```
function [theta]=LeastSquaresPolicyIteration(L, M, T, B, options, win_w,win_h)
  startSimulation (win_w,win_h);    % 本体のウィンドウを表示
  actions = [-50, 0, 50];           % 行動の候補
  nactions = 3;             % 行動数

  % デザイン行列X, ベクトルrの初期化
  X = zeros(M*T,B*nactions);
  r = zeros(M*T,1);

  % モデルパラメータの初期化
  theta = zeros(B*nactions,1);

  % 政策反復
  for l=1:L
    dr = 0;
    rand('state',1);

    % 標本
    for m=1:M
      resetSimulation();

      for t=1:T+1

        % 状態 (psi1, psi2, dpsi1, dpsi2) の観測
        state = getJointState();

        % 距離
        dist = sum((options.centers - repmat(state',B,1)).^2,2);

        % 現在の状態に関する基底関数
        phis = exp(-dist/2/(options.var^2));

        % 現在の状態における価値関数
        Q = phis'*reshape(theta,B,nactions);

        % 政策
        policy = zeros(nactions,1);
        switch options.pmode
          case 1 % greedy
            [v,a] = max(Q);
            policy(a) = 1;

          case 2 % e-greedy
            if l==1
              policy = ones(nactions,1)./nactions;
            else
              [v,a] = max(Q);
              policy = ones(nactions,1)*options.epsilon/nactions;
              policy(a) = 1 - options.epsilon+options.epsilon/nactions;
            end

          case 3  % softmax
            policy = exp(Q./options.tau)/sum(exp(Q./options.tau));
        end
```

```
        % 行動選択
        ran = rand;
        if(ran < policy(1))
          action = 1;
        elseif(ran < policy(1)+policy(2))
          action = 2;
        else
          action = 3;
        end
        u(2) = actions(action);

        % 行動の実行
        stepSimulation (u,0.005);
        if(t==0 || mod(t,10)==0)
           drawWorld;
        end

        if t>1
          % 現在の状態に関する基底関数の政策に関する平均
          aphi = zeros(B*nactions,1);
          for a=1:nactions
            aphi = aphi + getPhi(state,a,options.centers,B,options.var,nactions) * policy(a);
          end

          % 一つ前の状態と行動に関する基底関数
          pphi = getPhi(pstate,paction,options.centers,B,options.var,nactions);

          % (M*T)*Bデザイン行列X, M*T次元ベクトルr
          X(T*(m-1)+t-1,:) = (pphi - options.gamma * aphi)';
          r(T*(m-1)+t-1) = -cos(state(1));

          % 割引き和の計算
          dr = dr + r(T*(m-1)+t-1)*options.gamma^(t-1);
        end

        paction = action;
        pstate = state;
      end
    end

    % 政策評価
    theta = pinv(X'*X)*X'*r;

    printf("%d)Max=%.2f Avg=%.2f Dsum=%.2f numtop=%d¥n",l,max(r),mean(r),dr/M,size(find(r>0.9),1));
    fflush(stdout);
end
```