



## 기획자 노트-머시네이션즈, 게임 메카닉 설계의 새로운 방법 (The Designer's Notebook: Machinations, A New Way to Design Game Mechanics)

작성자: 어니스트 아담스(Ernest Adams), 요리스 도르만스(Joris Dormans)  
작성일: 2012년 8월 16일

게임 메카닉은 게임 기획 가운데 가장 이야기하기 어려운 측면이다. 사용하는 용어는 대부분 추상적이다. - 네거티브 피드백 루프(Negative feedback loop)라거나 난이도 진행(difficulty progression) 같은 식으로. 또 프로토타입을 만들거나 테스트하기도 성가시다. 보드게임처럼 단순한 게임이 아닌 한 직접 코드를 작성하거나 스프레드시트를 이용해야 하는데, 두 가지 모두 그닥 빠르지도 직관적이지도 않기 때문이다.

최근에 나는 이런 일을 더 잘하는 방법이 (일부) 담긴 책을 공동집필했다. 책 제목은 <게임 메카닉: 고급 게임 디자인(Game Mechanics: Advanced Game Design)> 으로 Peachpit 에서 출판되었고 [여기](#)<sup>1</sup>에서 살 수 있다. ([교사](#)<sup>2</sup>나 [미디어 리뷰](#)<sup>3</sup>용 증정본도 신청할 수 있다.) 이번달 컬럼은 이 책의 챕터 5를 요약한 것이다. 공동저자인 요리스 도르만스는 네덜란드의 게임디자이너이자 학자이다.

요리스는 머시네이션즈(Machinations) 라는 무료 툴을 제작했는데. 나는 이 툴이 우리가 게임메카닉을 개발하고 가르치는 방법에 혁신을 가져다줄거라고 생각한다.

---

<sup>1</sup> 참조링크: <http://www.peachpit.com/gamemechanics>

<sup>2</sup> 참조링크:

[http://www.pearsonhighered.com/pearsonhigheredus/educator/exam\\_copy/bookBag.page?isbn=978-0-321-82027-3](http://www.pearsonhighered.com/pearsonhigheredus/educator/exam_copy/bookBag.page?isbn=978-0-321-82027-3)

<sup>3</sup>참조링크: [http://www.peachpit.com/store/media\\_review.aspx](http://www.peachpit.com/store/media_review.aspx)

(나는 과대광고를 정말 싫어하기 때문에<sup>4</sup> 이걸 가볍게 이야기하는 게 아니다.)  
지금부터, 머시네이션즈 이 무엇이고 이것이 하는일이 무엇인지 소개할 것이다.

머시네이션즈는 게임 경제(game economy) 도해를 위한 시각적인 언어이자, 그리기 위한 도구이고 무엇보다도 코드를 작성하지 않고 그것들을 시뮬레이션하는 도구이다.

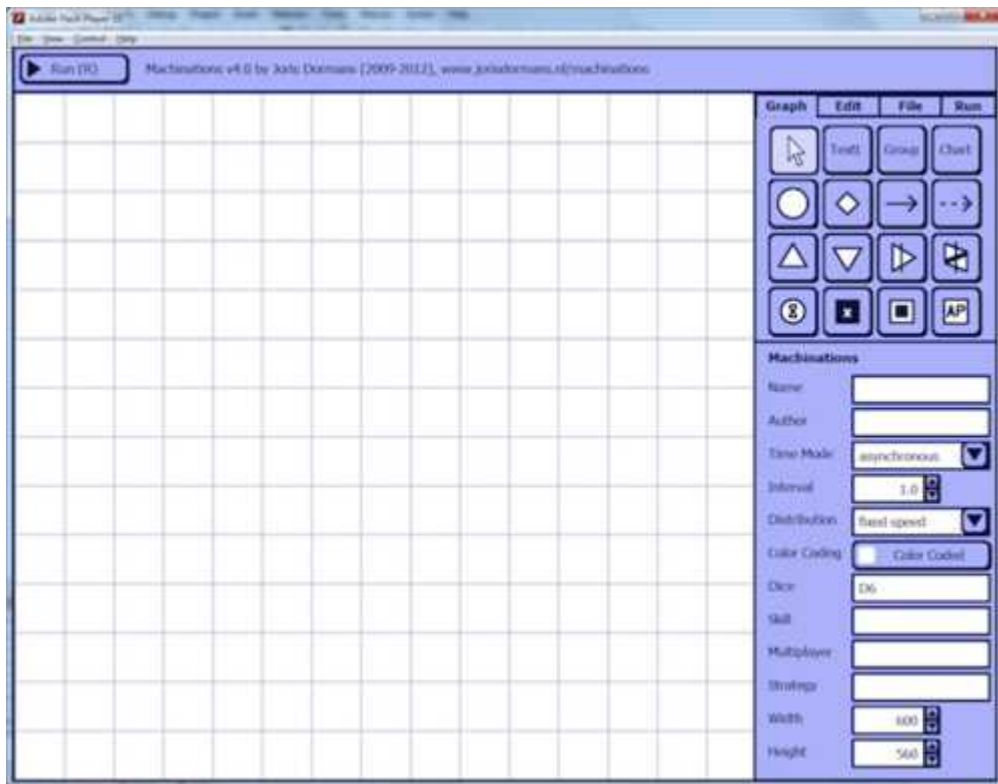
요리스는 내 책 <Fundamentals of Game Design>을 읽고 내가 이야기했던 게임 내부 경제(internal economy)의 기본요소들(자원, 소스, 드레인, 컨버터, 트레이더)을 도식화할 방법을 만들어냈다. 그리고 만들어진 시스템을 더 강력하고 유연하게 하는 기능을 몇 가지 추가했다.

## 머시네이션즈 툴은 어떻게 동작하는가

머시네이션즈 툴은 머시네이션즈 다이어그램의 디지털 버전을 만들고 저장할 수 있고, 그것들이 시간이 흐름에 따라 어떻게 변하는지 확인할 수 있게 해준다. 이 도구는 마이크로소프트 비지오(Visio) 같은 객체지향 2D 드로잉 프로그램처럼 생겼다. 워크스페이스가 가운데 위치하고 있고 사이드패널에 다양한 도구들이 있다.

