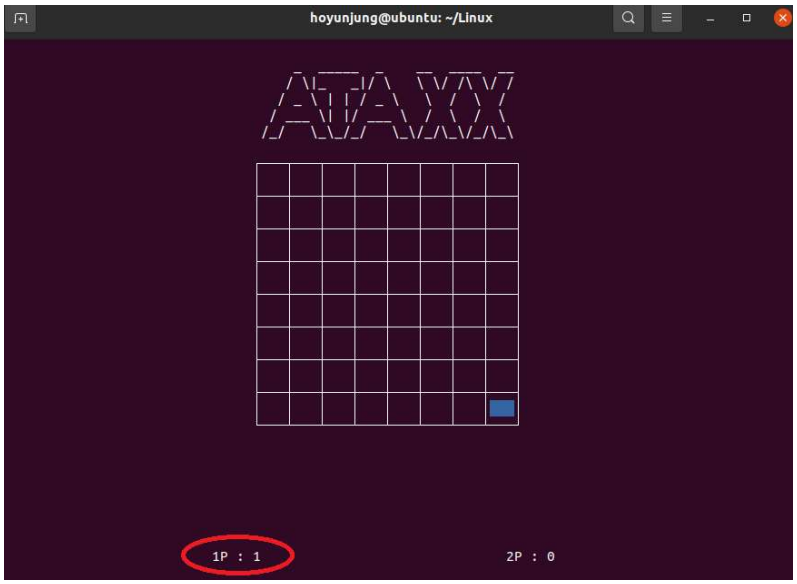
방향키 이동입니다. 방향키가 입력될 때마다, index의 값이 변경되어 현재 선택된 index에만 흰색으로 출력되도록 설정했습니다.

Step7. '맵 1'- 엔터 -> 칸 색깔 변경 구현



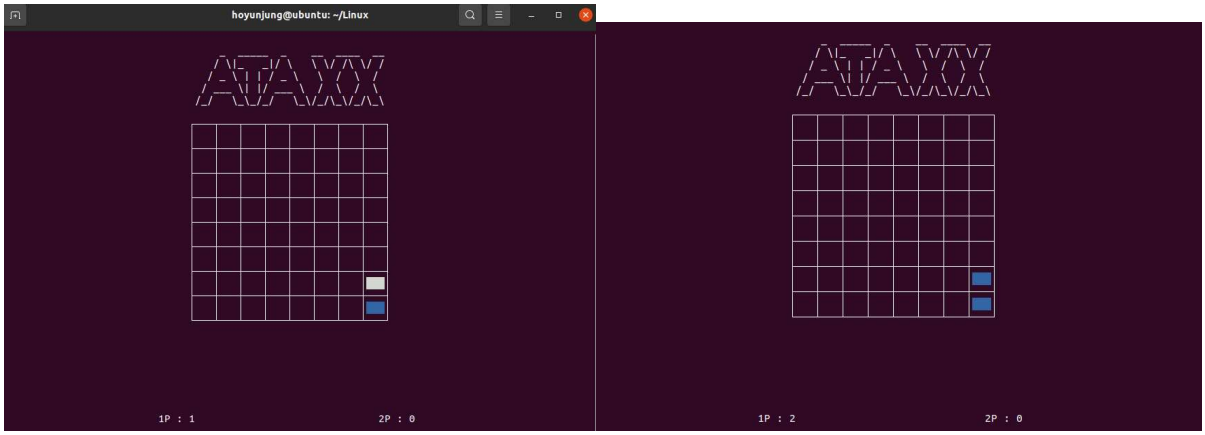
칸 색깔 변경입니다. 칸은 주어진 8 X 8의 맵의 칸을 64개의 배열을 통해 각각의 index 값으로 저장되어 있습니다. 이 값은 초기에 0으로 설정되어 있고, 엔터를 입력시, index의 값이 0이라면 이 값을 1로 변경하게 됩니다. 그리고 배열의 index값을 조사해 index가 1인 경우에만 파란색 칸으로 출력되도록 설정했습니다.

