[오픈소스 소프트웨어 기초 프로젝트 일지 3회차]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 일시 | 2019년 12월 13일 18:00 ~ 23:00 | | | |
| 팀원 | 윤성빈 | 2019082206 | 김현진 | 2019089034 |
| 토의내용 | **보고서 작성: 2019082206 윤성빈**   1. **미니게임(숫자맞추기게임) 방법**   : 숫자맞추기게임은 사용자가 잘못된 선택지를 골라 사망 위기에 처했을 때 주어지는 생존 기회이다. 이를 통과해야 다음 칸으로 전진할 수 있다.  컴퓨터가 임의로 세자리 수를 만든다. 이 때, 각 자리의 수는 모두 달라야 하며 첫째 자리 수에 0을 입력할 수 없다. 이 세자리 수를 주어지는 단서, 즉 strike와 ball을 이용해 맞추는 게임이다. 단, 기회는 8번이다.  1. Strike: 숫자와 위치가 전부 맞음  2. Ball: 숫자는 맞았지만 위치가 틀림.  3. Out: 숫자와 위치가 전부 틀림.  예를 들어, 컴퓨터가 만든 세 자리 수를 172 이라고 해보자. 이 때 사용자가 123 이라는 수로 추측하여 이를 입력한다면, 1은 숫자와 위치 모두 맞으므로 strike: 1 이라고 호출될 것이고, 2는 숫자는 맞지만 위치가 다르므로 ball: 1 이라고 호출될 것이다. 나머지 3은 숫자와 위치 모두 틀렸으므로 out이다. 총 기회가 8번이므로 단서를 얻을 수 있는 기회는 7번이다. 이 안에 모든 숫자와 위치를 알아내서 입력해야 한다.  **<함수 분석 및 코드 설명>**   1. 숫자야구게임 코드      * **def number\_baseball()**   : 숫자 야구 게임을 관장하는 함수   * **def get\_random\_number()**   : 100부터 1000까지의 숫자를 임의로 생성하는 함수     * **def is\_between\_100\_and\_999(user\_input\_number)**   : 사용자가 입력하는 수가 알맞은 수인지 true 또는 false를 리턴하는 함수     * **def is\_duplicated\_number(three\_digit)**   : 사용자로부터 입력 받은 수를 한 자리씩 떼어내서 유효한 수인지 확인하는 함수     * **def is\_validated\_number(user\_input\_number)**   : 사용자가 입력한 값에 대해 조건에 맞지 않는 수를 확인하는 함수     * **def get\_not\_duplicated\_three\_digit\_number()**   : 임의로 생성된 수(답)와 사용자의 수가 일치하는지 확인하는 함수     * **def get\_strikes\_or\_ball(user\_input\_number, random\_number)**   : 랜덤으로 생성된 수와 사용자로부터 입력받은 수를 비교하여 strike 또는 ball의 상황을 출력하는 함수     * **def main()**   : 숫자 야구 게임 def number\_baseball()의 메인 함수로, 게임이 시작되기 전, 후 문장 출력과 게임 시 문장 출력, 사용자의 입력값을 받아 유효한 수인지 함수를 이용해 확인하고 이를 다시 함수를 이용해 랜덤으로 생성한 수와 일치하는 지 판별하는 역할을 한다.   1. **미니게임(스도쿠게임) 방법**   : 스도쿠게임은 숫자맞추기게임과 같이 사용자가 잘못된 선택지를 골라 사망 위기에 처했을 때 주어지는 생존 기회이다. 이를 통과해야 다음 칸으로 전진할 수 있다.  가로와 세로 각각 6칸씩, 모두 36칸으로 이루어진 정사각형의 빈칸을 채우는 게임이다. 이를 다시 ‘구역’ 이라고 하는 가로 세로 각 2칸, 모두 4칸인 작은 사각형으로 나눈다. 몇몇 자리에는 이미 수가 채워져 있다. 채워진 숫자들을 이용하여 나머지 빈칸에 1부터 6까지의 숫자가 겹치지 않도록 채워나간다. 각 세로줄과 가로줄의 빈 칸에는 숫자를 한 번만 사용해야 한다.  사용자는 먼저 난이도(상, 중, 하)를 선택한다. 난이도에 따라 빈칸의 수가 정해진다. 난이도가 높을수록 빈칸이 많다.  추론한 수를 입력하기 위해서는 먼저 가로의 몇 번째 줄인지, 세로의 몇 번째 줄인지 입력해야한다. 순서대로 입력 후 사용자가 추론한 수를 입력한다.  **<함수 분석 및 코드 설명>**   1. 스도쿠게임 코드      * **def create\_board()**   : 무작위 추출과 배열을 이용해 게임판(board)을 생성하는 함수이다.     * **def transpose(board)**   : 게임판(board)의 행과 열을 바꾸는 함수로, 게임판의 각 행과 열끼리의 무작위 생성 역할을 한다.     * **def shuffle\_ribbons(board) & def shuffle\_ribbons\_2(board)**   : 게임판(board)을 top\_half, mid\_half, bottom\_half 이렇게 세부분으로 나누어 이들을 무작위로 섞는 함수로, 게임판의 무작위 생성 역할을 한다.     * **def create\_solution\_board()**   : 정답 board를 따로 생성하는 함수이다. 이는 사용자가 입력한 수로 만들어진 board와 비교하여 정답인지 오답인지를 판별하게 한다.     * **def copy\_board(board)**   : 만들어진 게임판(board)를 복사하여 생성하는 함수이다.     * **def get\_level()**   : 스도쿠 게임의 레벨을 상,중,하 로 나누고 사용자로부터 레벨을 입력 받아 오는 함수이다.     * **def make\_holes(board, no\_of\_holes)**   : 게임판(board)에 사용자가 입력할 부분, 즉, 구멍을 만드는 함수이다.     * **def show\_board(board)**   : 게임판(board)의 뼈대와 숫자가 아닌 ‘.’으로 만들어진 구멍을 구성하고, 이를 호출하여 사용자에게 게임판(board)을 보여준다.     * **def get\_integer(message,i,j)**   : 사용자로부터 가로줄, 세로줄, 정답이 되는 수를 입력 받는 함수이다.     * **def sudokmini\_6()**   : 스도쿠 게임의 메인 함수로, 게임 시 문구 출력, get\_integer 함수를 이용한 사용자의 값 입력, 사용자로부터 입력 받은 수와 ‘.’ 으로 메워진 빈칸에 들어가야할 수가 일치하는 지 판별하여 결과를 보여주는 역할을 한다. | | | |
| 결론/끝맺음 | * 본 게임에서 잘못된 선택지를 골랐을 경우 출력되는 미니게임(숫자야구게임, 스도쿠게임을 만들었다. * 스도쿠게임에서는 게임판(board)을 만들 때 행 안의 숫자끼리 섞고, 각 행끼리 섞고, 각 열끼리 섞고, 보드를 세 부분으로 나누어 그들끼리 또 섞어야 하는 알고리즘을 제대로 이해하게 되었다. * 본 게임(스토리 게임) 안에 미니게임을 넣어 만들어 보는 것은 처음이라 낯설었지만 짜임새 있게 체계적인 코드가 완성되어 가는 것을 보며, 또 이러한 숫자 야구 게임과 스도쿠 게임의 원리를 이해하고 직접 플레이해보며 즐거웠다. | | | |