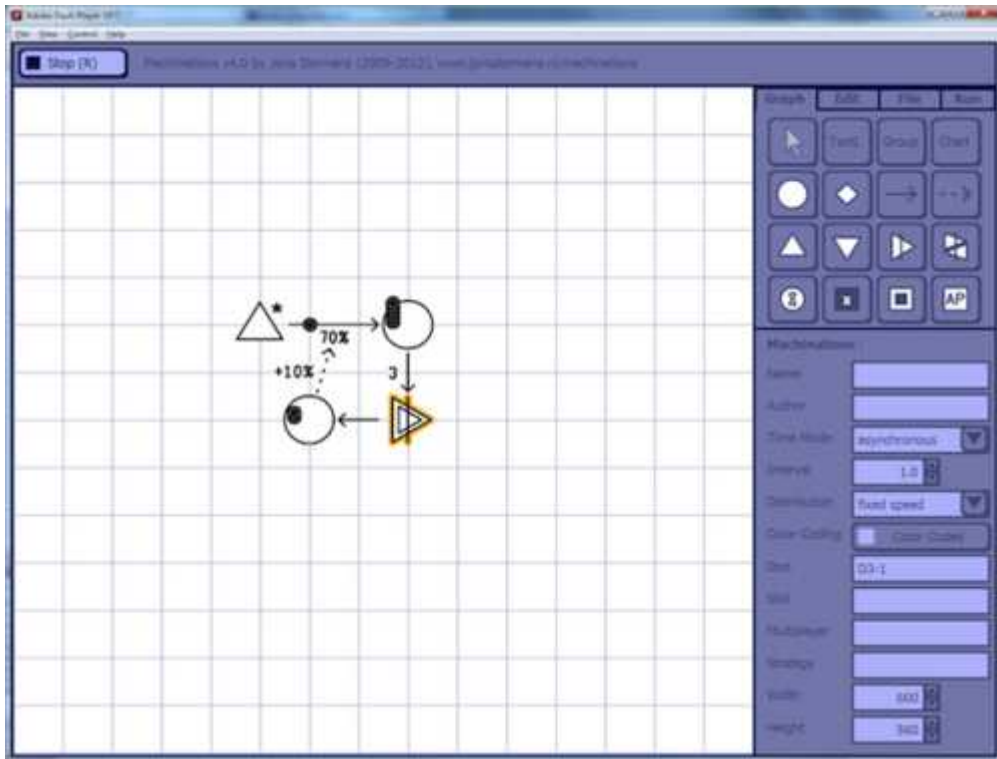
---

<sup>4</sup> 참조링크: [http://gamasutra.com/view/feature/3325/the\\_designers\\_notebook\\_shut\\_up\\_.php](http://gamasutra.com/view/feature/3325/the_designers_notebook_shut_up_.php)



머시네이션즈 툴, [www.jorisdormans.nl/machinations](http://www.jorisdormans.nl/machinations)에서 받을 수 있다.

머시네이션즈의 실행버튼을 누르면, 다이어그램에 정의된 이벤트를 일련의 타임스텝(time step)이나 이터레이션(iteration)만큼(이 두 용어는 서로 바꿔 쓸 수 있다) 수행하게 된다. 그리고 다이어그램의 상태가 바뀐다. 한 번의 이터레이션을 완료하면, 다이어그램의 새로운 상태를 가지고 또 다른 이터레이션을 수행하고, 멈추라고 하기 전까지 계속해서 수행할 것이다. 각 타임스텝의 길이는 인터벌(interval) 값을 수정해서 조정할 수 있다. 실행 속도를 느리게 하려면, 타임스텝의 인터벌을 몇 초 정도로 세팅하면 된다.



간단한 머시네이션즈 다이어그램이 실행되는 모습.

작은 원은 자원이고, 어느 한 노드에 저장되거나 노드 사이를 옮겨다닐 수 있다.

머시네이션즈 다이어그램을 이용하면 원하는 만큼 많이 혹은 적게 추상화할 수 있다. 게임 메카닉의 전체나 특정 부분에만 집중해서 사용할 수도 있다. 머시네이션즈 다이어그램을 사용하여 게임의 메카닉을 서로 다른 수준의 디테일로 디자인하고 테스트할 수 있다. 예를 들면, 게임이 실제로는 멀티플레이로 진행되더라도, 싱글플레이 관점에서 게임의 모델을 만들어 보면 충분할 때가 자주 있다. 또는 게임 메카닉을 특정 한 유저에 대해서만 디테일하게 모델링해보는 것이(여러 플레이어에 대해서 해보는 것보다) 유용할 수 있다. 아니면, 게임의 특정 측면(플레이어가 돌아가면서 턴을 돌리는)을 덜어낼 수도 있다.

## 머시네이션즈 다이어그램의 기본 요소들

머시네이션즈 프레임워크는 게임 내부 경제요소들끼리의 활동, 상호작용, 의사소통을 모델링하기 위해 디자인되었다. 게임의 경제시스템은 자원의 흐름에 의해 주도된다. 게임의 내부경제를 모델링하기 위해서 머시네이션즈 다이어그램은

자원(resource)을 끌어오고(pull), 보내고(push), 모으고(gather), 분배(distribute)하는 몇몇 노드들을 사용한다. 자원의 연결(Resource connection)은 자원이 요소들 사이를 어떻게 이동하는지 결정하고, 상태의 연결(State connection)은 현재의 자원 분포가 어떻게 다이어그램상의 다른 요소들을 변경할지를 결정한다. 이러한 요소들은 머시네이션즈 다이어그램의 핵심이 된다. 가장 단순한 것부터 시작해보자.

### 자원과 리소스풀의 연결

리소스풀(pool)은 다이어그램에서 자원이 모이는 장소이다. 리소스풀은 원으로 표현된다. 자원들이 리소스풀에 저장되어 있을 때는 더 작은 원이 쌓여있는 형태로 표현된다. 만약에 리소스풀에 자원이 너무 많이 있어서 쌓아서 표현할 수 없는 경우에는 숫자의 형태로 대신 표현한다.



