



※ 본 아티클은 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

퓨어의 진짜 이점: 첨단형 레이싱 게임 AI

The Pure Advantage: Advanced Racing Game AI

에두아르도 지메네즈(Eduardo Jimenez)
가마수트라 등록일(2009. 2. 3)

http://www.gamasutra.com/view/feature/3920/the_pure_advantage_advanced_.php

[불공평해 보일만큼 강한 게임 AI를 적절히 조절하면서도, 경기를 치열하게 끌고 나가려면 어떻게 해야 하나? 블랙 락의 지메네즈가 critically acclaimed 퓨어에 쓰인 AI 기술을 심도있게 다루고자 한다.]

여는 말

이 기사는 레이싱 게임에서 AI 행동의 밸런스를 맞추는 데 사용하는 고무줄 메서드(*rubber band method*)에 대한 것이다. 기사에서는 이 개념을 연대순으로 살펴보고 새로 출시된 트릭 기반 레이싱 게임인 <퓨어>의 개발 기간 동안에 있었던 발전내용을 살펴볼 것이다.

처음에는 이 개념 이면의 3개의 주요 체계를 다뤄보고자 한다. 그 체계란 기술, 동적 경쟁 밸런싱, 레이스 스크립트이다. 그리고 이 체계들이 <퓨어>에서 어떻게 구현되었는지를 살펴보고, 앞에서 언급한 도구 세트를 어떻게 사용하여 플레이어가 원하는 게임 경험을 전달했는지를 살펴볼 것이다. 마지막으로 결론을 내리면서 이 체계의 또 다른 사용방법을 논할 것이다.

고무줄

<퓨어>의 AI를 개발할 때 항상 납득할 수 있고 공정하며 흥미있는 경주를 벌일 수 있는 체계를 만들고 싶었다. 일단 필드를 넓히고, 항상 경쟁 차량들을 플레이어 가까이 붙이되 플레이어의 실수에 대해서는 그리 큰 벌을 부과하지 않는 식으로 플레이어들을 자극하고 싶었다. 비디오 게임 속의 경주는 한편으로는 흥미를 유발하면서도 한편으로는 플레이어가 대열에서 너무 앞서거나 처지지 않게 난이도가 적절해야 한다.

고무줄 시스템은 플레이어 주변에 AI 캐릭터들을 항상 머무르게 함으로써 경주의 긴장을 유지한다. 그 구체적인 방식은 플레이어보다 앞서 나가는 차들의 속도나 회전 기술, 장애물 회피능력 등의 실력을 과감하게 줄이고, 플레이어보다 뒤의 차들의 실력은 과감하게 늘리는 것이다.

고무줄 메서드는 속도 변환에 주로 의지하고 있으며 AI 차량들의 움직임이 초인적으로 빨라지거나 바보같이 느려지는 게 확연히 보인다는 비판을 받고 있다.

이 메서드는 플레이어가 경주를 치르는 동안 항상 경주 차량과 함께 다니게 해주므로 매우 효과적이다. 그러나 큰 약점도 있는데, 지나치게 불공평한 점이 명백히 드러나는 점이다. 경기가 공정하게 치러진다는 인식을 확 깨버린다. 이 시스템에서는 플레이어가 경기의 초반 75%를 어떻게 치르는지는 신경도 안 쓰고, 오직 나머지 25%를 어떻게 하느냐에만 집중할 뿐이다. 잘 하다가 마지막에 조그마한 실수만 해도 경주 전체를 망쳐버리는 것이다.

반면 경기 초반에는 아무리 실수를 많이 해도 상관이 없고, 여전히 이길 기회가 있다. 그 결과 플레이어들은 실망하고 경기가 공정치 못하게 돌아간다고 느끼는 것이다. 이 때문에 우리는 고무줄 시스템을 쓰지 않았다.



기술

이 프로젝트를 처음 시작했을 때 우리는 AI의 성능의 기반을 기술 개념에 두고 싶었다. 모든 AI 캐릭터의 성능은 오직 독특한 기술 집합에 의해서만 결정된다. AI가 제대로 행동하는지 못 하는지는 관련된 다양한 능력에 따라 정해진다.

예를 들어 트릭 성능 기술은 캐릭터가 트릭을 얼마나 잘 구사하는지를 결정한다. 그리고 점프 효율성 기술에 따라 캐릭터의 점프 실패율도 정해진다.

