## Код программы

%character(Name, HP, Dmg, Armor)

player1(character(Name, HP, Dmg, Armor)):-  name1(Name),

                                            reset\_hp1(HP),

                                            reset\_dmg1(Dmg),

                                            reset\_armor1(Armor).

name1("Player1").

reset\_hp1(HP):- HP=5,

                retractall(hp1(\_)),

                asserta(hp1(HP)).

reset\_dmg1(Dmg):-   Dmg=2.5,

                    retractall(dmg1(\_)),

                    asserta(dmg1(Dmg)).

reset\_armor1(Armor):-   Armor=0.75,

                        retractall(armor1(\_)),

                        asserta(armor1(Armor)).

assert2(X):-asserta(X).         %, format('assert( ~w )\n', [X]).

assert2(X):-retract(X), fail.   %, format('retract( ~w )\n', [X]).

change\_hp1(D):- hp1(HP),

                NewHP is HP+D,

                %retractall(hp1(HP)),

                assert2(hp1(NewHP):-!).

change\_dmg1(D):-dmg1(Dmg),

                NewDmg is Dmg+D,

                %retractall(dmg1(Dmg)),

                assert2(dmg1(NewDmg):-!).

change\_armor1(D):-  armor1(Armor),

                    NewArmor is Armor+D,

                    %retractall(armor1(Armor)),

                    assert2(dmg1(NewArmor):-!).

player2(character(Name, HP, Dmg, Armor)):-  name2(Name),

                                            reset\_hp2(HP),

                                            reset\_dmg2(Dmg),

                                            reset\_armor2(Armor).

name2("Player2").

reset\_hp2(HP):- HP=5,

                retractall(hp2(\_)),

                asserta(hp2(HP)).

reset\_dmg2(Dmg):-   Dmg=6,

                    retractall(dmg2(\_)),

                    asserta(dmg2(Dmg)).

reset\_armor2(Armor):-   Armor=0.4,

                        retractall(armor2(\_)),

                        asserta(armor2(Armor)).

change\_hp2(D):- hp2(HP),

                NewHP is HP+D,

                %retractall(hp2(HP)),

                assert2(hp2(NewHP):-!).

change\_dmg2(D):-dmg2(Dmg),

                NewDmg is Dmg+D,

                %retractall(dmg2(Dmg)),

                assert2(dmg2(NewDmg):-!).

change\_armor2(D):-  armor2(Armor),

                    NewArmor is Armor+D,

                    %retractall(armor2(Armor)),

                    assert2(armor2(NewArmor):-!).

%HIT

skill1(Skill):- dmg1(Dmg1), armor2(Armor2),

                DHP = Dmg1\*(1-Armor2), change\_hp2(-DHP),

                %name1(Player1), format('~s uses HIT dealing ~3f damage\n', [Player1, DHP]),

                Skill = "HIT1".

%SHIELD SLAM

skill1(Skill):- armor1(Armor1),

                DA = Armor1\*0.5, change\_armor2(-DA),

                armor2(NewArmor2),

                DHP = Armor1\*(1-NewArmor2), change\_hp2(-DHP),

                %name1(Player1), format('~s uses SHIELD SLAM decreasing enemy Armor by ~3f and dealing ~3f dmg\n', [Player1, DA, DHP]),

                Skill = "SHIELD SLAM1".

%HIT

skill2(Skill):- dmg2(Dmg2), armor1(Armor1),

                DHP = Dmg2\*(1-Armor1), change\_hp1(-DHP),

                %name2(Player2), format('~s uses HIT dealing ~3f damage\n', [Player2, DHP]),

                Skill = "HIT2".

%LIFE LEACH

skill2(Skill):- dmg2(Dmg2), armor1(Armor1),

                LL = Dmg2\*0.6\*(1-Armor1), change\_hp1(-LL), change\_hp2(LL),

                %name2(Player2), format('~s uses LIFE LEACH for ~3f dmg\n', [Player2, LL]),

                Skill = "LIFE LEACH2".

winner(Winner, \_, WinStrat, WinStrat):- hp1(HP1), (HP1 = 0.0 ; HP1 < 0.0), !,

                                        name2(Player2),

                                        %format('~s wins!!!!!!\n\n', [Player2]),

                                        Winner = Player2.

winner(Winner, \_, WinStrat, WinStrat):- hp2(HP2), (HP2 = 0.0 ; HP2 < 0.0), !,

                                        name1(Player1),

                                        %format('~s wins!!!!!!\n\n', [Player1]),

                                        Winner = Player1.

winner(Winner, N, Strat, WinStrat):-

    %format('Turn ~d\n', [N]),

    turn(N, Skill),

    %name1(Player1), hp1(NewHP1), dmg1(NewDmg1), armor1(NewArmor1),

    %format('~s: HP=~3f; Dmg=~3f; Armor=~3f\n', [Player1, NewHP1, NewDmg1, NewArmor1]),

    %name2(Player2), hp2(NewHP2), dmg2(NewDmg2), armor2(NewArmor2),

    %format('~s: HP=~3f; Dmg=~3f; Armor=~3f.\n', [Player2, NewHP2, NewDmg2, NewArmor2]),

    NN is N+1, append(Strat, [Skill], NewStrat),

    winner(Winner, NN, NewStrat, WinStrat).

turn(N, Skill):-T is (N mod 2), T = 1, skill1(Skill).

turn(N, Skill):-T is (N mod 2), T = 0, skill2(Skill).

get\_win\_strategy(Winner, Strat):-   player1(\_),

                                    player2(\_),

                                    winner(Winner, 1, [], Strat).

get\_all\_win\_strats1(Strats, WinsCount):-name1(Winner),

                                        findall(WinStrat, get\_win\_strategy(Winner, WinStrat), Strats),

                                        length(Strats, WinsCount).

get\_all\_win\_strats2(Strats, WinsCount):-name2(Winner),

                                        findall(WinStrat, get\_win\_strategy(Winner, WinStrat), Strats),

                                        length(Strats, WinsCount).

get\_win\_percentage(P1, P2):-get\_all\_win\_strats1(\_, WinsCount1),

                            get\_all\_win\_strats2(\_, WinsCount2),

                            P1 is WinsCount1/(WinsCount1+WinsCount2),

                            P2 is WinsCount2/(WinsCount1+WinsCount2).

%get\_win\_strategy(Winner, WinStrat).

%name1(Winner), get\_win\_strategy(Winner, WinStrat).

%name2(Winner), get\_win\_strategy(Winner, WinStrat).

%get\_all\_win\_strats1(Strats1, WinsCount1).

%get\_all\_win\_strats2(Strats2, WinsCount2).

%get\_win\_percentage(P1, P2).