자원이 5개인  
리소스풀



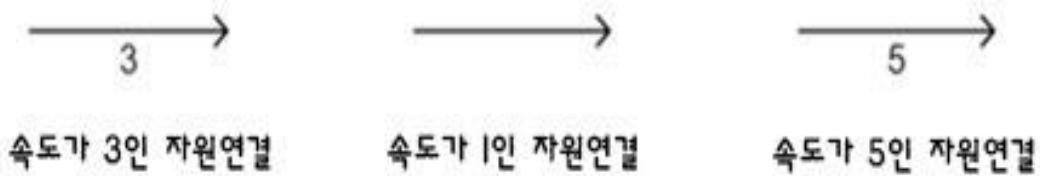
빈 리소스풀



자원이 125개인  
리소스풀

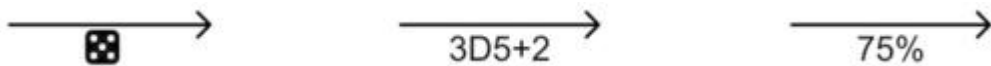
리소스풀은 주로 엔티티(entity)를 모델링한다. 예를 들면, 만약에 당신이 돈이라고 불리는 자원을 가지고 있고, 플레이어의 은행 계좌라고 불리는 엔티티가 있다면, 당신은 아마 리소스풀을 은행계좌를 모델링하는 데 사용할 것이다. 이 리소스풀은 분수는 저장할 수 없고, 단지 정수만 저장할 수 있다는 것을 기억해 두자.

가장 단순한 종류의 연결은 자원연결(resource connection)이다. 그리고 자원연결은 자원을 한 노드에서 다른 노드로 전송한다. 자원연결은 다이어그램에서 노드들 간에 연결된 실선으로 표현되며, 자원을 다양한 속도(rate)로 보낼수 있다. 자원연결 옆에 있는 라벨은 한 타임스텝에 얼마나 많은 자원이 이 커넥션을 통해 이동하는지를 나타낸다. 만약 자원연결이 라벨이 없으면, 그것의 속도는 1 이라고 생각하면 된다.



자원연결의 흐름 속도를 랜덤으로 하려면 라벨 상자에 입력하면 된다. 랜덤 속도는 다양한 방법으로 표시된다. 만약에 단순히 D 라고 입력하면, 주사위(dice) 심볼이 자원연결 옆에 나타날 것이고, 이것은 정의되지 않은 임의의 값을 나타낸다. 머시네이션 툴은 종지와 펜으로 하는 롤플레이팅 게임에서 주로 사용하는 방식으로 임의의 숫자를 생성할 수 있다. 이때 D6 은 6면체 주사위를 한번 굴려서 나오는 임의의 숫자를 의미한다. 반면에 D6+3 은 같은 주사위에 3을 더하는 것이고, 2D6 은 두 육면체 주사위의 값을 더하는 것으로 즉, 2에서 12사이의 값이 될 것이다.

또, 퍼센트를 이용해서 임의의 값을 만들어낼 수 더 있다. 25%라고 적혀있는 자원연결은 각 타임스텝마다 25 퍼센트의 확률로 자원이 흐를 수 있다는 의미이다. 퍼센트를 사용할 때에는 100% 이상의 값을 사용할 수도 있다. 예를 들면, 250%는 최소한 2 개의 자원과 함께, 50% 확률로 자원 한 개가 흐를 수 있음을 의미한다.



### **활성모드(activation mode)**

각각의 이터레이션에서 머시네이션즈 다이어그램의 노드들이 활성화(fire)화 될 수 있다. 노드가 활성화될 때 노드는 연결을 따라 자원을 보내거나 가지고 온다. 노드가 발화될지 안 될지는 그것의 활성화모드(activation)에 따라 다르다. 머시네이션즈 다이어그램의 노드의 활성화모드는 다음 4 가지 모드 중 한 가지가 된다.

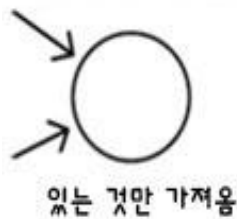
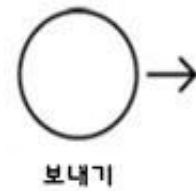
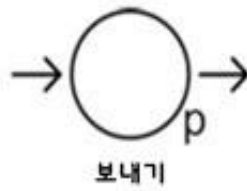
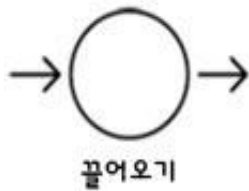
- 노드가 자동(automatically)으로 발화할 수 있다. 이것은 노드가 매 이터레이션마다 발화된다는 의미이다. 모든 오토매틱 노드는 동시에 발화된다. 오토매틱 노드는 별표(\*)를 달아 표시한다.

- 노드가 상호작용(interactive)할 수 있다. 이것은 사용자의 행동을 나타낸다. 상호작용 노드는 유저가 클릭을 한 후에 발화된다. 상호작용 노드는 두 줄의 아웃라인으로 표시한다.
- 노드가 시작동작(starting action)일 수 있다. 이것은 노드가 첫 인터레이션을 돌 적에 딱 한번만 발화됨을 의미한다. 머시네이션즈 도구에서 시작동작은 유저가 Run 버튼을 누른 바로 뒤에 발화되며, S 를 달아 표시한다.
- 노드가 수동(passive)일 수 있다. 이것은 다른 요소에 의해 생성된 트리거(trigger)에 반응할 때만 발화한다. (트리거에 대해서는 뒤에서 다시 설명한다.) 자원은 다른 노드에 의해서 나오거나 들어갈 수 있는데, 트리거되지 않는 한 그 스스로 자원을 보내거나 끌어오지 않는다. 수동 노드는 따로 표시하지 않는다.



### 자원을 보내거나 끌어오기

리소스풀이 발화할 때, 리소스풀은 연결된 입력을 통해 자원을 끌어오려고 시도할 것이다. 리소스풀이 끌어오는 자원의 양은 각 입력 자원연결의 속도에 따라서 결정된다. 반대로 리소스풀을 보내기 모드로 세팅할 수도 있다. 보내기 모드에서는 리소스풀이 발화될 때 자원을 출력연결을 통해 지정된 속도만큼 보내게 된다. 보내기 모드인 리소스풀은 P 로 표시한다. 출력만 있는 리소스풀은 항상 보내기 모드로 여겨지고, 이때에는 P 를 생략한다.



