

미니축구게임 **AI**

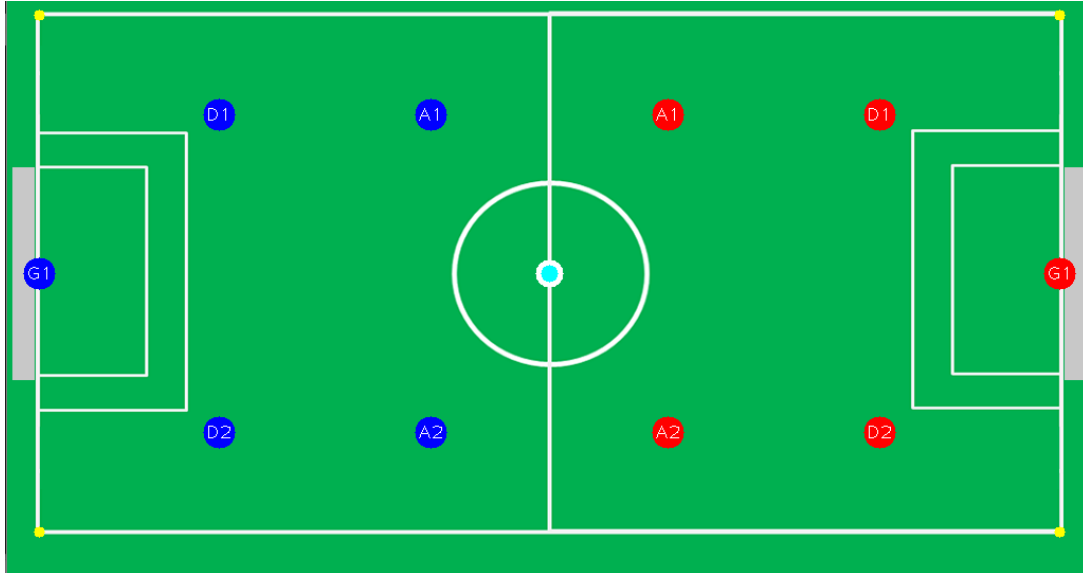
<목 차>

1. 목적 및 축구 게임 룰 설명
2. AI 설계 방향
 - 2.1. G1 (골키퍼)
 - 2.2. D1, D2 (수비수)
 - 2.3. A1, A2 (공격수)
3. 확률 기반 FSM
4. 부록

1. 목적 및 축구 게임 룰 설명

간단한 축구 게임 적용되는 AI를 설계한다. 게임 AI가 최대한의 득점과 최소한의 실점을 얻도록 설계하는 것이 목적이며, A1,A2는 확률에 기반한 FSM모델을 사용했다.

1.1. 축구 게임 룰 설명



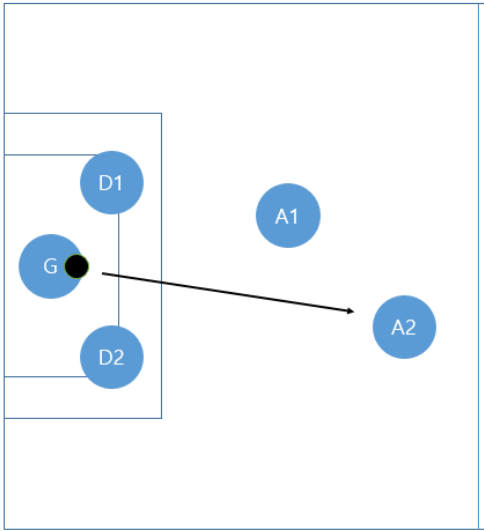
- G1(골키퍼) D1,D2(수비수), A1,A2(공격수) 5명이 한 팀
- 터치라인 밖은 아웃으로 생각하지 않는다.
- 골라인 밖은 아웃이다.
- 득점 시, 상대방이 먼저 공을 가지고 시작한다.
- 오프사이드는 없다.
- 공을 뺏을 확률은 동일하다
- 모든 선수의 이동속도는 동일하다. 하지만 드리블을 할 경우 속도는 20% 감소한다.
- 골키퍼는 골라인에서만 움직일 수 있다.

2. AI 설계 방향

수백번의 게임 실행 결과, 공격적인 플레이가 득점률이 높다. 이를테면, 상대방 골대 앞에서 몸싸움과 슈팅이 자주 일어날수록 득점으로 쉽게 할 수 있었다. 이를 근거로, 공격적인 AI를 설계할 예정이다.

