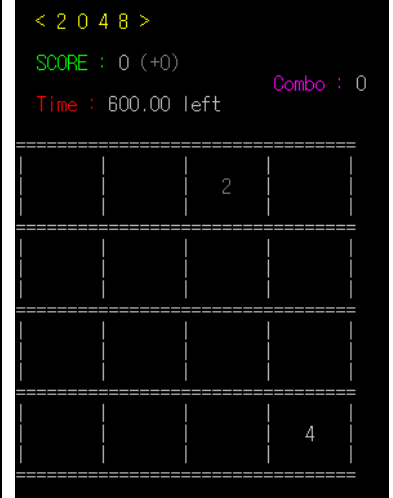


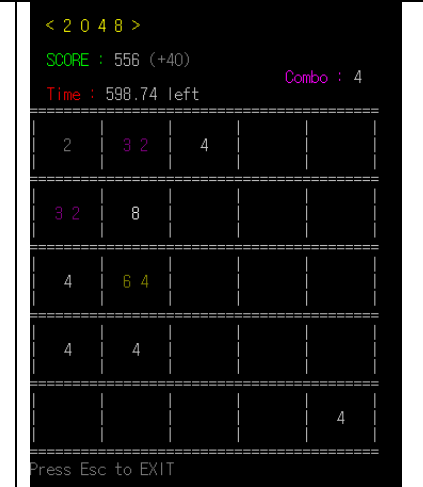



게임은 W, A, S, D키를 이용해 플레이한다. 각 키는 블록을 위, 왼쪽, 아래, 오른쪽으로 몰아넣는 기능을 한다. 이 과정에서 맞닿은 숫자들이 같다면 그 숫자들은 서로 합쳐져 하나의 블록이 된다.

		
초기 화면. 랜덤한 두 칸에 2 또는 4가 있다.	패배 화면. 어느 방향으로도 더 이상 움직일 수 없다.	승리 화면. 2048 블록이 생성되는 즉시 게임에서 승리한다.

3) how to implement several features

<pre>int randoms; int randomy; int randomnumber=rand()%4; if(randomnumber==3){randomnumber=1;}else{randomnumber=2;} while(1){ randomx=rand()%4; randomy=rand()%4; if(board4[randomx][randomy]==0){ board4[randomx][randomy]=randomnumber; break; } }</pre>	<pre>int returnvalue=0; if(*a1==*a2&&*a2!=*a3&&*a3!=*a4){ if(*a1==*a2){returnvalue=0;} else{returnvalue= *a2+*a1;} *a2+=*a1; *a1=0; return returnvalue; }else if((*a1!=*a2&&*a2==*a3&&*a3!=*a4){ if(*a2==*a3){returnvalue= 0;} else{returnvalue= *a3+*a2;} *a3+=*a2; *a2=*a1; *a1=0; return returnvalue; }</pre> <p>SCORE : 148 (+40)</p>	
랜덤 숫자 생성은 위 함수를 통해 구현된다.	moving()에서 합쳐진 숫자가 return되고, 이 값은 변수 this turnscore에 더해진다. 이동이 종료되면 이동으로 얻은 점수가 전체 score에 합산된다.	대부분의 함수는 5x5 모드를 위한 버전이 따로 존재한다. 이를 이용하여 5x5 모드 2048을 구현하였으며, 기본적인 규칙은 동일하다.
		<pre>clock_t end=clock(); float losttime=((float)end-(float)start)/(float)CLOCKS_PER_SEC; timeleft-=losttime; if(timeleft<0){timeleft=0;} if(timeleft==0){ ending=-1; }</pre> <p>SCORE : 344 (+24) Time : 597.47 left Combo : 4</p>

메인 메뉴, W와 S키로 화살표를 움직일 수 있다. 이는 입력마다 화면을 새로 그려서 구현했다. Exit는 게임을 종료한다.	How To Play. 간단한 설명이 적혀 있으며 Esc를 누르면 메인 메뉴로 돌아간다. while, getch()로 입력을 기다린다.	남은 시간은 clock_t를 통해 매 움직임마다 계산 및 차감되며 화면에 업데이트된다. 콤보의 구현은 이동 전후 빈칸 개수의 비교로 이루어진다.
---	---	--

<pre>===== 4 X 4 Ranking ===== 이름 클리어여부 점수 남은 시간 최대콤보 이동횟수 ===== 전승하 CLEAR 4818 135.48 32 230 Test2 CLEAR 3218 3.48 12 351 Test3 LOSE 2876 230.48 6 182 클리어 CLEAR 2108 593.80 3 16 Test1 CLEAR 1853 47.56 15 291 ===== Press Esc to EXIT 이름 : 김철수 점수 : 656 남은 시간 : 599.35 최대콤보 : 12 이동횟수 : 82 Press Esc to EXIT</pre>		<p>프로그램 시작 시 테스트용 랭킹 txt 파일이 작성된다. 그리고 각 게임이 클리어되거나 실패할 때마다 플레이어의 이름을 입력받고, 이 정보는 fprintf를 통해 txt 파일에 업데이트된다. 메인 메뉴에서 Ranking에 들어가면 이 txt 파일을 불러오게 되고, 점수 순서에 따라 정렬되어 출력된다. 랭킹은 4x4와 5x5를 따로 저장하고 볼 수 있도록 구현되었다.</p>
---	--	--

4) troubleshooting points

랭킹에서 점수를 정렬하는 방법으로는, 각 플레이어의 점수를 배열에 저장하고 반복문을 통해 자신보다 높은 점수의 개수를 세서 플레이어별로 새로운 배열에 저장하였다. 이를 통해 각 플레이어의 순위를 결정할 수 있었다.

게임 중 Esc를 누르면 메인 메뉴로 돌아갈 수 있다. Esc를 누르면 break;을 통해 반복문을 빠져나온다. 변수를 변경하고 나중에 체크하게 하여 이 때는 랭킹 등록 화면이 나오지 않도록 했다.

랭킹 메뉴나 플레이 메뉴에서는 4x4와 5x5를 선택할 수 있는 화면이 등장한다. 이 역시 system 함수로 화면을 모두 지우고, 다시 출력하는 방식으로 구현하였다.

Combo가 현재 최대 콤보를 넘어서면 이를 저장하도록 하여, 게임 종료 및 랭킹 화면에서 최대 콤보 숫자를 활용할 수 있게 하였다.

보드판 배열에 저장된 숫자를 출력할 때 사용할 printnum 함수를 구현하였다. 이 함수는 숫자에 따라 다른 색을 출력하고, 간격을 조절하여 전체 보드판의 모양이 망가지지 않도록 하였다.

게임을 승리하거나 패배하면 보드판 테두리의 색깔이 그에 맞게 변하도록 하여 시각적인 효과를 주었다.