이러한 기술 수준은 0, 1같은 숫자를 통해 나타난다. 0은 그 캐릭터가 관련 카테고리에서 최악이라는 뜻이고 1은 최고라는 뜻이다.

기술은 성능 이외에도 캐릭터의 특성을 나타낼 때도 쓸 수 있다. 예를 들어 캐릭터의 저돌성(플레이어를 트랙 밖으로 밀쳐내려는 성질) 또는 코너를 돌 때의 속도도 조절할 수 있다. 기술을 통해 AI 캐릭터의 성능을 눈에 띄게 바꿀 수는 없어도 행동을 바꿀 수는 있다. 이 기사에서는 성능에 영향을 주는 기술만을 살펴보겠다.

동적 경쟁 밸런싱

처음에는 AI의 기술 범위를 정함에 따라 게임 난이도를 결정할 수 있다고 생각했다. 예를 들어 0.4~0.6 사이의 범위가 정상 난이도에 적합하다고 생각했다. 0.6이면 기술이 최고인 캐릭터, 0.4이면 최악의 캐릭터이다. 이러한 접근방식은 여러 문제를 드러냈다.

- 보통 난이도 레벨은 너무 어렵거나 너무 쉽다. 따라서 플레이어의 기술과 맞추기가 매우 어렵고 언제나 게임의 난이도 설정이 엉망인 것처럼 느껴진다.
- *플레이어 혼자 경주를 하게 된다.* 이는 앞서 지적한 문제점의 부작용이다. 난이도 레벨이 잘못되었기 때문에 플레이어가 너무 오랫동안 혼자 달려야 한다. 당연히 재미가 없다.

이 방식은 너무 유연치 못하므로 우리는 더욱 공정하고도 도전의욕을 불러일으키는 경주를 구현하고자 노력하게 되었다. 따라서 우리는 이러한 가치를 '초기 기술'이라고 부르고, 초기 기술을 바탕으로 적용될 실제 기술에 대해 별도의 설정을 만들었다.

이러한 새로운 실제 기술(이하 스킬)은 초기 기술에 오프셋(포지티브 또는 네가티브)을 더해 계산한다. 이 오프셋은 경주의 상황에 따라 바뀐다. 우리는 이 경주 중에 기술을 동적으로 변화시키는 시스템을 '동적 경쟁 밸런싱'(이하 DCB)이라고 부른다.

ODCB가 고무줄 메서드의 변형에 불과하다고 할 사람도 있을 것이다. 고무줄이라는 이름에 있는 나쁜 이미지를 씻어버리기 위해 다른 이름을 사용했다. 내부적으로는 그 용어에 대한 편견을 갖지 않으려 했다.

처음에는 DCB값을 앞서나가는 캐릭터에 적용했다. 모든 AI 캐릭터는 이 인덱스에서 앞자리를 차지하는 것이 목표이다. 플레이어 0(인간 플레이어)이 가장 첫 번째 자리이고, 플레이어 1이 그 다음 자리인 식이다.

이 시스템은 원래 가지고 있던 문제를 어느 정도 완화했으나 해결은 하지 못했다. 여전히 반응이 늦고 제대로 움직이지 않았다. AI 캐릭터들이 결승선에 들어오는 순서는 무작위였다. 그리고 여전히 인간 플레이어가 대부분의 시간 동안 혼자 경주해야 한다는 것이 가장 큰 문제였다.

우리는 캐릭터들을 경기장 전체에 3그룹으로 나누어, DCB에 그룹이라는 개념을 부여하기로 했다. 이는 그룹 리더는 상위 성적으로 경주를 마치려고 하며 나머지 그룹 구성원들이 다음 그룹의 리더를 따돌리려고 애쓴다는 의미이다.

역시 제대로 움직여 주지 않는 지나치게 복잡한 DCB 시스템만 얻고 말았다. 그룹이 탄탄해지지 않았으며, 그룹 리더와 다른 자동차들이 무작위로 상위권을 차지하는 경우가 많았다.

이 개념을 사용했다면 그럭저럭 용납 가능한 결과를 얻어냈을지도 모른다. 그러나 우리는 훨씬 적합한 혁신적인 다른 개념을 사용했다. 그 개념에 대해서는 나중에 설명하기로 한다.



다시 기술로!

DCB를 구현할 때 기술 관련 문제들이 여러 가지 있었다.