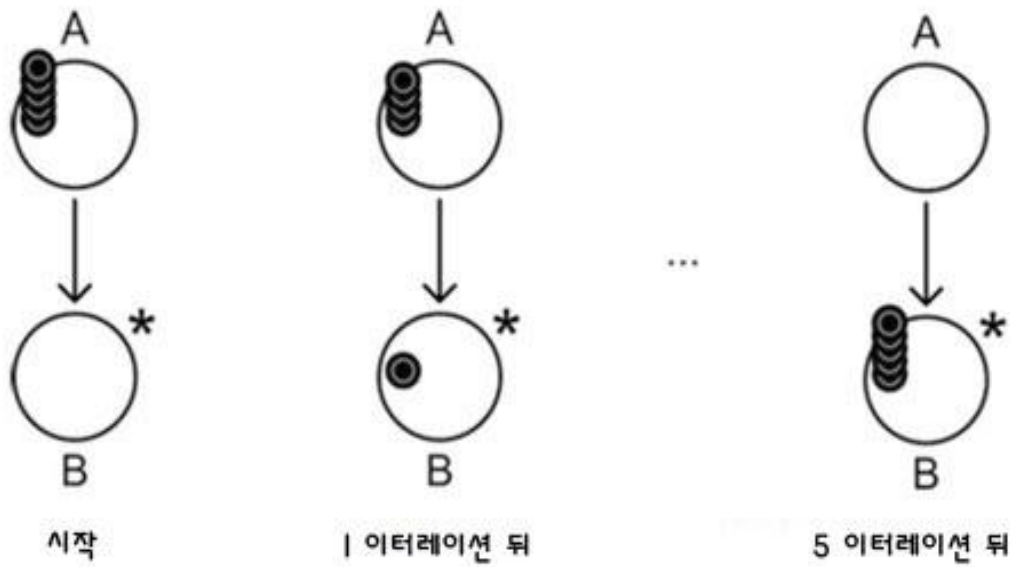
만약에 리소스풀이 입력의 가장 끝에 있는 자원보다 많은 양을 끌어오려고 한다면, 다음 둘 중 하나의 방식으로 처리될 것이다.

- 기본적으로 노드는 가능한 한 최대한의 자원을 입력의 최대 속도까지 끌어온다. 만약에 자원이 부족하다면, 자원이 존재하는 만큼만 끌어온다.
- 그렇지 않으면 노드는 자원을 전부 다 가지고 오거나 하나도 가지고 오지 않을 수 있다. 이 모드에서는 자원이 충분하지 않으면 아무것도 끌어오지 않는다. 모두 가지고 오거나 가지고 오지 않는 타입의 노드는 & 으로 표시한다.

### 모래시계 예제

리소스풀과 자원연결을 사용해서 단순한 모래시계를 만들 수 있다. 이 경우 두 개의 리소스풀이 하나의 자원연결로 연결되어 있다. 위에 있는 리소스풀(A)는 수동 모드이고 자원을 5 개 갖고 있다. 반면에 아래에 있는 리소스풀(B)는 자동 모드이고 자원 없이 시작한다. 각 이터레이션 뒤에, B 는 모든 자원이 A 에서 B 로 이동할 때까지 A 에서 자원을 하나씩 가지고 올 것이다. 자원이 모두 이동한 다음에는 다이어그램의 상태가 더이상 변하지 않을 것이다.





## 상태변화

머시네이션즈 다이어그램의 상태(state)는 현재 노드들 간의 자원 분배를 말한다. 자원이 한 곳에서 다른 곳으로 이동할 때 상태가 바뀌게 된다. 머시네이션즈 프레임워크에서 자원연결의 흐름의 속도를 조정하기 위해서 상태변화를 사용할 수 있다. 또한 자원의 분배를 조정함으로써 노드가 발화되도록 트리거하거나, 노드를 활성화 혹은 비활성화 시킬 수 있다.

이를 위해 머시네이션즈는 두번째 종류의 연결을 제공하는데, 이것을 상태연결(state connection)이라고 부른다. 상태연결은 현재 노드의 상태(해당 노드가 가지고 있는 자원의 수)의 변화가 다이어그램의 다른부분에 어떻게 영향을 미칠지를 의미한다.

상태연결은 조정하는 노드(controlling node)로부터 목표로 이어지는 점선 화살표로 표시된다. 목표는 노드일 수도 있고, 자원연결이나, 드물게는 다른 상태연결일 수 있다. 상태연결의 라벨은 타겟을 어떻게 변경할지 나타낸다.

상태연결은 연결하고 있는 요소의 종류와 라벨에 따라서 4 가지로 나눌 수 있다. 라벨변경자(label modifier), 노드변경자(node modifier), 트리거(trigger),

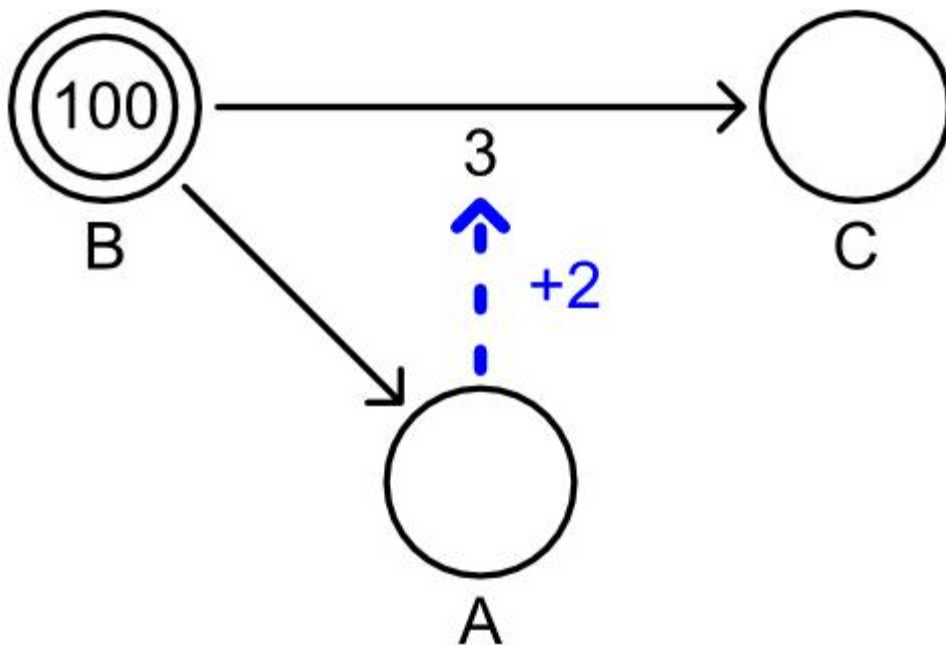
액티베이터(activator)가 그것이다. 이 글은 요약본이기 때문에 라벨변경자와 액티베이터 두가지만 다루도록 하자.