2.1. G(골키퍼)

기본적으로 골키퍼는 공의 위치에 따라 x축은 고정, y축은 이동한다. 선방에 성공했을 경우, 공을 가장 멀리 떨어진 공격수에게 전달한다. 이는 많은 유효슈팅을 발생시키기 위함이다.

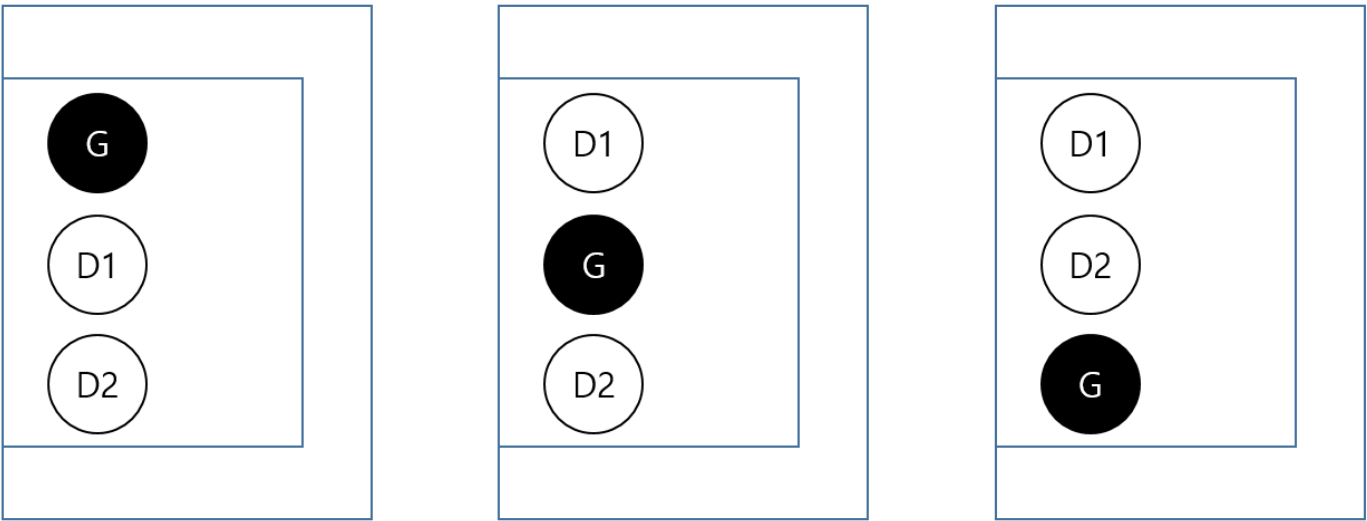


(그림 2.1.a)