그 중 첫 번째 문제는 기술이 $[0..1]$ 범위 밖일 때에 어떻게 하느냐였다. 우리는 적용될 기술을 $[0..1]$ 범위 내로만 정의했다. 그러나 우리는 DCB 변경자를 초기값에 적용한 후에도 초기 기술이 가끔씩 범위 밖으로 나간다는 것을 알게 되었다.

범위를 벗어나면 말이 안 되므로, 이 문제를 해결하기 위해 기술을 타당한 범위 내에 고정시켰다. AI의 성능이 -20% 나 110% 로 나오게 할 수는 없었다. 성능을 조정하려면 범위의 양끝을 재할당해 허용할 수 있고 원하는 만큼의 좋은 성능 또는 나쁜 성능을 낼 수 있도록만 하면 되었다.

우리는 AI의 밸런스를 맞추고 AI를 플레이어 가까이 계속 두는 데 생기는 문제를 AI 자체의 기술과 그 기술의 구현방식으로 풀어야 한다는 것을 알았다. 기술의 속성 확인은 일의 성격에 적합한 방식으로 해야 한다는 것을 알았다.

- **넓은 범위.** 우리는 AI 캐릭터의 행동을 결정하는 기술이 필요했다. 그래야 모든 기술에 다양한 변형을 만들 수 있기 때문이다. 0값은 AI의 행동이 매우 나쁘다는 것을 의미하고 1값은 AI의 행동이 매우 좋다는 것을 의미한다. 기술 범위가 충분히 넓지 않으면 플레이어의 다양한 기술 레벨에 맞춰줄 수가 없다.
- **반응성.** 기술 표현이 폭이 넓다고 해도 여러 큰 개별 단계에 배분되어 있다면 그걸로는 충분치 않다. 예를 들어 트릭 기술이 0.5 미만 값으로 배분된 AI 캐릭터는 트릭 시도 시 항상 실패하고, 0.5 이상 값으로 배분된 AI 캐릭터는 트릭 시도 시 항상 성공한다면 기술의 동적 범위가 충분하다고 볼 수 없다.

기술이 0인 캐릭터는 성능이 매우 안 좋고, 기술이 1인 캐릭터는 가급적 최고의 성능을 낸다. 그러나 캐릭터 기술이 0.3이면 이것이 0.4나 0.45로 업그레이드된다거나 0.2나 0.1로 떨어진다고 해도 현격한 기술 변화가 나타나지

않는다. 수적 변화에 따라 기술이 바로 변하는 것이 아니므로 기술이 쓸모가 없다.

대부분의 변화는 0.2 미만이므로 우리는 가급적 연속적이며 변화를 잘 나타내는 기술 표현을 원하게 되었다. 단계가 개별적이라면 단계를 많이 만들면 결국 공평하게 분배될 것이다.

발전은 또한 선형이어야 한다. 행동의 90% 변화가 값의 10% 변화와 연관되어 있다면 그리 유용하지 못하다. 행동 변화는 가급적 값 변화와 맞아야 한다.

경주 스크립트

우리의 간절한 소망에도 불구하고 AI는 여전히 차세대 AAA급 레이싱 게임에서 기대하는 것만큼 잘 움직여주지 않았다. 어렵지도 않고 재미있지도 않았다. 그룹이 잘 뭉쳐다니지도 않았고, 플레이어는 여전히 혼자서 경주를 벌여야 했다.

정의

이 문제를 해결하기 위해 아이디어 회의를 했다. 그 결과 경주 스크립트 개념을 생각해냈다. 디자이너들이 원하는 방향으로 경주를 진행시키도록 스크립트를 짜는 것이다. 스크립트는 일어날 수 있는 모든 상황을 염두에 두고 작성되어야 한다. 예를 들어 플레이어의 기술이 가장 뛰어난 AI보다도 우수하다거나, 가장 실력이 떨어지는 AI보다도 낮은 경우, 그 둘 사이의 모든 경우를 다 감안해야 한다.

경주가 항상 이상적인 스크립트대로 되는 것은 아니다. 경주는 플레이어의 실력에 따라 좌우된다. 스크립트 덕분에 디자이너들은 게임이 플레이어의 실력에 따라 어떻게 움직여야 할지를 다시금 생각하게 되었다. 여기서 이야기하는 스크립트는 지침일 뿐 반드시 지켜야 하는 엄격한 선형 스크립트는 아니다.

이러한 스크립트는 어떤 모양을 갖춰야 하는가? 그 사례는 다음과 같다.