### 라벨변경자

자원연결의 라벨은 주어진 타임스텝에서 연결을 통해 이동하는 자원의 양을 결정한다는 것을 기억하는가. 라벨변경자는 시작노드와 자원연결(혹은 다른 상태연결)의 라벨을 연결한다.

라벨변경자는 시작노드의 상태변화가 목표 라벨의 현재값을 어떻게 변경하는지를 의미한다. 변화는 현 타임스텝에 일어나고 얼마나 변경할지는 상태연결의 라벨에 따라 다르다. 변경된 새로운 값은 다음 타임스텝에 영향력을 행사하게 된다. 시작노드의 변하는 양은 라벨 변경자의 라벨에 곱해진다.

그래서 만약에 라벨변경자의 값이 +3 이고, 시작노드가 2 만큼 증가하면, 목표 라벨은 다음 타임스텝에 6 만큼 증가하게 된다. (시작노드의 값이 하나 바뀔때마다 3 씩 더한다.) 만약에 라벨변경자의 값이 +3 이고 시작노드가 2 만큼 감소하게 되면, 목표라벨은 6 만큼 감소할 것이다.



파란 점선은 라벨변경자를 나타낸다.

라벨변경자의 라벨은 항상 + 나 - 부호로 시작한다. 예를 들면, 위의 그림에서, 리소스풀 A 에 추가된 모든 리소스는 리소스풀 B와 C 사이의 자원흐름에 2씩 더하게 된다. 따라서 처음에 B가 활성화되면, 리소스 하나는 A 에 추가되고 3개는 C에 추가된다. 두번째 이터레이션에서, A 로는 하나의 자원이 추가되지만, C 로는 5개의 자원이 추가된다.

### **액티베이터**

액티베이터는 두개의 노드를 연결한다. 액티베이터는 목표노드를 시작노드의 상태와 특정한 조건에 의거해서 활성화하거나 억제하고, 액티베이터의 라벨에 이 조건을 명시한다. 조건들은 산술표현(예를 들면,  $=0$ ,  $<3$ ,  $\geq 4$  혹은  $\neq 2$ ) 혹은 값의 범위(예를 들면, 3-6)로 나타내어진다. 만약 시작노드의 상태가 이 조건을 만족하면 목표노드가 활성화된다. (발화할 수 있다). 조건이 만족되지 않았을 때에는 목표노드는 억제된다. (발화할 수 없다.)

## **고급노드의 종류**

머시네이션즈 다이어그램에서 리소스풀 노드만 존재하는 것이 아니다. 이번 섹션에서는 4 가지 경제기능(소스 source, 드레인 drain, 컨버터 converter, 트레이더 trader)을 포함해 우리가 사용할 수 있는 좀 더 많은 종류의 노드들 다룰 것이다. 이 네 가지 경제기능들은 <Fundamentals of Game Design>에 처음 소개되었다.

### **게이트(gate)**

리소스풀과는 반대로 게이트는 자원을 모으지 않는다. 대신에 게이트는 자원을 바로 재분배한다. 게이트는 다이아몬드 형태로 나타내고 출력이 여러 개인 경우가 많다. 각 출력의 라벨에는 흐름 속도 대신에 확률이나 조건이 명시된다. 라벨에 확률을 표시하는 출력을 확률출력(probable output)이라고 부른다. 반면에 라벨에 조건을 표시하는 출력은 조건부출력(conditional output)이라고 한다. 하나의 게이트의 출력은 반드시 모두 같은 종류여야 한다.

확률은 퍼센트(예를 들면 20%)나 숫자로 표시된 가중치(예를 들면 1 이나 3)로 나타낼 수 있다. 퍼센트일 때는 게이트로 들어오는 자원의 확률이 각 출력의 확률과 같다. 출력 확률들의 합은 100 퍼센트를 넘을 수 없다. 출력확률의 합이

100 퍼센트보다 적을 경우에는, 리소스가 다른 출력으로 보내지지 않고 파괴되어 사라질 수도 있다.

가중치의 경우, 자원이 특정한 출력으로 보내어질 확률은 게이트 출력의 가중치의 합에서 그것의 출력을 나눈것과 같다. 다시 말해, 두 개의 출력이 있다고 할 때 하나는 가중치가 1 이고 다른 하나는 3 이라고 하자. 자원이 첫번째 출력으로 보내질 확률은 1/4 이고 두번째 출력으로 보내질 확률은 3/4 이다.

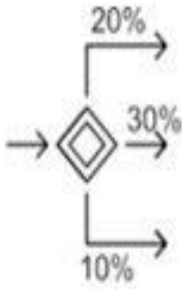
출력 라벨이 조건인 경우(>3, ==0, 3-5 와 같이) 이것은 조건부출력이다. 이 경우에는 어떤 자원이 게이트에 다다를 때 마다 모든 조건이 체크되고, 해당 조건이 만족된 모든 출력으로 자원이 보내지게 된다. 조건은 중복될 수 있다. 조건이 중복되면 자원을 복제하거나, 만족된 조건이 없는 경우에 자원을 파괴할 수 있다.

게이트의 분배모드는 두 개의 분배모드 중 하나다. 결정론적 분배 혹은 임의의 분배. 결정론적 게이트(deterministic gate)는 가능한 출력이 있을 때 자원을 퍼센트나 가중치로 표시된 분배 확률에 따라서 고르게 분배할 것이다. 조건부 출력이 있다면 각 타임스텝마다 해당 게이트를 지나간 자원의 수를 계산해서 그 수를 이용해 출력의 조건을 체크하게 된다. (조건부 출력을 가진 결정론적 게이트는 카운팅 게이트라고 생각하면 편할것이다.) 카운트는 매 타임스텝마다 새로 시작된다는 것을 기억해 두자. 그래서 만약 자원이 하나 도착하면 결정론적 게이트는 특정한 기호 없이 작은 다이아몬드로 표시한다.

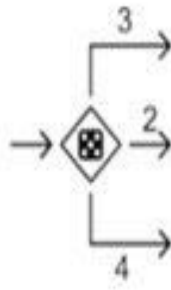
임의의 게이트(random gate)는 들어온 자원을 어디에 분배할지 결정하기 위해서 임의의 값을 생성한다. 가능한 출력이 있을 때, 게이트는 적합한 숫자를 생성할 것이다(0 에서 100 퍼센트 사이일 수도 있고 출력 전체 가중치의 합보다 작은 하나의 값일 수도 있다.) 출력이 조건부인 경우에는 마치 다이어그램이 6 면체 주사위 역할을 하듯 1 에서 6 사이의 값을 만들어서 조건들을 체크할 것이다. 임의의 게이트는 주사위 기호로 표시한다.

