

모바일 환경에서 실시간 시뮬레이션 축구게임을 위한 인공지능 에이전트 설계 및 구현

백경진*

*고려대학교 컴퓨터정보통신공학과

e-mail : ramyen@gmail.com

Design and Implementation of Artificial Intelligence Agent for Real-Time Simulation Football Game in a Mobile Environment

Kyeongjin Baek*

*Dept. of Computer & Communication Engineering, Korea University

요약

최근 모바일 게임에서의 인공지능과 관련된 연구가 활발히 진행되고 있다. 본 논문에서는 모바일 축구 시뮬레이션 게임에서 활용할 수 있는 인공지능 에이전트를 Hierarchical FSM 기반으로 설계하고 구현하여 실제 축구경기 결과와 비슷한 결과 도출하였다. 이러한 Hierarchical FSM 을 기반으로 한 지능형 에이전트는 코드의 재활용성이 높고 개념적으로 간단하여 인공지능 에이전트를 설계 및 구현하기에 적합하다.

1. 서론

게임에서 게임에서 수준 높은 인공지능은 유저들에게 깊은 호기심과 지속적인 도전 욕구를 불러 일으키며, 다양한 상황을 사실적으로 표현하여 보다 현실감 높은 몰입감을 제공하는 재미를 즐길 수 있게 한다. 스포츠 구기 종목과 같이 다수의 에이전트들이 다양한 조건에서 실시간으로 현실감있게 다양한 행동을 구현하고 제어하는 것은 매우 어려운 과제이다.

특히, 전 세계적으로 가장 인기 있는 축구 실시간 시뮬레이션 게임은 많은 자본과 오랜기간 누적된 개발 경험을 바탕으로 EA 사의 FIFA 시리즈, KONAMI 사의 Winning Eleven 과 SEGA 사의 Football Manager 시리즈가 비디오게임과 PC 게임 시장을 선도해왔으며, 최근 이러한 대규모 개발사들이 모바일 게임 시장까지 진출하고 있다.

국내에서는 10 년 전부터 몇몇 중소 업체들이 실시간 시뮬레이션 축구 게임에 도전하였지만 자금력과 기술력의 부족으로 제대로 구현하지 못하여, 텍스트 혹은 정해진 패턴을 시뮬레이션 하는 수준으로 대체하는 것이 대부분 이였고, 국내 학술적 연구도 활발히 진행되고 있지 않는 상태이다.

게다가, 최근 모바일 환경 중심으로 게임 산업이 변화하고 있는 상황에서 PC 하드웨어보다는 다소 성능이 부족한 모바일 디바이스 환경에서의 인공지능은 사실상 있는 에이전트 행동 패턴 구현뿐만 아니라 하드웨어적인 최적화 이슈까지 더해져 모바일 디바이스 환경에서 실행 가능한 적절한 인공지능 모델 제시가 필요하다.

이러한 문제점에 착안하여 본 논문에서는 필자가 개발한 ‘캡슐풋볼매니저’ [1]에서 사용한 인공지능을 기반으로 모바일 디바이스 환경에서 실시간 축구시뮬레이션하기 적합한 인공지능 에이전트 모델을 제시하고 구현하고자 한다.

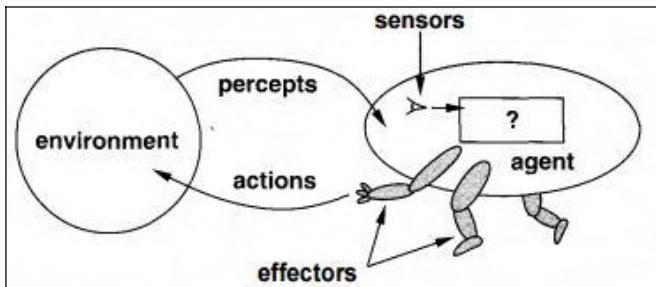
2. 관련 연구

축구게임에서 가장 핵심적인 기술은 인공지능 시뮬레이션과 물리 시뮬레이션이며, 본 장에서는 필자가 제시하는 인공지능 에이전트에서 사용된 주요 기반 기술들을 살펴본다.