- 플레이어의 기술이 가장 뛰어난 AI 캐릭터와 비슷할 경우 경주가 시작된 지 얼마 되지 않아 플레이어는 AI 경쟁 차량들로 이루어진 후위 그룹의 선두에 도달할 것이다. 그리고 그는 경주 초반 70~80%에 걸쳐 계속 앞서 나갈 것이다. 그가 다음 그룹에 들어오면 일부 AI 캐릭터들이 그를 따라잡으려고 하여, 그가 혼자서 경주를 하지 않도록 한다. 경주 후반 20~30% 부분에서 그는 1위를 놓고 경쟁하게 될 것이다.

선두 그룹을 달리는 AI들은 특히 경주 최후의 순간, 플레이어의 실수에 관대하게 대처함으로써 인간 플레이어가 선두를 조금 수월하게 차지할 수 있도록 해 준다. 이렇게 하면 경기 처음부터 하던 페이스로 달리는 것 같은 느낌이 들고 마지막 부분에서 실수를 저지르더라도 성적에 악영향이 가지 않는다. 이것이 경주의 주요 스크립트이다.

경주 스크립트가 없을 경우 일반적인 경주 시나리오는 어떻게 돌아가는가?

- 플레이어의 기술이 AI보다 뛰어날 경우 플레이어는 경주가 시작된 직후부터 선두를 차지하며 모든 랩에서 압도적인 차이로 AI를 따돌린다.

- 플레이어의 기술이 AI보다 못할 경우 플레이어는 경주가 시작된 직후부터 뒤처지기 시작하여 경주 내내 하위 그룹에 머물러 있고, 랩이 반복될 때마다 앞서가는 AI와의 거리가 벌어진다.
- 플레이어의 기술이 최하위 AI보다는 높지만 최상위 AI보다는 낮을 경우(한마디로 최고 기술 수준은 아닐 경우) 그는 경주 내내 중위 그룹, 또는 하위 그룹의 AI 캐릭터들과 사투를 벌여야 한다.

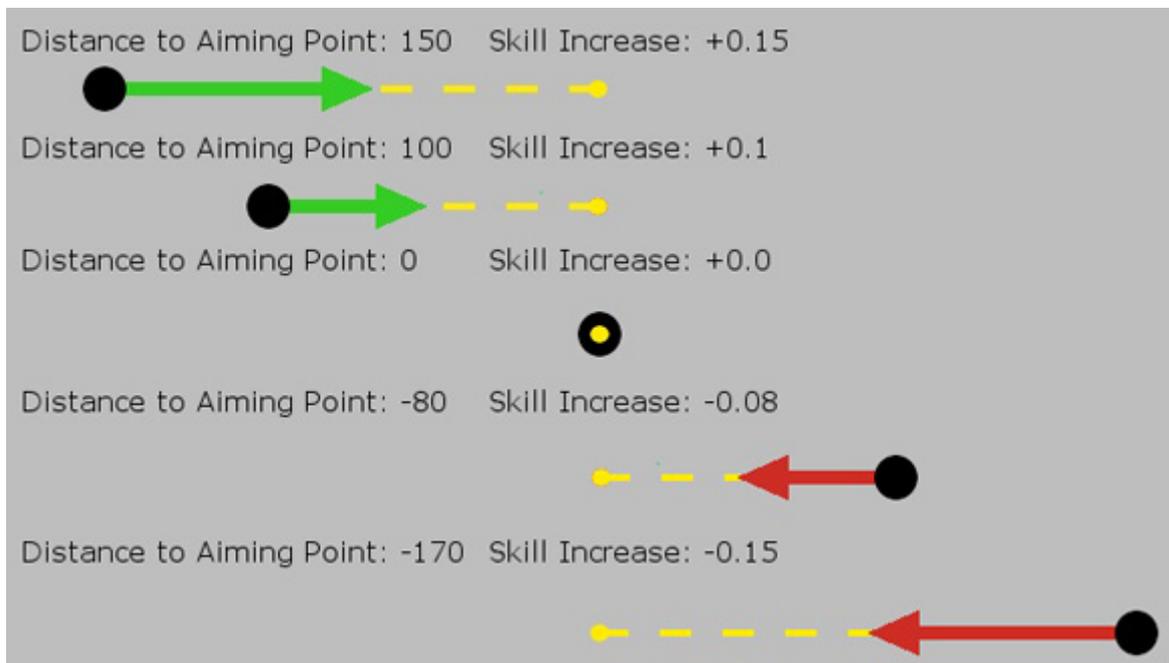
경주 스크립트를 사용할 경우 프로그래머는 표준 사례에 중점을 둘 수 있다. 앞의 사례의 경우 이는 가장 먼저 짚고 넘어가야 할 점이 될 것이다.