게이트는 매우 강력하고 공간제약상 여기서 설명하지 않는 다른 여러 기능이 있다. 아래는 게이트 종류 중 일부이다

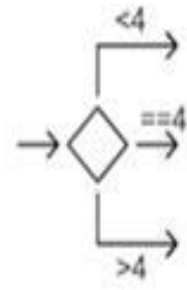
.



가능한 출력이 있는 상호작용적 결정론적 게이트



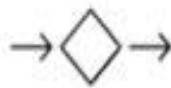
가능한 출력이 있는 임의의 게이트



조건부 출력의 결정론적 게이트



가능한 출력이 하나인 임의의 게이트

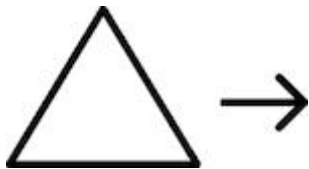


가능한 출력이 결정론적 게이트



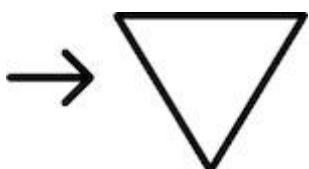
조건부 출력 하나의 결정론적 게이트

## 소스



소스는 자원을 생성하는 노드를 말하며 위를 향하고 있는 삼각형으로 표현된다. 머시네이션즈 다이어그램의 다른 노드들과 마찬가지로, 자동적(기본값), 상호작용, 수동적, 다이어그램이 시작할때만 한번 작동할 수 있다. 자동적인 소스의 예중 하나로 <스타워즈: 엑스윙 얼라이언스(Star Wars: X-Wing Alliance)> 에서 플레이어의 스타파이터 방어쉴드가 계속 꾸준히 생성되는 것을 들수 있다. <리스크(Risk)> 에서 군대를 양성하기 위한 행동은 군대의 상호작용 소스로 모델링 될 수 있다. <모노폴리(Monopoly)> 에서 시작지점을 지나가는 것은 게임이벤트에 의해 트리거 되는 수동적인 소스가 될 수 있다. 소스가 생성하는 자원의 양은 소스의 기본적인 특징이고 소스 출력의 라벨에 그 양을 표시할 수 있다.

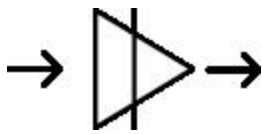
## 드레인



드레인은 자원을 소모하는 노드이다. 드레인으로 들어간 자원은 영구적으로 사라진다. 머시네이션즈 프레임워크에는 아래를 향하고 있는 삼각형으로 표시하는 특별한 드레인

노드가 있다. 드레인이 소모하는 양은 그것의 입력 자원연결의 라벨에 따라 결정된다. 어떤 드레인은 자원을 일정한 양으로 소모하고, 반면에 다른것들은 임의의 양이나 특정값 사이의 양을 소모한다. 또한 입력 자원연결의 라벨을 all(\*)으로 정의하면 드레인이 입력 자원연결에 있는 모든 자원을 소모하게 만들 수 있다.

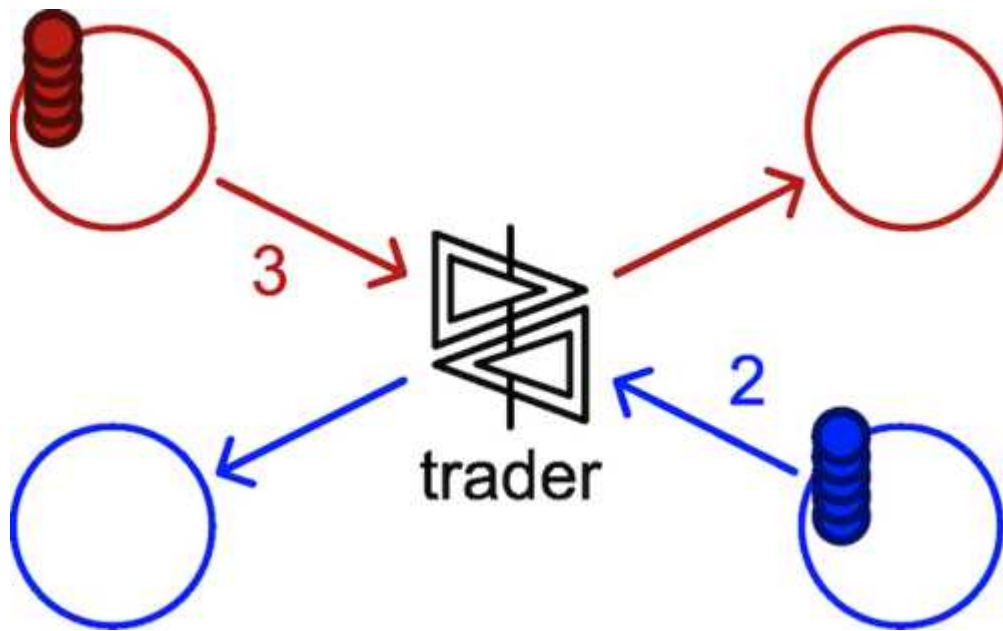
### 컨버터



컨버터는 하나의 자원을 다른 자원으로 변환한다. 오른쪽을 향하는 삼각형과 이를 관통하는 수직선이 있는 모양으로 표시된다. 컨버터는 원자재를 완성된 제품으로 바꾸는 공장과 같은 것을 모델링하기 위해서 만들어졌다. 예를 들면, 풍차는 밀을 밀가루로 변환한다. 컨버터는 정확히 소스를 트리거하는 드레인(다른 리소스를 만들기 위해서 하나의 리소스를 소모하는 것)과 같이 동작한다. 소스와 드레인과 마찬가지로, 컨버터도 소모하고 만들어내는 자원의 양을 입력과 출력 라벨의 값에 따라 다르게 할 수 있다. 예를 들면, 제제소를 의미하는 컨버터는 하나의 나무를 50 개의 합판으로 변환한다.

