기믹 사용 설명서

업데이트 날짜: 2015년 9월 18일

작성자: 정지완

본문서는 TT프로젝트에 사용되는 기믹의 구조와 설명해 대해 다룹니다.

1. **프리팹 위치**

모든 프리팹의 위치는 **Prefabs/Obstacles** 폴더 아래에 존재합니다.

**현재 작업완료된 기믹 프리팹:**

G\_001\_Spine

G\_002\_DownPlatform

G\_003\_MovePlatform

T\_001\_Trigger는 트리거 프리팹인데 현재 사용은 가능하지만 현재 부착된 아트 리소스가 없습니다. 관련 스케쥴일정이 논의되어야할 것 같습니다.

1. **기믹의 구조**

모든 기믹 프리팹들은 모두 SCPT\_로 시작하는 이름의 하위 객체를 가지고 있으며 기믹에 관련된 모든 컴포넌트와 스크립트는 여기에 부착되어있습니다.

하위객체명을 placeA\_Here로 해둔 이유는 제가 아닌 다른 분이 기믹을 조립할 경우에도

아트팀에서 만들어주신 A\_ 로 시작되는 아트 리소스를 이 아래 부착하는 것만으로도 기믹 프리팹 조립을 가능케 하기 위해서입니다.

1. **기믹의 기능**

모든 기믹들은 SCPT\_ 에 그 기능을 조정할수 있는 스크립트가 부착되어 있으며 **isTriggered가 체크되어 있지 않으면 그 기능을 발휘 하지 않습니다**.

모든 기믹들은 최초 배치시 **isTriggered가 체크되어있지 않으니** 게임이 시작되자마자 **기능을 발현하게 하고 싶은경우 isTriggered를 체크한후 배치** 해주십시오.

각 기믹들의 스크립트 인스펙터에 공개된 변수들은 모두 수정이 가능한 변수이며, 건드리지 말아야할 변수들은 모두 private으로 숨겨두었으니 안심하고 조작하셔도 됩니다.

* 1. 상세기능
     1. D\_G\_001\_Spine : 가시

부착된 스크립트는 CommonSpine입니다.

**Damage**: 가시에 닿았을 때 플레이어가 입는 데미지의 양입니다.

**KnockBackDist**: 가시에 닿았을 때 플레이어가 밀려나는 넉백의 양입니다. 미터 기준.

.

**MaxLoop**: 0이면 무한루프 0이외의 양수일경우는 설정된 만큼만 반복합니다.

**Speed**: 가시기둥의 이동속도입니다. m/s 기준

**IsRotateDir**: 가시가 이동시 이동방향으로 바라보도록 회전 시킬 것 인지?

**IsPingPong**: 가시가 이동시 마지막 루트에 도달하면 역순으로 돌아올것인지 아님 첫번째 루트로 이동할것인지를 정합니다.

**StoppingDist**: 루트에 얼마나 가까워졌을 때 다 도달한것으로보고 다음 루트로 이동할것인지 설정합니다. 미터 기준.

**MoveWaitTime**: 가시가 루트에 도착하고 나서 몇초를 대기하고 다음 루트로 이동하는지 설정하는 값. 초 기준.

* + 1. D\_G\_002\_DownPlatform : 낙하다리

부착된 스크립트는 FallPlatform입니다.

**EnduranceTime**: 낙하다리가 낙하하기 전에 버티는 시간. 초 단위

**FallSpeedMultiplier**: 다리가 몇배속으로 떨어지는가. 기본값 1배속.

* + 1. D\_G\_003\_MovePlatform : 이동다리

**MovingRoute**: 빈 게임 객체를 생성하여 **GoPatrolPattern**에 등록해주면 이동루트가 만들어집니다.

새로 생성한 빈게임 객체는 RouteXX 라는 네이밍 규칙으로 movingRoute라는 객체의 하위객체로서 관리하는 것을 권장합니다

* + 1. D\_T\_001\_Trigger : 트리거

부착된 스크립트는 TrapTrigger입니다.