구현

그리고 나서 스크립트를 구현할 방법을 찾아야 했다. 우리는 잘 들어맞는 메서드를 찾아냈다. 모든 AI 캐릭터에게 인간 플레이어와의 사이에 뒤야 할 거리를 미터 단위로 부여하는 것이다. 양수일 경우 플레이어보다 앞서가는 것, 음수일 경우 플레이어를 뒤따라가는 것을 의미한다.

그리고 나서 게임 속에서 플레이와의 관계와는 상관 없이 사전에 할당된 거리에 따라 AI 캐릭터의 기술이 동적으로 변하도록 설정하는 것이다. 물론 이는 플레이어의 위치와는 직접적인 연관이 없다.

따라서 AI가 인간 플레이어보다 앞서는 경우라도 이러한 목표를 달성하려면 AI 캐릭터의 기술력이 바뀔 수 있다. 반대의 경우도 마찬가지이다. 즉 AI 캐릭터가 이미 인간 플레이어보다 뒤처지는 경우라도 사전에 할당된 거리를 맞추기 위해 기술을 더 낮추는 것이다.



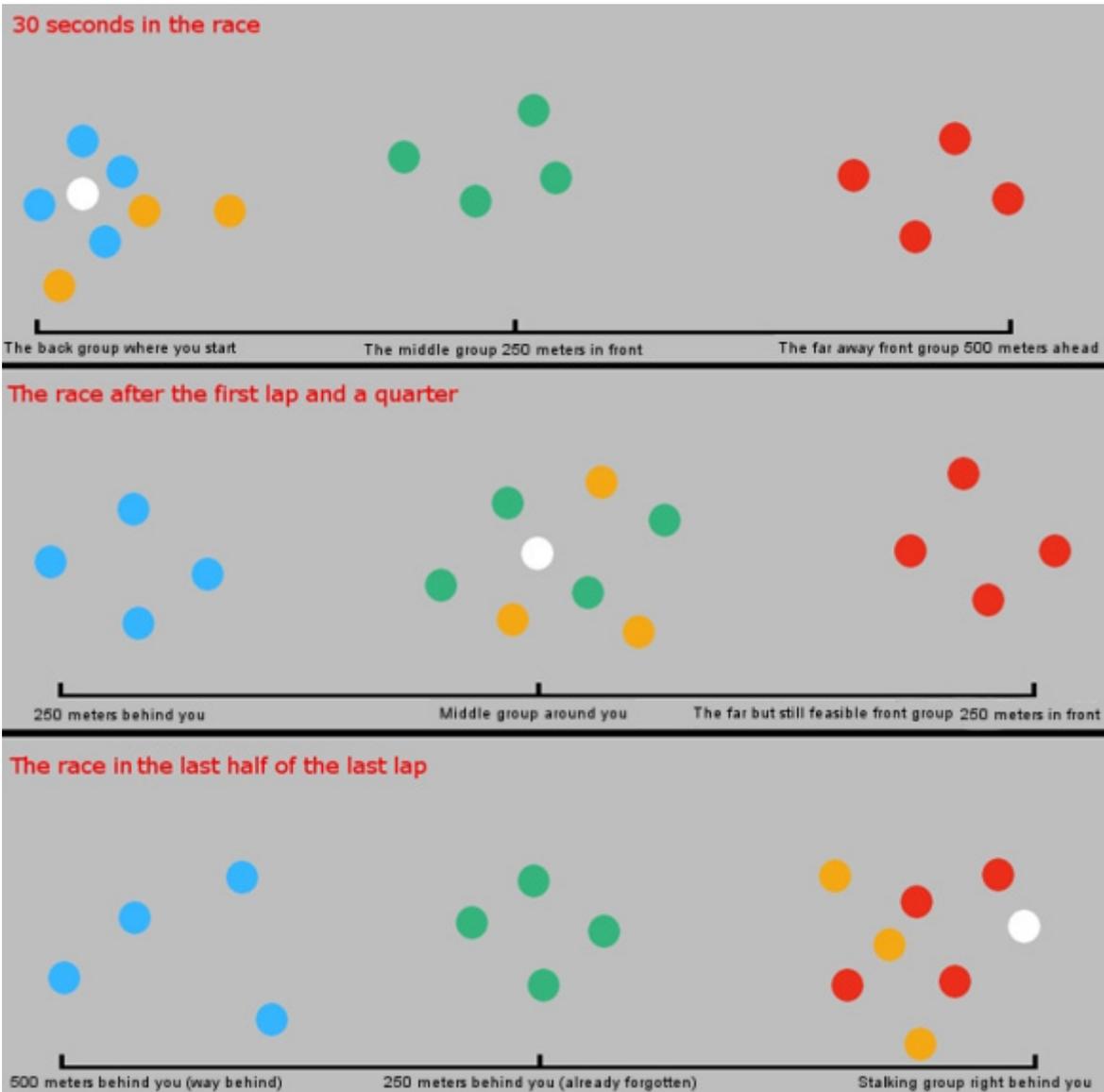
앞의 그림은 목표 지점까지의 거리에 따라 기술이 어떻게 바뀌는지를 보여준다. 기술 향상은 목표 지점까지의 거리에 비례한다. 양적, 음적 기술 향상이 제한된 후의 최대 거리(표에는 150m)가 나와 있다.