2.2. D1, D2(수비수)

수비수는 최대한 수비적으로 플레이한다. 공격적인 플레이가 득점률이 높으나, 이는 공격기회가 많다는 전제가 필수이다. 수회 플레이 결과

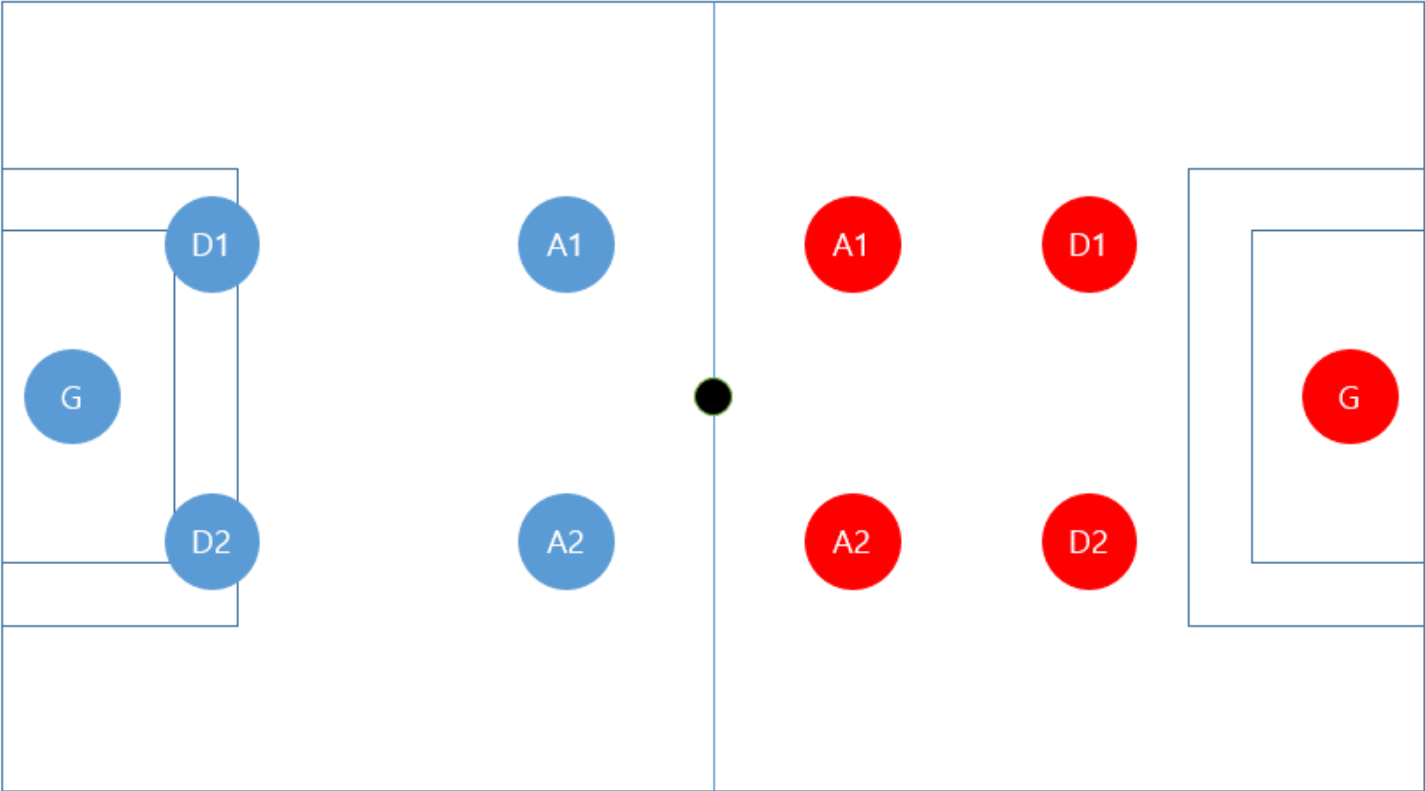
경기가 시작되면 수비수들은 우리팀 골대를 향해 특정 위치에 위치한다.



<수비수 예상 이동 상황판>
(그림1.3.b)

2.3 A1, A2(공격수)

공격수는 멋있다.



< 예상 상황판 >
(그림 1.3.a)

2. 코드 설명

2.1. G(골키퍼)

2.1.1. function Goalkeeper:FrameMove()

```
pos,getball ← GetHomeAgentPos(GOALKEEPER)
posA1,_ ← GetHomeAgentPos(OFFENSER1)
posA2,_ ← GetHomeAgentPos(OFFENSER2)

AgentMove(GOALKEEPER, t)                      =====> i)

IF getball == true THEN                        =====> ii)
  IF posA1.x > posA2.x THEN
    AgentPass(GOALKEEPER, OFFENSER1)
  else
    AgentPass(GOALKEEPER, OFFENSER2)
  ENDIF
ENDIF
```

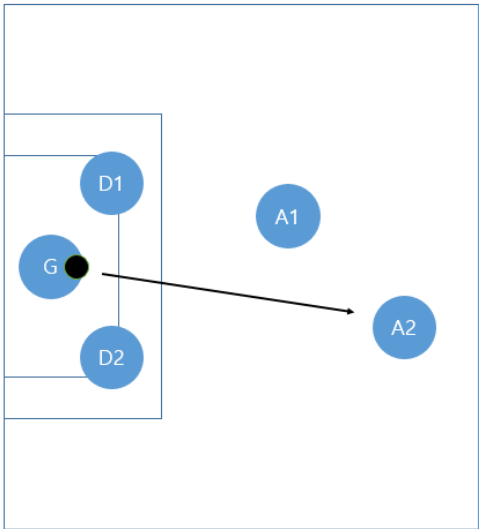
골키퍼의 행동 코드이다. 골키퍼는 움직임이 제한적이지만, 일직선상에서의 슈팅은 잘 선방한다. 공을 선방 후에는 공격기회를 위해 멀리있는 공격수에게 패스를 한다.

- i) 공을 향해 y 축으로 움직인다
- ii) 멀리있는 공격수에게 패스를 한다.