### 트레이더

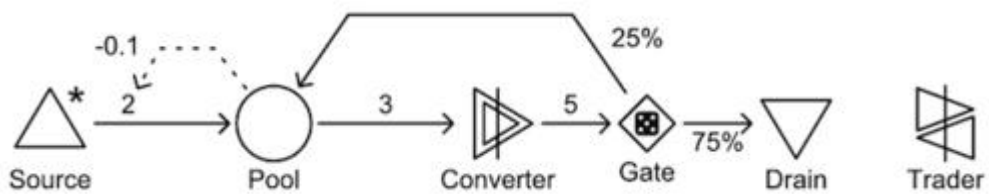
트레이더는 발화되었을 때 자원의 소유권을 바꾸는 노드이다. 두 명의 플레이어가 자원을 교환하기 위해 트레이더를 사용할 것이다. 머시네이션즈 다이어그램에서 트레이더는 서로 마주보고 있는 두개의 삼각형을 수직선이 가르고 있는 모양으로 표현된다. 한 종류의 자원 몇 개가 다른 종류의 자원으로 교환(변환이 아닌)될 때 트레이더를 사용한다. 이것은 쇼핑과 유사한 상황에 다 적합하다. 상인은 돈을 받고 고객은 정의된 양(가격)만큼의 상품을 받는다. 만약에 상인이나 고객 중 어느 한 쪽이라도 필요한 자원을 가지고 있지 않으면 이 거래는 성사되지 않는다. 모든 상인의 물자가 제한되어 있는 <폴아웃 3(Fallout 3)>가 좋은 예이다.



상호작용하는 트레이더(두 줄로 표현됨).  
 클릭하면 3개의 빨간 자원을 2개의 파란 자원으로 교환한다.

**모두 모아보자**

여기까지가 <게임 메카닉> 챕터 5 발체의 끝이다. 하지만, 여러분이 머시네이션즈를 이용해 만들 수 있는 예를 몇 가지 보여주고 싶다. 아래의 다이어그램은 앞에서 소개한 모든 요소를 포함하고 있다.



머시네이션즈 요소들의 모습

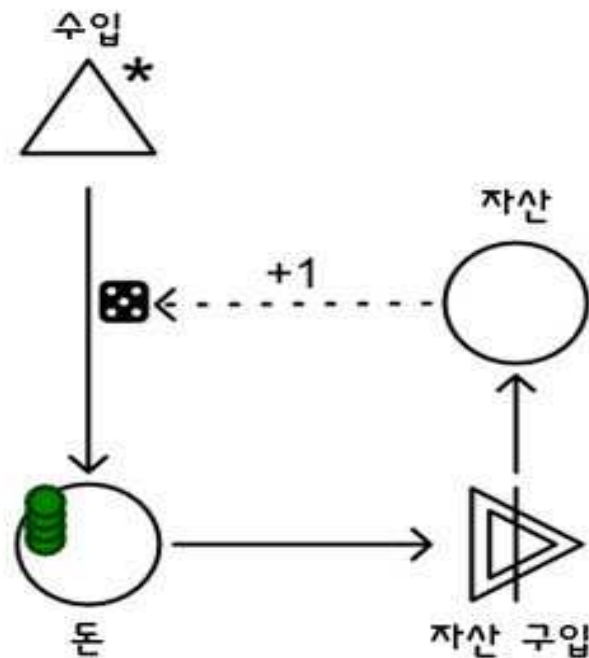
왼쪽에서 오른쪽으로 진행하며 자동 소스(소스옆에 표시된 \* 은 자동을 의미한다)가 있다. 이 소스는 매 타임스텝마다 2개의 자원을 생성하고, 생성된 자원을 수동

리소스풀로 보낸다. 리소스풀은 상태연결의 시작노드이고(점선으로 표시됨) 자원이 도착할 때마다 입력 자원 연결의 라벨을 0.1씩 줄인다. 20 이터레이션 후에는 라벨은 0 으로 줄어들 것이고, 소스는 더이상 자원을 보내지 않을 것이다.

리소스풀은 상호작용하는 컨버터와 연결되어 있다. 이 컨버터는 클릭할 때마다 3 개의 자원을 5 개로 변환한다. 출력된 5 개의 자원은 임의 게이트로 보낸다. 그 중 25%는 리소스풀로 돌아가고, 나머지 75%는 드레인 노드로 보내지고 사라진다. 오른쪽에 있는 트레이더는 아직 어느것도 연결되어 있지 않다.

## 몇 가지 예제

머시네이션즈는 임의의 수준으로 추상화한 동적 시스템을 모델링하고 시뮬레이션할 수 있다. 여기 <모노폴리> 의 기본적인 포지티브 피드백 루프(positive feedback loop) 예제가 있다.



모노폴리 의 양의 피드백 루프

각각의 타입시스템마다 1에서 6사이의 수입(Income)이 들어온다. (머시네이션즈 는 기본적으로 6면체 주사위값을 생성하게 되어있지만, 이 값은 변경할 수 있다.) 이

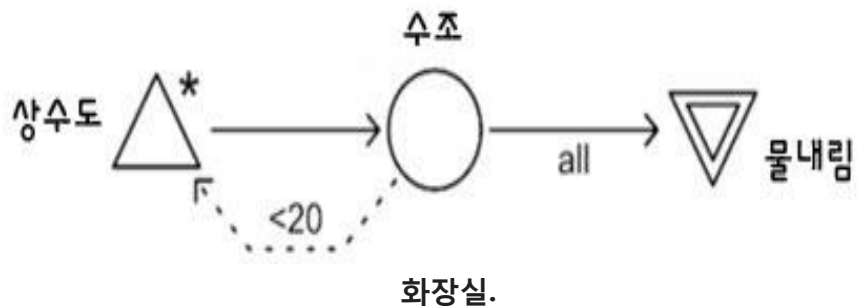


수입은 수동 리소스풀로 들어간다.

어느 시점에 플레이어는 자산 구입(Buy Property) 컨버터를 클릭할 수 있다. 이것은 하나의 돈(money) 자원을 자산(property) 자원으로 변환하고 자산 리소스풀로 보낸다. 자산 리소스풀의 상태변화는 수입 소스의 출력 연결의 라벨로 전송이 되고, 다음 턴에 받을 수입을 증가시킨다.

이것이 <모노폴리> 부터 <스타크래프트(Starcraft)> 에 이르는 고전적인 업그레이드 프로세스이다. 생산 건물 투자하려면 저장된 자원을 소모해야 하지만, 다른 자원의 생산을 높인다.

