**Programing Environment**

Windows 10 Home

Intel(R) Core(TM) i3-6300 CPU @ 2.30GHz

VMware Workstation Player 12.1.1

Ubuntu Linux 64

Ubuntu Desktop 16.04 LTS

Memory 2GB

**Source code& Brief description**

|  |
| --- |
| #define \_GNU\_SOURCE //유닉스 기반이 아닌 소스 정의  #define \_CRT\_SECURE\_NO\_WARNINGS //scanf\_s 대신 scanf 사용하기 위함  #include<stdio.h> //표준 입출력 라이브러리  #include<string.h> //문자열을 다를수 있는 함수 포함  #include<stdlib.h> //문자열 변환, 난수 생성, 동적메모리할당  #include<unistd.h> //Standard symbolic constans and types synopsis  #include<sys/utsname.h> //system name structure synopsis  #include<time.h>  #include<math.h>  #define MAX\_SIZE 100  int luck, team\_num, teamA\_life, teamB\_life, winner = 0;  char temp[MAX\_SIZE];  char \*teamA, \*teamB, \*ans;  int i = 0, j = 0;  typedef struct teamates { //구조체 형식  char \*teamname;  int id;  int weapon;  int life;  } teamate;  int getWeapon(int weapon) { //랜덤으로 무기를 받음  luck = (rand() % 2);  if (luck == 0)  return weapon;  }  void selection(teamate man[], char name[], int num) { //구조체 생성  int i;  for (i = 0; i < num; i++) {  man[i].teamname = name;  man[i].id = i + 1;  man[i].weapon = (rand() % 3) + 1;  man[i].life = 1;  }  }  void printInfoA(teamate man[]) { //자신의 팀상황 출력  int i;  for (i = 0; i < team\_num; i++)  printf("Team name : %s ID : %d weapon : %d Life : %d\n", man[i].teamname, man[i].id, man[i].weapon, man[i].life);  printf("\n");  }  void printInfoB(teamate man[]) { //적의 상황 출력, 단 무기는 공개 x  int i;  for (i = 0; i < team\_num; i++)  printf("Team name : %s ID : %d weapon : 0 Life : %d\n", man[i].teamname, man[i].id, man[i].life);  printf("\n");  }  int main(void) {  teamate \*team\_A, \*team\_B;  srand(time(NULL)); //난수발생  printf("Input player's team name : ");  fgets(temp, sizeof(temp), stdin);  teamA = (char\*)malloc(strlen(temp) + 1);  strcpy(teamA, temp);  printf("Input opponent's team name : ");  fgets(temp, sizeof(temp), stdin);  teamB = (char\*)malloc(strlen(temp) + 1);  strcpy(teamB, temp);  printf("Input number of players in each team : ");  scanf("%d", &team\_num);  teamA\_life = team\_num;  teamB\_life = team\_num;  team\_A = (teamate \*)malloc(team\_num \* sizeof(teamate)); //구조체를 위한 메모리 할당  team\_B = (teamate \*)malloc(team\_num \* sizeof(teamate));  printf("\n");  printf("Scissor = 1, Rock = 2, Paper = 3 \n"); //가위 = 1, 바위 = 2, 보 = 3  printf("\n");  int k;  for (k = 0; k < team\_num; k++) {  selection(team\_A, teamA, team\_num);  selection(team\_B, teamB, team\_num);  }  printInfoA(team\_A); //현재 상황 출력  printInfoB(team\_B);  do { //상대편 지목  printf("Choose the player(ID) in the opponent's team. \n");  scanf("%d", &j);  if (j > team\_num + 1 || j < 1)  continue;  j = j - 1;  if (team\_B[j].life == 0) //지목한 상대의 목숨이 0일 경우 재실행  continue;  else  break;  } while (1);  do {  printInfoA(team\_A);  printInfoB(team\_B);  if (teamB\_life == 0 || teamA\_life == 0) //아군 혹은 적군의 전체 목숨이 0이 될 경우 가위바위보 종료  break;  if (team\_A[i].weapon == team\_B[j].weapon) { //무승부일 때 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) { //승리  printf("%s %d Win!! \n", team\_A[i].teamname, i + 1);  teamB\_life--; //적 전체의 목숨의 합 1감소  team\_B[j].life--; //적 플레이어의 목숨 감소    if (teamB\_life == 0 || teamA\_life == 0)  break;  printInfoA(team\_A);  printInfoB(team\_B);  printf("Do you want exchange the weapon?(Y/N) \n"); //무기변경 의사 질문  fgets(temp, sizeof(temp), stdin);  ans = (char\*)malloc(strlen(temp) + 1); //메모리 할당  strcpy(ans, temp);  if (strcmp("Y\n", ans) == 0) { //동의 의사를 밝히는 경우  team\_A[i].weapon = getWeapon(team\_B[j].weapon);  if (team\_A[i].weapon == team\_B[j].weapon)  printf("weapon is changed to %d\n", team\_A[i].weapon);  else  printf("weapon is not changed \n");  }  free(ans); //할당 해제  do {  printf("Choose the player(ID) in the opponent's team. \n");  scanf("%d", &j);  if (j > team\_num + 1 || j < 1)  continue;  j = j - 1;  if (team\_B[j].life == 0) continue;  else break;  } while (1);  continue;  }  if (team\_A[i].weapon == team\_B[j].weapon - 1 || team\_A[i].weapon == team\_B[j].weapon + 2) { //패배시  printf("%s %d Win!! \n", team\_B[j].teamname, j + 1);  teamA\_life--;  team\_A[i].life--;  team\_B[j].weapon = getWeapon(team\_A[i].weapon);  i++;  continue;  }  } while (1);  winner = (teamA\_life < teamB\_life) ? 1 : -1; //승자 판별  if (winner > 0)  printf("winner is %s!\n", teamB);  else if (winner < 0)  printf("winner is %s!\n", teamA);  free(team\_A); //할당해제  free(team\_B); //할당해제  system("pause");  return 0;  } |

**Program Description**

Purpose : 가위바위보 게임

Time Complexity : T(n^2)

Space Complexity : O(n+m)

**Screenshot of the Results**