GetHomeAgentPos(x)

: return 값은 x의 위치와 공을 갖고 있는지, 없는지에 대한 bool값이다.

GetHomeAgentPos(OFFENSERn)함수를 이용하여, posA1, posA2의 값을 가져온다.
대소 비교를 통해 ii) 멀리있는 공격수에게 패스한다.



(그림 2.1.a)

2.2. D1, D2(수비수)

2.2) D1 코드

2.2.1. function Defender:FrameMove() : D1

```

posSelf,getballSelf = GetHomeAgentPos(self.AgentID)
posGK,getBallGK = GetHomeAgentPos(GOALKEEPER)
posA1,getBallA1 = GetHomeAgentPos(OFFENSER1)
posA2,getBallA2 = GetHomeAgentPos(OFFENSER2)

d1_location ← vector(d1_aim_x, d1_aim_y) - posSelf
goal_Mid ← vector(gk_mid_x, gk_mid_y) - posSelf

d1_location:norm()
goal_Mid:norm()

IF getballSelf == true THEN
  IF posA1.x < ground_mid_x or posA2.x < ground_mid_x THEN =====> i)
    AgentShoot(DEFFENDER1)
  ENDIF

  IF posA1.x < posA2.x THEN                                =====> ii)
    AgentPass(self.AgentID,OFFENSER2)
  ELES
    AgentPass(self.AgentID,OFFENSER1)
  ENDIF
ENDIF

IF posSelf.x ~= d1_location.x THEN                        =====> iii)
  AgentMove(self.AgentID, d1_location)

  IF posGK.y > 234 THEN                                    =====> iv)
    AgentMove(self.AgentID, d1_location)
  ENDIF

```

```

IF posGK.y < 234 THEN
    AgentMove(self.AgentID, goal_Mid)
ENDIF
ENDIF

```

수비수 D1의 행동코드이다.

수비수는 골키퍼의 미흡한 수비를 돕는다. 특정 위치는 골키퍼의 위치에 따라 결정된다.

- i) 공격 진형이 아군 팀에 형성되어있으면 공을 멀리 보낸다.
- ii) 수비에 성공했으면 가장 먼 거리에 있는 공격수에게 공을 패스한다.
- iii) 수비수를 특정 위치로 이동시킨다.
- iv) 수비수의 움직임 수비진형을 만든다.

2.2.1.1) 변수

```

posSelf.getBallSelf = GetHomeAgentPos(self.AgentID)
posGK.getBallGK = GetHomeAgentPos(GOALKEEPER)
posA1.getBallA1 = GetHomeAgentPos(OFFENSER1)
posA2.getBallA2 = GetHomeAgentPos(OFFENSER2)

```

GetHomeAgentPos(x) 를 이용하여, 위치와 공소유 여부를 가지고 온다.

```

d1_location ← vector(d1_aim_x, d1_aim_y) - posSelf
goal_Mid ← vector(gk_mid_x, gk_mid_y) - posSelf

d1_location: norm()
goal_Mid: norm()

```

d1_aim_x, d1_aim_y는 수비수가 가야할 특정 위치의 x, y좌표이다.
d1_loaction, posSelf(자신의 위치벡터)로 d1_location을 구한다.

gk_mid_x, gk_mid_y는 골대의 중심의 x, y좌표이다.
벡터연산으로 goal_Mid(골대의 중심벡터)를 구한다.

norm()
벡터를 정규화시키는 함수이다.

2.2.1.2) 수비수 상호작용

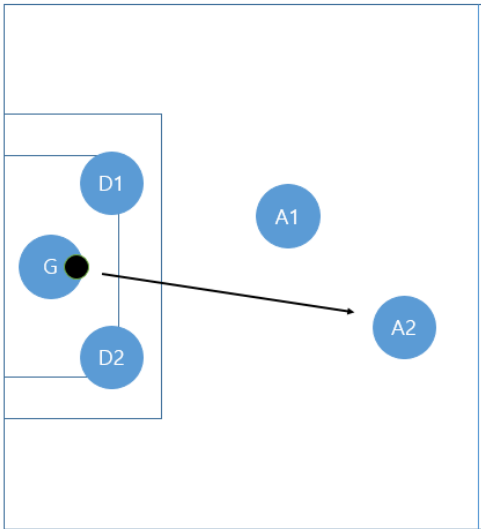
```
IF getballSelf == true THEN
  IF posA1.x < ground_mid_x or posA2.x < ground_mid_x THEN =====> i)
    AgentShoot(DEFFENDER1)
  ENDIF

  IF posA1.x < posA2.x THEN                                     =====> ii)
    AgentPass(self.AgentID,OFFENSER2)
  ELES
    AgentPass(self.AgentID,OFFENSER1)
  ENDIF
ENDIF
```