우리는 인간 플레이어와의 상대적 거리를 따지고 있으므로 경주 내내 이 거리를 동적으로 줄일 수 있다는 것도 알게 되었다. 이 핵심 개념 덕분에 우리는 AI의 행동을 더 잘 제어할 수 있게 되었다.

또한 우리가 작성한 이상적인 스크립트를 재현하기 위해 플레이어와의 상대적 거리를 따지고 있다. 다음 사례를 사용하면 이를 가장 잘 묘사할 수 있을 것이다.

- 우리는 AI 캐릭터를 선두 그룹, 중위 그룹, 하위 그룹 3그룹으로 나누었다. 이 그룹은 AI 캐릭터를 4개씩 가지고 있다. 따라서 총 12개의 AI 캐릭터가 있는 것이다. 여기에 인간 플레이어가 들어가고, 어느 그룹에도 속하지 않는 캐릭터가 3개 있다.
- 남은 3개의 캐릭터는 소위 ‘근처 그룹’에 배당된다. 이 그룹은 언제나 인간 플레이어 곁에 다니며(인간 플레이어와의 목표간격 0m). 플레이어가 그룹을 옮겨 다닐 때 마다 따라간다. 이 그룹 때문에 플레이어의 기술이 AI보다 현저히 뛰어나거나 떨어지지 않는 한 언제나 일부 AI 캐릭터들이 플레이어 곁에 있게 된다.
- ‘하위 그룹’의 목표는 경주 시작 시에는 인간 플레이어와의 거리가 +0m, 그리고 경주 종료 시에는 -500m를 유지하는 것이다. 이들 그룹은 처음에는 플레이어 주변에 있으나 경주 시작 후 쉽게 따돌릴 수 있다.
- ‘중간 그룹’의 목표는 경주 시작 시 플레이어의 전방 250m(+250m), 그리고 경주 종료 시에는 플레이어의 후방 250m(-250m)에 위치하는 것이 목표이다. 이 그룹은 처음에는 플레이어를 추월하고, 앞서나가지만 점점 플레이어에게 재추월당한다. 플레이어의 관점에서는 서서히 위치를 되찾아가는 것이며 기술을 통해 중간 그룹을 따돌리는 것이다. 이 그룹이 플레이어가 추월하게끔 길을 비켜주는 것처럼 느껴지지는 않는데, 그것은 이들이 뒤처지는 과정이 매우 천천히 일어나므로 플레이어의 힘으로 이들을 따돌리는 것처럼 느껴지기 않기 때문이다.
- 마지막으로 ‘선두 그룹’의 목표는 경주 시작 시에는 플레이어의 전방 500m(+500m), 경주 종료 시에는 플레이어와 함께 결승선을 끊는 것이다(+0m). 따라서 경주 시작 시에는 이 그룹은 플레이어를 쉽게 따돌리게 되지만 플레이어가 중간 그룹을 따돌리고 나면 선두 그룹도 쉽게 따라잡을 수 있다.

경주가 끝나갈 무렵 플레이어는 결국 선두 그룹을 따라잡고 선두 그룹에서 선두 자리를 차지하게 된다. 플레이어들은 처음 시작할 때 저 멀리 사라져갔던 선두 그룹을 따라잡아 합류하고, 결국에는 그들을 따돌려서 그간의 노고를 보상받는 기분을 느낀다.