2.1 지능형 에이전트

지능형 에이전트(Agent)[2]는 자신만의 목표(Goal)을 추구하면서 감각기관(Sensors)을 통해 주변환경을 인지(Percept)하고 작용기(Effectors)를 통해 스스로 반응(Actions) 하는 시스템을 말한다.[3]

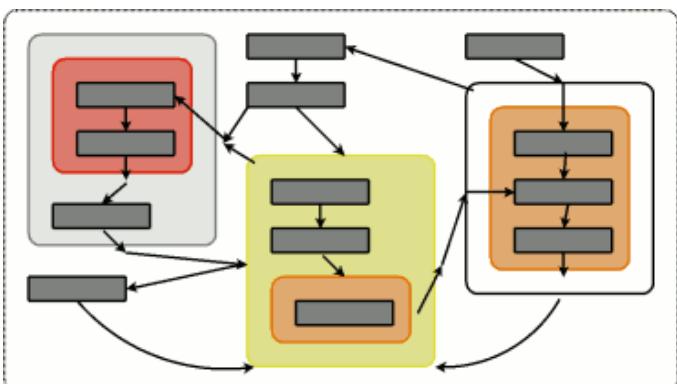
지능형 에이전트는 사람이나 다른 사물의 직접적인 간섭없이 스스로 판단하여 동작하는 자율성(autonomy)과, 다른 에이전트들과 상호작용 할 수 있는 사회성(Social ability)을 갖고 있고 단순히 환경에 반응하여 행동하는 것이 아니라 주도권을 가지는 능동성(proactivity)과 목표 지향적으로 행동하는 목표지향성(goal-orientedness)을 성질로 가지고 있다.[4] 이러한 개념은 축구게임에서 필수적으로 구현해야하는 선수(player)들을 표현 하기에 적합하다.



(그림 1) Agent interact with environment through sensors and effectors.

2.2 Hierarchical FSM

Hierarchical FSM[5]은 일련의 상태(state)가 여러 레벨에 걸쳐 결합된 계층구조를 가지고 다른 상태들을 포괄하는 상위상태로 모듈화 하여 기존 FSM[6]과는 다르게 코드 재사용이 용이하다. 특히, 새로운 상태를 추가할 경우 해당 그룹의 전이(transition)를 공유할 수 있어 중복적인 전이 로직을 피할 수 있다. 그리고 Hierarchical FSM은 상태간의 전이뿐만 아니라 상위상태간의 전이가 가능하도록 구현되어 디자인 과정이 비계층적인 FSM 보다 비교적 간단하다. [7] 이러한 특성 때문에 Hierarchical FSM 다양한 행동(Behavior)을 표현해야 되는 축구 시뮬레이션에서는 필수적인 기술이며, 인공지능 분야뿐 아니라 다양한 컴퓨팅 분야에서도 다양한 연구가 진행되고 있다.



(그림 2) A simple hierarchical state machine.

3. 시스템 설계

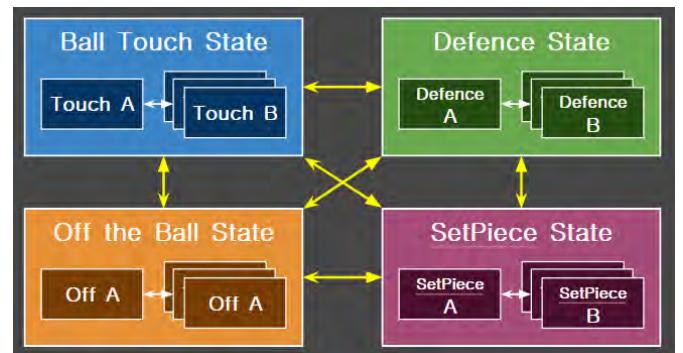
3.1 행동 구성

에이전트는 실제 축구에서 선수들이 행동(Behavior) 하는 상태(State)들을 다중 FSM으로 구성하고, 아래와 같이 볼터치/ 오프더볼/ 디펜스/ 세트피스 등으로 그룹화 시켜 상위상태로 구성한다. 그룹화된 상위상태는 볼 소유 유무, 경기 진행 상태들이 조건으로 등으로 다른 상위상태로 전이되며, 상위상태가 결정되면 계층적으로 구성된 FSM이 동작하도록 하였다.