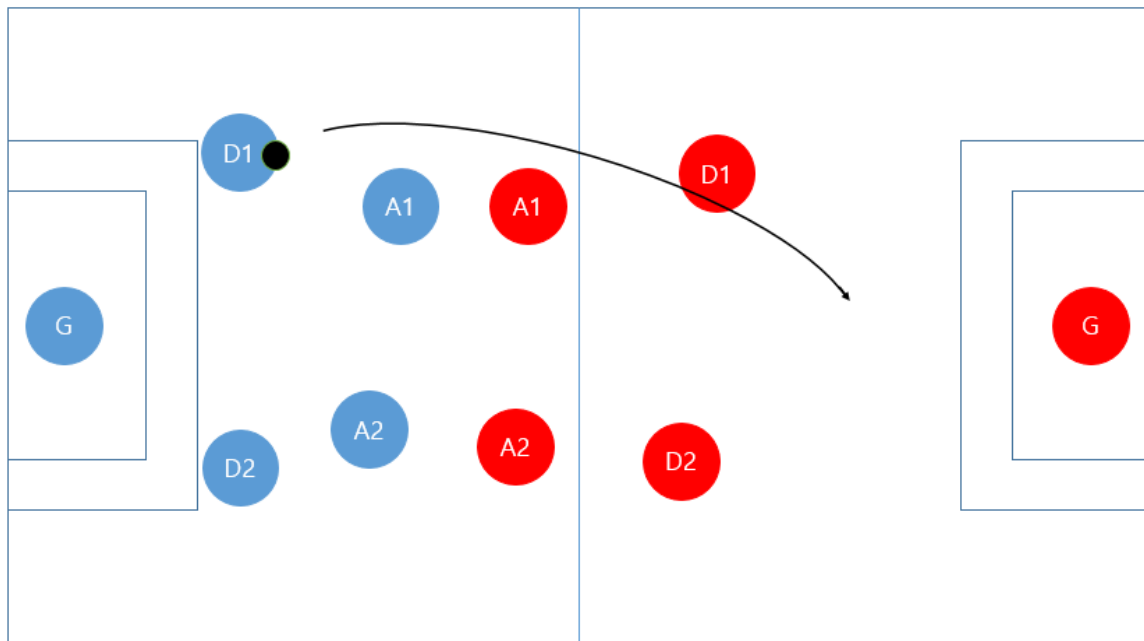
AgentPass(x, y) : x 가 y에게 패스하는 함수이다.

수비수가 공을 얻게 되는 경우

- i) 공격 영역이 아군 중앙선 안으로 들어왔을 경우에는 실점 기회가 증가하므로 슈팅하여 공을 멀리 보낸다.
- ii) 멀리있는 아군에게 패스한다. (그림 2.1.a)



(그림 2.1.a)



(그림 2.1.b) <공격진형이 아군 중앙선을 넘은 경우 공을 걷어낸다.>

2.2.1.3) 수비수 D1 진형 형성

```

IF posSelf.x ~= d1_location.x THEN                                =====> iii)
    AgentMove(self.AgentID, d1_location)

    IF posGK.y > 234 THEN                                           =====> iv)
        AgentMove(self.AgentID, d1_location)
    ENDIF

    IF posGK.y < 234 THEN
        AgentMove(self.AgentID, goal_Mid)
    ENDIF
ENDIF

```

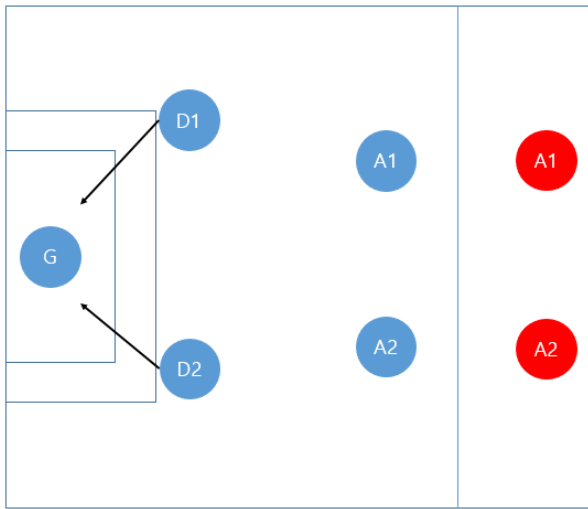
AgentMove(x, y) : x 가 y로 이동하는 함수이다. y는 방향벡터이다.