Step8. '맵 1'- 엔터 -> 1P 칸 수 증가 구현



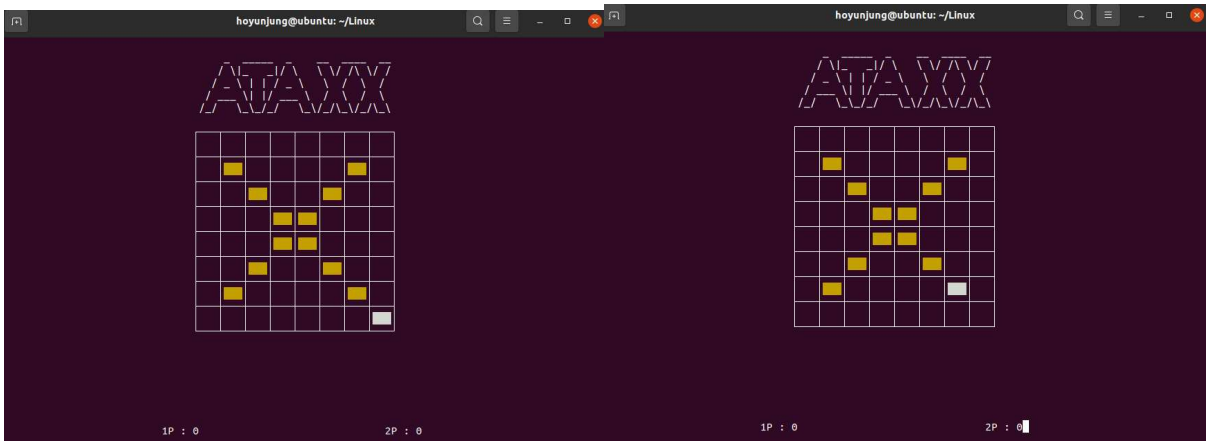
칸 수 증가입니다. 칸 수는 현재 index가 1인 항목이 몇 개인지 카운트합니다. 배열의 index의 1 값의 개수로만 판단하기 때문에 중복으로 증가하지 않습니다.

Step9. '맵 1'- 칸 색깔 변경 후 커서 이동 구현

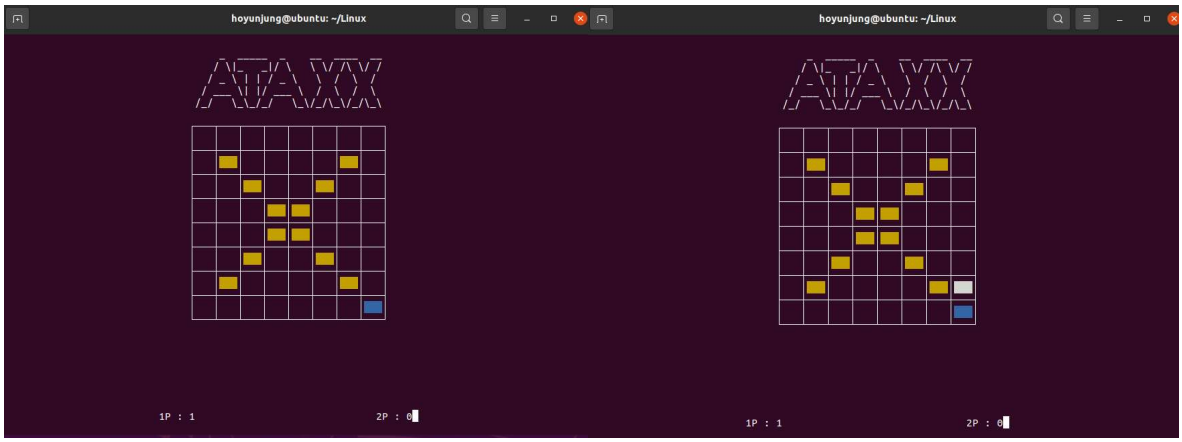


칸 색을 변경 후 이동 또한 가능합니다. 이동 방식은 동일하며, 커서의 위치에만 흰색으로 표시됩니다. 엔터를 통해 돌을 놓을 시, 정상적으로 카운트 되며 색상도 파란색으로 변경됩니다.

Step10. '맵 2'- UI 구현 및 5~9번 기능 삽입



맵 2 역시 맵 1과 동일하게 커서 및 이동이 정상적으로 작동합니다.
대신, 맵 2에선 노란색 부분이 index의 값이 2로 설정되어 있기 때문에, 0을 1로 변경하는 선택 및 1의 개수를 세는 카운팅이 불가능합니다.



이외의 다른 칸 선택 및 커서 이동은 맵1과 동일하게 가능합니다.

고찰

1. 맵을 출력할 방식에 대해 고민을 많이 했습니다. 텍스트의 박스를 구현하는 시간만큼은 아니었지만, 맵을 구현하는 방식을 떠올리는 과정이 과제 구현 과정의 1/3을 차지했습니다.
2. 맵의 구현은 특수문자를 활용해 구현을 성공했습니다. 그렇지만 다른 방법으로 구현할 방법은 없는지에 대한 의문이 생겼습니다.
3. 현재 구현한 방식 역시 userdata의 접근이 쉽게 가능하기 때문에 보안성을 강화할 필요가 있지 않을까라고 느꼈습니다.