앞의 이미지는 경주가 진행되는 전형적인 상황을 보여주고 있다. 흰 점은 플레이어이고 파란 점은 하위 그룹, 녹색 점은 중위 그룹, 빨간 점은 선두 그룹. 주황색 점은 근처 그룹이다.

그 외에도 이 스크립트를 적용하는데 여러 미세한 메커니즘이 필요하지만, 가장 중요한 것은 위에 설명한 것이라고 할 수 있다. 모든 AI 캐릭터는 인간 플레이어의 위치에 따라 다른 게임 진행을 보이게 설정되지만 기술 제한을 받는다. 이들이 초기 기술을 늘이거나 줄이는 것은 원래 정해놓은 것을 초과하거나 미달하는 성과를 보여, 원래 있어야 할 자리에 가지 못할 때뿐이다.

AI가 원하는 만큼 그 능력을 스스로 얼마든지 늘이거나 줄이지 못하게 하고, 대신 난이도에 맞게 정해놓은 레벨 안에서만 능력을 조절할 수 있게 한 것이 중요하다. 이렇게 해야 게임이 공정하게 돌아가고 있다는 느낌을 줄 수 있다.

우리는 플레이어들이 게임에서 노력해야 선두 자리를 얻는다는 느낌을 받기를 바란다. 경주를 잘 하지 못한다면 경주 종료 시 선두 그룹에 들지 못하는 식으로, 경주를 잘 하면 뒤에 따라오는 차와의 거리를 늘려나가는 식으로 상벌이 주어질 것이다.

<퓨어>에 사용된 추가 메커니즘

이전에도 언급했듯이 AI의 성능이 예상한 만큼 나오게 하기 위해 다음과 같은 몇 가지 추가 메커니즘을 개발했다.

- 경주의 최초 순간은 AI가 플레이어의 진행을 막는 장애물 구실을 하여 플레이어를 압도하는 중요한 순간이다. 그룹 구조를 제대로 만들어 플레이어가 쉽게 전진할 수 있게 만들기는 역시 힘들다.

따라서 경주 최초의 순간(경주 개시 10~20초 동안)에는 AI는 모든 카테고리에서 최고의 성능을 내야 한다. 그 이후 AI는 최종 기술을 또한 몇 초간(5초 가량) 더 보정한다. 이 덕분에 그룹을 만들 수 있다. 물론 이는 AI의 기술이 이미 1값에 가까워진 고난이도 레벨에서는 잘 작동하지 않는다.

- 게이머들이 경주 초기 단계를 쉽다고 생각하는 것을 알아냈다. 모두가 다 그렇게 느끼는 건 아니지만, 일부 사람들은 다른 사람들보다 그렇게 느끼는 경향이 강하다. 그래서 최종 랩에서 0이 될 때까지 각 랩의 AI 기술을 줄여주는 변경자를 추가했다. 예를 들어 3개의 랩이 있는 경주에서 첫 번째 랩은 기술이 원래보다 0.05가 더 높다. 그리고 두 번째 랩에서는 0.025가 더 높다. 마지막 랩에서는 기술이 원래대로이다.
- 스크립트에서는 플레이어가 경주의 마지막 20~25%에서 경쟁 차량들을 물리쳐야 한다고 나와있다. 선두 그룹의 목표가 경주 종료 시 플레이어와의 거리를 0m로 만드는 것이라면 플레이어들은 문자 그대로 최후의 몇 m를 남겨두고 다른 차량들과 경쟁을 벌이게 될 것이다.

그래서 우리는 경주의 초반 80% 부분에만 이동 표적 포인트 메커니즘을 적용하기로 했다. 따라서 인간 플레이어는 마지막 20~25% 거리에서는 선두 그룹에 둘러싸이게 되며 첫 번째 위치로 가기가 쉬워질 것이다. 우리는 현재 말하고 있는 경주의 퍼센티지는 총 경주 거리의 퍼센티지임에 주의하라.

- 이전의 수단을 써도 플레이어가 선두를 차지하기는 어려울 수 있으며 그것은 쉽게 용납할만한 상황이 아니다. 경주 종료 직전에 실수를 저지른다면 돌이킬 수 없는 결과를 초래할 것이다. 그래서 경주의 최후 부분에는 AI 캐릭터의 초기 기술 증진이 일어나지 않도록 했다.

