자료구조 실습

개인 프로젝트

보고서

2019203021

이승헌

프로젝트명 : 인디언 홀덤

프로젝트에 사용된 기술

1.Signal

-사용된 곳

i)SIGINT, SIGTERM 시그널에 시그널 핸들러 걸기

ii)SIGWINCH 시그널에 시그널 핸들러 걸기

-사용한 이유

i)의 경우, 이렇게 종료하면 게임의 기록이 저장되지 않습니다. 게임의 기록을 저장하고 종료시키기 위해 시그널 핸들러를 걸었습니다.

ii)의 경우, ncurses의 특성상 화면이 갱신될 때마다 refresh하는 동작이 필요합니다. 그런데, 화면의 크기를 늘렸다 줄이는 것에 대해서는 refresh하지 않아서 화면을 크기를 늘렸다 줄인 후 화면의 크기를 돌려놓으면 정상적인 화면을 출력하지 않습니다. 그래서 SIGWINCH 시그널을 이용해 화면이 바뀌어도 계속 refresh를 하여 정상적인 화면을 출력하도록 했습니다.

2.파일 입출력

-사용된 곳

파일의 읽기, 쓰기

-사용한 이유

게임 판 수, 게임 승/패, 게임의 승률을 저장하기 위해 사용되었습니다. 명색이 게임인만큼, 이런 요소가 필요하다고 생각했습니다.

3.연결 리스트

-사용된 곳

덱 만들기

-사용한 이유

덱에서 카드를 자주 뽑아야 하기 때문에, 제거가 다소 까다로운 배열보다는 연결 리스트로 만드는 것이 좋다고 판단하여 연결 리스트를 이용해 덱 만들기를 구현하였습니다.

프로젝트에 대한 의견

장점 :

-인디언 홀덤을 컴퓨터로 안전하게 해볼 수 있습니다.

깃허브에 그대로 인디언 홀덤이라고 검색해도 2개밖에 없고, 게다가 깃허브가 아닌 검색 엔진에 쳐보면 안전하게 즐길 수 있는 방법은 마땅히 없어서 같이 인디언 홀덤을 한 번 해보고 싶다고 하면 한 번 정도는 괜찮은 선택지가 되지 않을까 생각합니다.

-게임 치고는 가볍습니다.

실행파일은 300KB도 되지 않아 매우 가볍습니다.

단점 :

-UI/UX 양쪽이 둘 다 매우 후진적이라는 단점이 있습니다.

UI는 1인이 개발하는 것이기에 여유가 없었다고 하더라도, UX의 경우는 아쉬운 부분이 있었습니다. 첫 번째로, 플레이어 측은 베팅을 마음대로 할 수 있도록 구현하려고 했으나, 오류가 나서 게임이 갑자기 꺼지는 확률이 매우 높았습니다. 그래서 결국 플레이어도 컴퓨터와 같이 무작위 숫자로 베팅하도록 구현했습니다. 그 결과 베팅 과정에 극적인 것이 없이 콜한 칩에서 하나에서 두 개정도 많은 칩만 거는 모습을 볼 수 있습니다. 해결 방법만 잘 찾을 수 있었다면 이 부분은 입력을 받을 수 있도록 했을 것입니다.

두 번째로, 게임의 모든 것이 전부 영어로 적혀있습니다. 한글을 억지로 집어넣을 수는 있지만, 개발 환경의 문제인지 제대로 출력되지 않았습니다. 물론 한국인 개발자가 집어넣은 영어인만큼 구글 번역기를 사용하면 개발자가 무슨 말을 하고 싶었는지 그대로 나옵니다. 하지만 한국인을 대상으로 만든 프로그램인만큼 한글을 썼다면 더 좋은 경험을 줄 수 있을거란 아쉬움이 남습니다.

그 외에도 UX에서는 아쉬운 점이 많이 남습니다.

-컴퓨터의 행동 방식 구현이 조금 아쉽습니다.

컴퓨터의 판단 기준은 자신의 족보와 상대의 족보입니다. 사실상 사기를 치는 상대로 게임을 하는 것이기에 조금 아쉬운 감이 있습니다.

가능하면 카드 카운팅을 하는 식으로 판단 기준을 다듬었겠지만, 다루어야 하는 경우의 수가 너무 많아져 결국 이런 방식을 채택했습니다.

-버그가 있습니다.

플레이어의 행동을 선택하는 부분에서 해당하는 행동에 커서를 두고 마구 커서를 누르면 몇 턴동안 그 행동만 하게 됩니다. 버퍼 문제인 것 같지만, 버퍼를 비우려고 하면 원래 터미널 창으로 나가지는 버그가 발생하는데다가 ncurses 자료도 그다지 많지 않아 방법은 찾지 못했습니다.

-확률이 자연스럽지 않습니다.

C++ 표준 라이브러리에서 제공하는 랜덤 함수를 사용해서 난수를 사용했는데 뭔가 난수를 이상하게 뽑아주는지 서로서로 플러시 족보가 나오는 경우가 잦습니다. 난수 자체를 어찌할 수는 없으니 아쉬운 부분입니다.