**Programing Environment**

Windows 10 Home 1607ver

Microsoft Visual Studio 2015

**Source code& Brief description**

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include<time.h>  #include<math.h>  #define MAX\_SIZE 100  int team\_num, teamA\_life, teamB\_life, winner = 0; //team\_Life 는 팀 전체 목숨의 합  typedef struct teamates {  char \*teamname;  int id;  int weapon;  int life;  } teamate;  int getWeapon(int weapon) { //50%확률로 적의 무기와 교환  srand(time(NULL));  int luck = (rand() % 2);  if (luck == 0)  return weapon;  }  void selection(teamate man[], char name[], int num) //num개의 구조체 생성, teamname 으로 구분  {  for (int i = 0; i < num; i++)  {  man[i].teamname = name;  man[i].id = i + 1;  man[i].weapon = (rand() % 3) + 1;//가위 = 1, 바위 = 2, 보 = 3  man[i].life = 1;  }  }  int main(void)  {  char teamA[MAX\_SIZE], teamB[MAX\_SIZE];  teamate \*team\_A, \*team\_B;  int i = 0, j = 0;  char ans[10];  srand(time(NULL));  printf("Input player's team name : ");  gets(teamA);  printf("Input opponent's team name : ");  gets(teamB);  printf("Input number of players in each team : ");  scanf\_s("%d", &team\_num); //  printf("가위 = 1, 바위 = 2, 보 = 3 \n");  teamA\_life = team\_num;  teamB\_life = team\_num;  team\_A = (int \*)malloc(team\_num \* sizeof(teamate)); //team\_num의 구조체가 들어갈 수 있는 메모리 할당  team\_B = (int \*)malloc(team\_num \* sizeof(teamate));  for (int i = 0; i < team\_num; i++) { //구조체 생성  selection(team\_A, teamA, team\_num);  selection(team\_B, teamB, team\_num);  }  do { //적의 아이디 입력을 통해 다음 상대 선택  printf("Choose the player(ID) in the opponent's team. \n");  scanf\_s("%d", &j);  if (j > team\_num + 1 || j < 1)  continue;  j = j - 1;  if (team\_B[j].life == 0) //선택한 적의 목숨이 0일 경우 재선택  continue;  else break;  } while (1);  do {  if (teamB\_life == 0 || teamA\_life == 0) //한팀의 목숨의 합이 0이 될 때 종료  break;  for (int i = 0; i < team\_num; i++) //각 팀의 상황 출력  printf("팀이름 : %s ID : %d 무기 : %d 목숨 : %d\n", team\_A[i].teamname, team\_A[i].id, team\_A[i].weapon, team\_A[i].life);  printf("\n");  for (int i = 0; i < team\_num; i++)  printf("팀이름 : %s ID : %d 무기 : 0 목숨 : %d\n", team\_B[i].teamname, team\_B[i].id, team\_B[i].life);//무기상태 출력하지 않음  printf("\n");  if (team\_A[i].weapon == team\_B[j].weapon) { //무승부시 무기 재분배  printf("DRAW!! \n");  team\_A[i].weapon = (rand() % 3) + 1;  team\_B[i].weapon = (rand() % 3) + 1;  continue;  }  if (team\_A[i].weapon == team\_B[j].weapon + 1 || team\_A[i].weapon == team\_B[j].weapon - 2) { //teamA의 승리  printf("%s %d Win!! \n", team\_A[i].teamname, i + 1);  teamB\_life--; //teamB의 목숨 1 감소  team\_B[j].life--; //teamB의 플레이어 목숨 = 0  getchar();  for (int i = 0; i < team\_num; i++) //각 팀의 상황 출력  printf("팀이름 : %s ID : %d 무기 : %d 목숨 : %d\n", team\_A[i].teamname, team\_A[i].id, team\_A[i].weapon, team\_A[i].life);  printf("\n");  for (int i = 0; i < team\_num; i++)  printf("팀이름 : %s ID : %d 무기 : 0 목숨 : %d\n", team\_B[i].teamname, team\_B[i].id, team\_B[i].life);  printf("\n");  if (teamB\_life == 0 || teamA\_life == 0) //한팀의 목숨의 합이 0이 될 때 종료  break;  printf("Do you want exchange the weapon?(Y/N) \n");//무기 교환 의사 질문  gets(ans);  if (ans == "Y") //교환 의사가 있는 경우 확률에 따라 교환  team\_A[i].weapon = getWeapon(team\_B[j].weapon);  do { //적의 아이디 입력을 통해 다음 상대 선택  printf("Choose the player(ID) in the opponent's team. \n");  scanf\_s("%d", &j);  if (j > team\_num + 1 || j < 1)  continue;  j = j - 1;  if (team\_B[j].life == 0) //선택한 적의 목숨이 0일 경우 재선택  continue;  else break;  } while (1);  for (int i = 0; i < team\_num; i++) //각 팀의 상황 출력  printf("팀이름 : %s ID : %d 무기 : %d 목숨 : %d\n", team\_A[i].teamname, team\_A[i].id, team\_A[i].weapon, team\_A[i].life);  printf("\n");  for (int i = 0; i < team\_num; i++)  printf("팀이름 : %s ID : %d 무기 : 0 목숨 : %d\n", team\_B[i].teamname, team\_B[i].id, team\_B[i].life);  printf("\n");  continue;  }  if (team\_A[i].weapon == team\_B[j].weapon - 1 || team\_A[i].weapon == team\_B[j].weapon + 2) { //teamB의 승리  printf("%s %d Win!! \n", team\_B[j].teamname, j + 1);  teamA\_life--; //teamA의 목숨 1 감소  team\_A[i].life--; //teamA의 플레이서 목숨 = 0  int luck = (rand() % 2); //50%확률로 적의 무기 변화  if (luck == 0)  team\_B[j].weapon = getWeapon(team\_A[i].weapon);  i++; //teamA의 다음 선수 출전  for (int i = 0; i < team\_num; i++) //각 팀의 상황 출력  printf("팀이름 : %s ID : %d 무기 : %d 목숨 : %d\n", team\_A[i].teamname, team\_A[i].id, team\_A[i].weapon, team\_A[i].life);  printf("\n");  for (int i = 0; i < team\_num; i++)  printf("팀이름 : %s ID : %d 무기 : 0 목숨 : %d\n", team\_B[i].teamname, team\_B[i].id, team\_B[i].life);  printf("\n");  continue;  }  } while (1);  winner = (teamA\_life < teamB\_life) ? 1 : -1;//목숨수가 많은 팀이 승리  if (winner > 0)  printf("winner is %s!\n", teamB);  else if (winner < 0)  printf("winner is %s!\n", teamA);  free(team\_A); //메모리 할당 해제  free(team\_B);  return 0;  } |

**Program Description**

Purpose : 가위바위보 게임

Time Complextiy : T(n^2)

Space Complexity : O(n^2)

**Screenshot of the Results**