우리는 경주의 마지막 20% 구간에서 최대 기술 변경자를 감소시켜 결국에는 0이 되도록 했다. 따라서 AI는 실수를 회복하기가 쉽지 않게 되어 인간 플레이어가 비집고 들어갈 틈을 만들게 된다. 만약 인간 플레이어가 뒤에 처질 경우에도 이들은 기술을 줄이게 된다. 따라서 플레이어는 막판에 실수를 했다 하더라도 약간의 여유를 얻게 되므로 손해는 최소에 그친다.

결론

여기 나온 알고리즘은 정적이고 사전 정의된 방식으로 경주를 설정하고 작성하는 방식을 제시한다. 그러나 또한 이 방식은 인간 플레이어의 움직임에 맞춰 경주 내용을 역동적으로, 그러나 한계 내에서 눈에 띄지 않게 변화시킬 수 있다. 따라서 플레이어는 게임이 불공정하게 진행된다는 느낌을 받지 않는다.

플레이어가 여기 소개된, AI 행동을 제어하는 <퓨어>의 시스템을 눈치채지 않고, 노력에 의해 조금씩 경주에서 앞자리를 차지해나가 노고를 보상받는 느낌을 받기를 바란다.

이 메서드는 우리 아케이드 레이싱 게임에 아주 잘 어울린다고 생각한다. 또한 플레이어들이 상대 차량들로 가득한 흥미로운 경주를 즐기며 쉽게 앞자리로 나아가 상대 차량을 따돌리는 경험을 만끽하길 바란다.

플레이어들이 맨 앞자리를 차지할 수 있기를 바라지만 그러면서도 긴장과 흥미를 유지하려면 또한 경주 막판에 상대 차량과 경쟁해야 한다. 그래서 우리는 이 메서드가 이 헐리우드 스타일의 목표에 가장 잘 들어맞는다고 생각했다.

또한 여기 소개된 알고리즘이 다른 종류의 레이싱 게임이나 시뮬레이션에도 사용될 수 있으며 구현하기 쉬울 것이라고 생각한다.



이 알고리즘은 FPS, RTS, RPG 등 다른 장르 게임에도 쓰일 수 있을 것이다. 예를 들어 슈팅 게임에서 플레이어가 체력을 자동 보충하지 않고 특정한 전략 지점까지 일정 수준의 체력을 유지한 채로 이동해야 하는 때 같은 경우이다.

만약 플레이어의 체력이 일정 수준 이상이라면 시적들의 공격은 더욱 강해질 것이지만, 체력이 일정 수준 이하라면 적들은 플레이어가 공격을 피해 체력을 회복할 수 있는 빈틈을 마련해 줄 것이다.

그리고 여기에 몇 가지 한계를 만든다. 예를 들어 적들이 결코 전투를 멈추지 않는다면, 헤드샷 성공률이 100%가 될 수 없게 한다면 하는 것이다. 그리고 난이도 레벨에 따라 이런 한계를 설정한다.

쉬운 난이도에서는 적의 압박이 그리 강하지 않아 인간 플레이어는 체력 참조값보다 높은 체력을 가지고 문제없이 전략 지점까지 갈 수 있을 것이다. 그러나 고난이도에서는 정반대일 것이다.

간단히 말해 이 방법론에서는 AI 캐릭터의 행동을 결정하는 일련의 도구와 값(기술)을 제시한다. 그리고 이러한 도구를 사용하는 플레이어의 경험을 사전 정의된 스크립트에 맞추고, 기술의 값을 플레이어의 실력에 맞추어 동적으로 변화시키는 것이다.

감사의 말

나와 함께 <퓨어>의 AI를 만들고 이 기사 속의 여러 아이디어를 제공한 이들을 칭찬하고 싶다. 이는 함께 일하고 아이디어를 계속 나누어 훌륭한 AI를 완성해낸 이들에 대한 감사 표시이다.

여기 나온 기사는 초기 아이디어에 기초한 1년간의 고되고 반복적인 노력과, 수많은 테스트를 통해 얻어낸 기존 시스템의 발전과 진화의 산물이다. 포커스 테스트는 특히 중요했으며 AI 이면의 문제가 무엇인지를 알게 해 주었다.

제이슨 아벤트, 마이클 벤필드, 짐 칼린, 이안 길피더, 톰 윌리엄스(순서는 무순)는 물론 개발기간 내내 블라인드 테스트를 도와주셨던 모든 분들에게 감사를 전하고 싶다. 이들이 없었다면 우리는 현재와 같은 품질을 낼 수 없었을 것이다.