<표 1> 행동 상태 리스트

| | | | |
|------|-----|-----|-----|
| 이동 | 숏패스 | 드리블 | 골킥 |
| 포지셔닝 | 롱패스 | 홀드 | 코너킥 |

| | | | |
|------|--------|----|------|
| 트래핑 | 쓰루패스 | 키핑 | 숏프리킥 |
| 태클 | 로빙쓰루패스 | 슛 | 롱프리킥 |
| 넘어짐 | 크로스 | 블럭 | 스로인 |
| 공격추격 | 수비추격 | 펑칭 | 캐칭 |

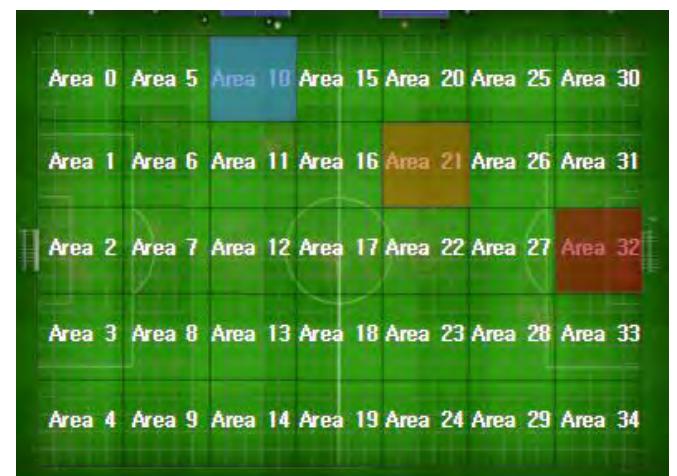


(그림 3) Hierarchical FSM 구성

3.2 필드 구성

축구게임에서 필드를 분할하여 에이전트의 행동패턴 혹은 로직에 영향을 미치게 하는 연구는 이미 오래전부터 시도해 왔던 방식 중에 하나이다.[8] 필자는 이러한 필드분할의 아이디어를 바탕으로 에이전트들이 경기를 진행 할 3D 필드(field)를 실제 정식 축구 경기장의 크기(가로:105m, 세로 75m)로 설정하고, 필드를 아래와 같이 Grid 와 Area 개념으로 세부 분할하였다. 이렇게 분할된 Area 는 에이전트가 수행할 수 있는 행동(Behavior)의 가중치에 로직적으로 영향을 미치게 하였다.

$$\begin{aligned} 3m * 3m &= 1\text{Grid} \\ 5\text{Grid} * 5\text{Grid} &= 1\text{Area} \\ 105m * 75m &= 35 * 25 \text{ Grid} = 7 * 5 \text{ Area} \end{aligned}$$



(그림 4) 필드 분할

<표 2> Area 별 행동 가중치

| 구분 | Area 10 | Area 22 | Area 33 |
|---------|---------|---------|---------|
| B-Pass | 1.0 | 0.8 | 0.2 |
| S-Pass | 2.0 | 1.5 | 0.7 |
| L-Pass | 1.5 | 1.6 | 0.0 |
| T-Pass | 1.0 | 1.1 | 0.6 |
| LT-Pass | 1.3 | 1.4 | 0.8 |
| Dribble | 1.0 | 1.1 | 1.5 |
| Crooss | 0.0 | 0.1 | 1.0 |
| Shoot | 0.0 | 0.1 | 2.0 |
| Mark | 1.0 | 1.1 | 0.0 |
| Press | 1.0 | 1.1 | 0.0 |
| Tackle | 0.5 | 0.6 | 0.1 |
| ... | ... | ... | ... |