수비수 D1이 골대 영역 근처로 이동하기 위한 코드이다.

iii) 골키퍼의 x좌표가 다르면, 이동한다.

iv) 골키퍼가 중앙으로 움직이면, 그에 따른 빈공간으로 수비수는 이동한다.

정규화된, d1_location 혹은 goal_Mid로 골키퍼의 y위치에 따라 이동한다.



(그림.2.1.c)

2.2) D2 코드

2.2.2. function Defender:FrameMove() : D2

```
posSelf, getBallSelf = GetHomeAgentPos(self.AgentID)
posGK, getBallGK = GetHomeAgentPos(GOALKEEPER)
posA1, getBallA1 = GetHomeAgentPos(OFFENSER1)
posA2, getBallA2 = GetHomeAgentPos(OFFENSER2)

d2_location ← vector(d2_aim_x, d2_aim_y) - posSelf
goal_Mid ← vector(gk_mid_x, gk_mid_y) - posSelf

d2_location: norm()
goal_Mid: norm()

IF getBallSelf == true THEN
  IF posA1.x < ground_mid_x or posA2.x < ground_mid_x THEN =====> i)
    AgentShoot(DEFFENDER2)
  ENDIF

  IF posA1.x < posA2.x THEN                                     =====> ii)
    AgentPass(self.AgentID, OFFENSER2)
```

```

ELES
    AgentPass(self.AgentID, OFFENSER1)
ENDIF
ENDIF

IF posSelf.x ~= d2_location.x THEN                                =====> iii)
    AgentMove(self.AgentID, d2_location)

    IF posGK.y > 334 THEN                                            =====> iv)
        AgentMove(self.AgentID, d2_location)
    ENDIF

    IF posGK.y < 334 THEN
        AgentMove(self.AgentID, goal_Mid)
    ENDIF
ENDIF

```

수비수 D2의 행동코드이다.

수비수 D1과 거의 동일하며 특정위치 d2_location과 iv) 수비진형 부분의 코드가 조금 다르다.

기존의 코드는 동일하고 다른 코드만 서술하였다.

2.2.2.1) 수비수 D2 진형 형성

```

IF posSelf.x ~= d2_location.x THEN                                =====> iii)
    AgentMove(self.AgentID, d2_location)

    IF posGK.y > 334 THEN                                            =====> iv)
        AgentMove(self.AgentID, d2_location)
    ENDIF

    IF posGK.y < 334 THEN
        AgentMove(self.AgentID, goal_Mid)
    ENDIF
ENDIF

```

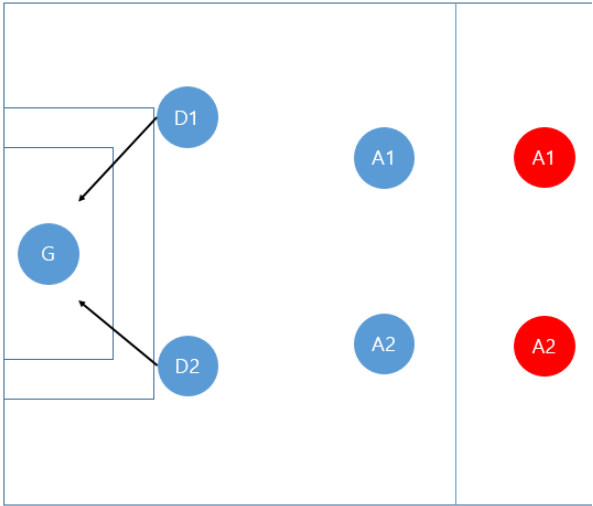
수비수 D2가 골대 영역 근처로 이동하기 위한 코드이다.

iii) 골키퍼의 x좌표가 다르면, 이동한다.

iv) 골키퍼가 중앙으로 움직이면, 그에 따른 빈공간으로 수비수는 이동한다.

AgentMove(x, y)

x 가 y 로 이동하는 함수이다. y 는 방향벡터이다.
정규화된, $d2_location$ 혹은 $goal_Mid$ 로 골키퍼의 y 위치에 따라 이동한다.



(그림.2.1.c)