# **Unity完全自制游戏纸箱战争项目记录（20180709）**

作者：[Pluto](http://gad.qq.com/user/index?id=2267380)

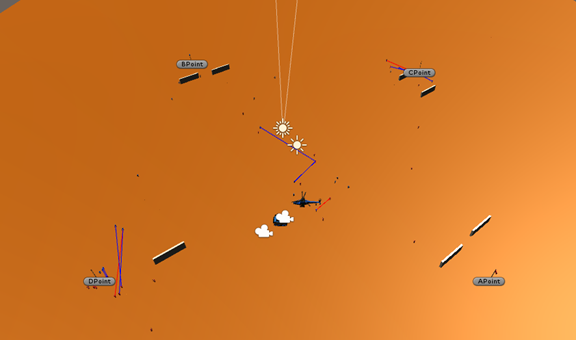
链接：<http://gad.qq.com/article/detail/228171>

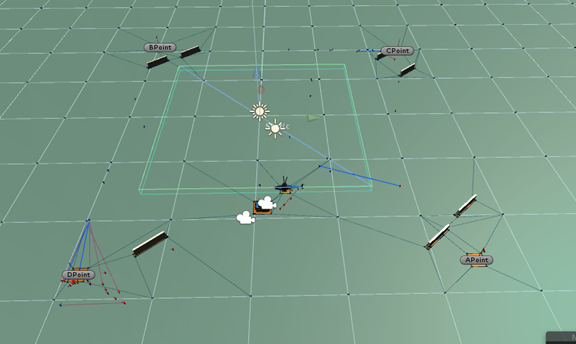
其实周六也是项目有些进展的，不过是在因为进展太少了，不值得单独写成一篇记录。

这两天主要都是在整合AI的功能实现，不得不说，写AI实在是太麻烦了，比想象中的还要麻烦的多得多。

今天实现了AI对目标发起攻击的算法设计，本来打算是要自己写出来一套躲避障碍物的算法，实在太难了，没办法不得不采用了Unity中自带的寻路导航系统。

如下图片是打印出了AI的设计检测射线和路径导航的预览图。





今天上午写了删删了又写，花了一上午的时间算是写出了一个AI目标判定的方法。

之前已经书写过了AI搜寻目标的方法TargetFinding（），在使用的时候只需要调用就行了，这就不需要在AI脚本中获取了。

我考虑到在AI运动的时候应该只有四个状态：

1.         AI没有攻击目标，AI处于占领区域内

2.         AI有攻击目标，AI处于占领区域内

3.         AI没有攻击目标，AI没有处于占领区域内

4.         AI有进攻目标，AI没有处于占领区域内

如果为3的情况，只需要让AI向着目标前进就行了，同时再路径行动的过程中搜索可以攻击的目标。

如果为4的情况，AI找到了攻击目标，但没有处在占领点内，这时候就需要让AI对目标发起攻击，但并不能耽误了进攻目标点。

如果为2的情况，就直接让AI进攻目标，并且在杀死了目标之后立即寻获目标。

如果为1的情况，就让AI呈现巡逻状态，始终处于寻获目标的状态，直到发现敌人。

以下代码为AI的职业为突击手时的AI算法：

    void CampIs1()//突击手AI

    ｛

        if (TemporaryTargetPoint == null || TemporaryTargetPoint.gameObject.GetComponent<TargetPoint>().PointValue == 1)

        ｛

            TargetFind();//目标点寻获,保证始终存在目标点

        ｝

        if ((transform.position - TemporaryTargetPoint.position).sqrMagnitude > (FlagRad \* FlagRad))

        ｛

            if (AttackTarget == null)

            ｛

                gameObject.transform.LookAt(new Vector3(TemporaryTargetPoint.position.x, transform.position.y, TemporaryTargetPoint.position.z));

                NavMeshAgentTransfromNavMeshAgent.SetDestination(TemporaryTargetPoint.position);

            ｝

            if (AttackTarget != null)//攻击目标不为空并且距离攻击物体距离小于50

            ｛

                if ((transform.position - AttackTarget.position).sqrMagnitude < 2500)

                ｛

                    gameObject.transform.LookAt(new Vector3(AttackTarget.position.x, transform.position.y, AttackTarget.position.z));