다음으로 우리에게 친숙한 자동 조절 메커니즘을 소개한다.



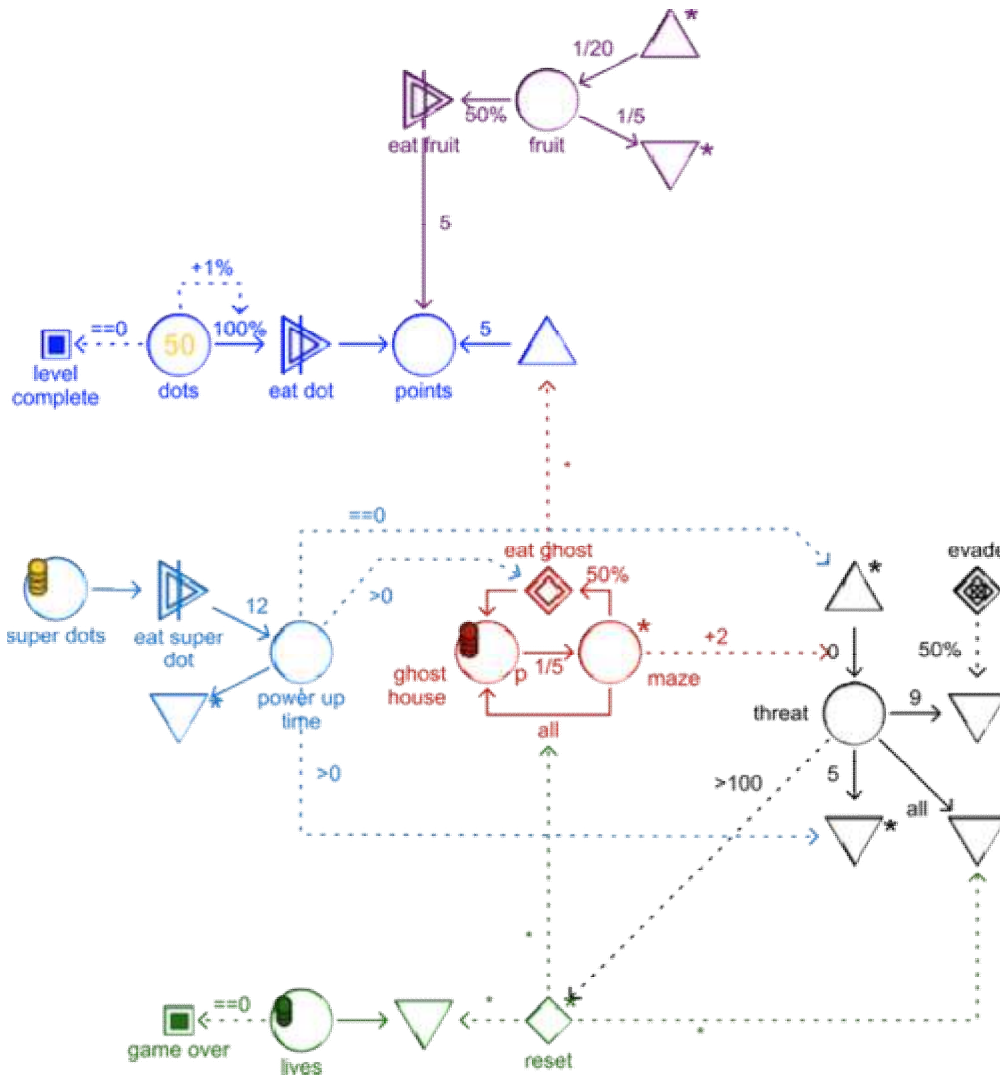
수조로부터 상수도에 이르는 상태연결은 라벨변경자가 아니라 액티베이터이다. 이것은 수조의 자원이 20 미만으로 떨어졌을 때 상수도가 물을 제공할 수 있게 해준다. 그리고 수조가 꽉 차게 되면 물 공급은 끊어지게 된다(머시네이션즈 틀에서는 회색으로 표시된다.) 상호작용하는 드레인은 현재 수조에 있는 자원을 클릭할 때 마다 모두 비우게 된다. all 이라는 단어는 자원연결에서 무한한 양을 나타내기 위해서 특별히 디자인된 값이다.

이것들은 머시네이션즈를 이용해서 여러분이 만들 수 있는 피드백루프나 컨트롤시스템 아이디어의 일반적인 예일 뿐이다. 이 책에는 디자인 패턴의 모든 라이브러리가 들어있다. — 여러 종류의 엔진과 마찰 시스템(friction systems), 에스컬레이션 패턴(escalation pattern), 테크놀로지 트리같은 것을 위한 기타 패턴들.

이 글에서는 머시네이션즈가 제공하는 많은 강력한 기능들의 1/4 도 채 안되는 분량만 소개했다. 우리는 머시네이션즈를 이용해 모든 종류의 게임을 만들어 낼 수

있다. <게임 메카닉>의 챕터 6 에서는 횡스크롤 액션 게임, 슈팅, RPG, 레이싱 등의 예제를 포함하고 있다. 이는 완벽한 게임을 만들기 위해서가 아니라 게임의 경제를 만들고 테스트하고 정교하게 만들기 위해서 디자인되었다. 또한 빠른 속도로 많은 실행을 시킬 수 있고, 자료를 모으고 표시하거나 뽑아낼 수 있을 것이다. 그리고 유저 행동의 다양한 전략들을 실험하고 그 중에 어느 하나가 다른 것들을 압도하는지도 볼 수 있을 것이다.

챕터 5 의 나머지 부분은 <팩맨(Pac-Man)>의 다양한 기능들을 유령부터 과일, 슈퍼도트(super dot) 까지 차례차례 모델링하는 것이다. 머시네이션즈 그 자체로 미로의 배치는 만들수는 없지만, 여러가지 메커니즘과 그들이 서로서로 어떻게 상호작용하는지 보여줄 것이다. 다음이 챕터 5 의 마지막 결과이다. (주의. 여기에는 이 글에서 설명하지 않는 기능들도 포함되어 있다.)



완성된 Pac-Man 의 내부경제

머시네이션즈 툴은 많은 예제 다이어그램과 함께 요리스 도르만즈의 웹사이트에서 무료로 받을 수 있다. <http://www.jorisdormans.nl/machinations>