4. 시스템 구현

4.1 개발환경

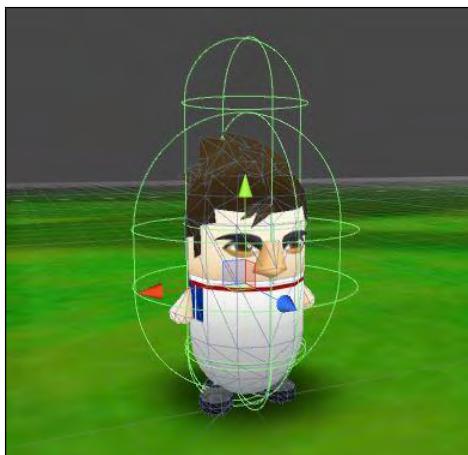
본 시스템의 구현을 위해서 아래의 표와 같은 개발 환경 및 도구를 사용하였다.

<표 3> 개발환경

| | |
|------|---------------------------------------|
| 개발환경 | Intel i5-3210M 2.5GHz CPU, 8GB Memory |
| IDE | Visual Studio 2010 |
| 개발엔진 | Unity3D 4.7.6 |
| 개발언어 | C# |

4.2 3D 에이전트

본 장에서 구현한 에이전트는 ‘캡슐풋볼매니저’에서 사용한 에이전트로 모바일 환경에서 무리 없이 랜더링할 수 있는 로우폴리곤으로 제작한 3D Mesh로 제작하였다. 그리고 볼(Ball) 충돌을 감지하는 충돌 트리거(Trigger)와 다른 에이전트와 겹치지 않도록 충돌을 체크하는 충돌체를 3D 캡슐로 구성하였고, 실제 선수와 신체사이즈가 비슷한 1.8m 정도로 제작하여 시뮬레이션에 활용하였다.



(그림 5) 3D Agent

3D 에이전트는 20 가지의 다양한 능력(Ability)을 가지게 하였는데 이러한 능력은 행동(Behavior)을 수행함에 있어서 해당 행동의 정확도 혹은 발생빈도에 영향을 미치게 하여 다양한 캐릭터 성을 갖게 할 수 있는 변수로 활용하였다.

그리고 모두 같은 능력의 수치를 갖게 하는 것이 아니라 일정 범위(30~99)사이에서 임의적으로 결정하여 시뮬레이션 중 다양한 상황이 발생되도록 하여 사용자들로 하여금 보는 즐거움을 줄 수 있도록 하였다.

| | | | | | | |
|-------|----|--|----|--------|----|----|
| 짧은 패스 | +5 | 77 | 속도 | 45 | 공격 | 62 |
| 긴 패스 | 68 | 신체 균형 | 58 | 수비 | 35 | |
| 드리블 | 79 | 지구력 | 56 | 창조력 | +3 | 58 |
| 헤더 | 58 | 챔프 | 58 | 시야 | +2 | 61 |
| 슈팅 | 71 | 부상 저항력 | 36 | 컨디션 유지 | 40 | |
| 컨트롤 | 23 | 골키퍼 능력 | 30 | 집중력 | 55 | |
| 프리킥 | 62 | 집중력 드리블 시 실수할 가능성과 넘어진 뒤 일어나는 속도에 관련 | 58 | | | |
| 감아차기 | 58 | | | | | |

(그림 6) 에이전트 능력(Ability) 리스트

4.3 행동 결정 프로세스

에이전트의 행동은 볼 터치 및 볼 소유 유무와 경기률에 따라 상위상태를 결정하고, 상위 상태에 등록된 하위상태의 행동에서 에이전트의 축구 포지션에 설정된 행동 RV(Random Value)와 에이전트의 현재 Area 의 행동 가중치를 조합하여 결정하였다.(패널티 박스 근처의 공격수(FW)는 같은 위치의 수비수(DF)보다 슛을 시도할 확률이 높으며, 중앙선 부근의 미드필더(MF)는 같은 위치의 수비수(DF)보다 고급패스인 로빙쓰루패스를 시도할 확률이 높다는 것을 의미한다.) 이렇게 모든 행동의 RV를 랜덤함수를 통하여 하나의 행동을 선택하여 에이전트가 실행하도록 하였다.