                    Camp hitHumanCamp = AttackTarget.gameObject.GetComponent<Camp>();

                    if ((hitHumanCamp.Type > 0 && hitHumanCamp.Type <= 5) &&

                        hitHumanCamp.CampNumber != gameObject.GetComponent<Camp>().CampNumber) //判断目标物体是否为步兵单位并且不与自己同阵营

                    ｛

                        Ray ray =

                            new Ray(new Vector3(transform.position.x, transform.position.y + 1, transform.position.z),

                                transform.forward);

                        RaycastHit hit;

                        if (Physics.Raycast(ray, out hit)) //射线是否碰撞到物体

                        ｛

                            if (gameObject.GetComponent<Camp>().CampNumber == 2)

                            ｛

                                Debug.DrawLine(ray.origin, hit.point, Color.blue);

                            ｝

                            if (gameObject.GetComponent<Camp>().CampNumber == 1)

                            ｛

                                Debug.DrawLine(ray.origin, hit.point, Color.red);

                            ｝

                            if (hit.transform.gameObject == hitHumanCamp.gameObject) //判断是否被阻挡

                            ｛

                                hitHumanCamp.HP -= UnityEngine.Random.Range(0, 2);

                                if (hitHumanCamp.gameObject.GetComponent<AI\_Move>() != null)//伤害传导

                                ｛

                                    hitHumanCamp.gameObject.GetComponent<AI\_Move>().HitAttackTarget =

                                        gameObject.transform;

                                ｝

                                if (hitHumanCamp.HP <= 0) //伤害增加

                                ｛

                                    AttackTarget = null;

                                    TargetFinding();

                                ｝

                            ｝

                            else

                            ｛

                                NavMeshAgentTransfromNavMeshAgent.SetDestination(AttackTarget.position);

                            ｝

                        ｝

                    ｝

                ｝

                else

                ｛

                    AttackTarget = null;

                ｝

            ｝

        ｝

        if ((transform.position - TemporaryTargetPoint.position).sqrMagnitude <= (FlagRad \* FlagRad))

        ｛

            if (AttackTarget == null)

            ｛

                TargetFinding();

                if (UnityEngine.Random.Range(0, 101) > 70)

                ｛

                    transform.Rotate(Vector3.down, 2);

                ｝

            ｝

            if (AttackTarget != null)

            ｛

                if ((transform.position - AttackTarget.position).sqrMagnitude < 2500)

                ｛

                    gameObject.transform.LookAt(new Vector3(AttackTarget.position.x, transform.position.y, AttackTarget.position.z));

                    Camp hitHumanCamp = AttackTarget.gameObject.GetComponent<Camp>();

                    if ((hitHumanCamp.Type > 0 && hitHumanCamp.Type <= 5) && hitHumanCamp.CampNumber != gameObject.GetComponent<Camp>().CampNumber) //判断目标物体是否为步兵单位并且不与自己同阵营

                    ｛

                        Ray ray = new Ray(new Vector3(transform.position.x, transform.position.y + 1, transform.position.z), transform.forward);

                        RaycastHit hit;

                        if (Physics.Raycast(ray, out hit)) //射线是否碰撞到物体

                        ｛

                            if (gameObject.GetComponent<Camp>().CampNumber == 2)

                            ｛

                                Debug.DrawLine(ray.origin, hit.point, Color.blue);

                            ｝

                            if (gameObject.GetComponent<Camp>().CampNumber == 1)

                            ｛

                                Debug.DrawLine(ray.origin, hit.point, Color.red);

                            ｝

                            if (hit.transform.gameObject == hitHumanCamp.gameObject) //判断是否被阻挡

                            ｛

                                hitHumanCamp.HP -= UnityEngine.Random.Range(0, 2);

                                if (hitHumanCamp.gameObject.GetComponent<AI\_Move>() != null)//伤害传导

                                ｛

                                    hitHumanCamp.gameObject.GetComponent<AI\_Move>().HitAttackTarget =

                                        gameObject.transform;

                                ｝

                                if (hitHumanCamp.HP <= 0) //伤害增加

                                ｛

                                    AttackTarget = null;

                                    TargetFinding();

                                ｝

                            ｝

                            else

                            ｛

                                NavMeshAgentTransfromNavMeshAgent.SetDestination(AttackTarget.position);

                            ｝

                        ｝

                    ｝

                ｝

                else

                ｛

                    AttackTarget = null;

                ｝

            ｝

        ｝