$$\text{행동 RV} = \text{포지션 행동 RV} * \text{Area 가중치}$$

```
// behavior RandomValue
CL_POSITION_DETAIL Detail = sc_Position_Manager.GetDetail(eType);
if (Detail == null)
    return;

if (_Player_Entity.Player_Info != null)
    nPosition_Level = _Player_Entity.Player_Info.GetPositionLevel(eType);

CL_BEHAVIOR_DETAIL_BEHAVIOR_DETAIL = Detail.GetBehaviorByLevel(nPosition_Level);
if (_BEHAVIOR_DETAIL == null)
    return;

foreach (KeyValuePair<E_BEHAVIOR_TYPE, int> pair in _BEHAVIOR_DETAIL.TrapTouchList)
    _TrapTouchList.Add(pair.Key, pair.Value);

foreach (KeyValuePair<E_BEHAVIOR_TYPE, int> pair in _BEHAVIOR_DETAIL.BallTouchList)
    _BallTouchList.Add(pair.Key, pair.Value);

foreach (KeyValuePair<E_BEHAVIOR_TYPE, int> pair in _BEHAVIOR_DETAIL.OffTheBallList)
    _OffTheBallList.Add(pair.Key, pair.Value);

foreach (KeyValuePair<E_BEHAVIOR_TYPE, int> pair in _BEHAVIOR_DETAIL.DefenseList)
    _DefenseList.Add(pair.Key, pair.Value);

foreach (KeyValuePair<E_BEHAVIOR_TYPE, int> pair in _BEHAVIOR_DETAIL.ActionList)
    _ActionList.Add(pair.Key, pair.Value);
```

(그림 7) Hierarchical FSM 관련 소스코드

4.4 시뮬레이션 테스트

필자가 제시한 인공지능 에이전트 시스템을 테스트하기 위하여 Unity3D Editor에 3D 경기장을 제작하여 시뮬레이션을 진행하였다. 골키퍼를 포함한 11 에이전트가 한 팀을 구성하여 A 팀과 B 팀으로 나눠서 경기를 진행하였으며, 모바일 게임이라는 것을 고려하여 경기 당 3 분으로 제한하였다.



(그림 8) Hierarchical FSM 관련 소스코드

5. 결론

본 논문에서는 모바일 환경에서의 실시간 시뮬레이션 축구게임에 적합한 에이전트를 구현하기 위해 지능형 에이전트와 Hierarchical FSM를 기반으로 한 인공지능 에이전트를 제시하였고 이를 바탕으로 구현하고 시뮬레이션 하였다. 이 시뮬레이션 테스트를 통해 필자가 제시한 인공지능 에이전트가 실제 축구선수와 유사한 플레이를 자주 시도하였고 팀의 최종 결과도 실제 축구경기결과와 흡사하게 도출되는 것을 확인할 수 있었다. 그러나 에이전트의 부적절한 행동선택이 종종 발견되어 개선의 여지를 남기게 되었다. 향후 연구에는 인공지능 에이전트의 행동 결정 프로세스를 필드의 Area 기반이 아닌 다른 방식으로의 접근이 필요해 보이며, 전반적인 인공지능 프레임워크를 설계하고 최적화하는 연구도 활발히 진행하고자 한다.

참고문헌

- [1] YJM Entertainment (2015), Capsule Football Manager
- [2] Stuart J. Russell and Peter Norvig (2003), Artificial Intelligence: A Modern Approach. Chapter 2.
- [3] 조영임, 최신 인공지능(1999), p293~323
- [4] https://ko.wikipedia.org/wiki/지능형_에이전트
- [5] Mihalis Yannakakis(2000), Hierarchical state machines
- [6] https://ko.wikipedia.org/wiki/유한_상태_기계
- [7] <http://aigamedev.com/open/article/hfsm-gist/>
- [8] Mat Buckland(2005), Programming Game AI by Example