**Traps**: 이곳에 모든 기믹들을 드래그해 부착할수 있습니다. 참고로 각 기믹들의 **SCPT \_**를 여기에 드래그하여 붙여야합니다. (실제 기능은 이 객체에 있으므로)

**DisableOnEnter**: 들어갈때 작동을 종료하도록 하는 옵션입니다. 이 기능을 사용하면 트리거를 밟았을 때 등록된 기믹의 작동을 종료하도록 할수 있습니다.

**DoReverseOnExit**: 트리거에 들어갈때 작동하고 나갈 때 작동을 종료하는 옵션입니다. 기본값은 이 기능을 사용하지 않습니다. (들어갈때만 작동)

DisableOnEnter와 같이 사용하면 들어갈 때 작동을 종료하고 나갈 때 작동이 실행될 것입니다.

* + 1. D\_G\_004\_RunningMachine : 러닝머신

부착된 스크립트는 AttractZone입니다.

스크립트는 프로토타입때 만들었던 퀵샌드 스크립트에서 착안하여, 플레이어를 빨아들이는 포인트와 빨아들이는 속도를 설정하는것으로 런닝머신을 구현하였습니다.

**Attract Point:** 플레이어가 빨려들어가는 포인트 (현재 러닝머신 뒷꽁무니 밖에 배치되어있습니다)

**Attract Speed:** 빨려들어가는 속도. (미터/초)

* + 1. D\_G\_005\_IronMace : 철퇴

부착된 스크립트는 AnimatedDamageZone입니다.

철퇴이외에 애니메이션을 이용해서 움직이는 객체가 플레이어에게 데미지를 주는 경우가 있어 공통으로 사용할만한 스크립트를 만들었습니다.

**Damage**: 플레이어가 철퇴에 닿았을 때 플레이어가 입는 데미지의 양입니다.

**KnockBackDist**: 철퇴에 닿았을 때 플레이어가 밀려나는 넉백의 양입니다. 미터 기준.

**ForceDir**: 플레이어가 철퇴에 충돌했을시 날아가는 고정 방향을 정할수 있습니다. 0으로 설정시 맞은 방향의 반대방향으로 무조건 날아갑니다.

**isLoop**: 철퇴가 아래 CycleWaitTime에 맞춰 반복 재생될것인지를 체크합니다.

**isLoop**가 꺼져있는경우 OneShot으로 플레이되며, 한 사이클이 재생되면 isTriggered가 체크 해제되면서 대기 상태가 됩니다.

**SwingAngle:** 철퇴가 스윙하는 각도를 설정합니다. 현재 90도와 180도 사이의 각도를 지원합니다.

**CycleWaitTime**: 철퇴가 한 사이클 애니메이션을 재생한 후, 입력된 값만큼 대기했다가 (초 단위) 다시 재생합니다.

**SpeedMultiplier**: 애니메이션 재생 배수를 설정합니다. 기본 값 1배속.

* + 1. D\_G\_006\_RotatingPlatform : 회전다리

부착된 스크립트는 RotatingPlatform입니다.

**MaxLoop**: 설정한 값만큼 반복합니다. 0으로 설정할경우 무한 루프.

**LoopCount**: 현재 몇바퀴를 돌았는지를 알려주는 값입니다.

**RotateSpeed**: 회전 속도를 설정합니다. 초당 n도.

**isClockwise**: 시계방향으로 회전할지 반시계로 회전할지 정합니다.

**E Axis**: 어떤 축을 기준으로 회전할지 정합니다. 기본으로는 우측 축을 이용하여 회전하므로 앞뒤로 회전합니다.

* + 1. D\_G\_007\_TimerBlock: 타이머 블록

부착된 스크립트는 TimerBlock, TimerFunctionCaller 두가지입니다.

**TimerBlock**

**Is Show**: 최초 시작시 보인상태 또는 숨겨진 상태로 할건지 정합니다. 이후 가동중일때는 현 상태를 표시해주는 값이 됩니다.

**Show Time**: 보인 상태를 유지하는 시간입니다. (초)

**Hide Start Rate**: 보인 상태를 유지하는 시간의 몇 %구간부터 사라지는 효과를 보여줄지 정합니다. (Show Time: 10, Hide Start Rate: 0.8일 경우, 8초 구간부터 사라지는 효과가 재생되기 시작합니다.)

**Hide Time**: 숨겨진 상태를 유지하는 시간입니다. (초)

**Blink Interval**: 반짝이는 효과를 몇초 간격으로 할지 정하는 값입니다. (초)

**TimerFunctionCaller**

**Wait Time**: 몇 초 이후에 설정된 함수를 호출할 것인지를 정합니다.

**Is Loop**: 설정한 시간마다 설정된 함수를 반복적으로 부를지를 설정합니다.

**Function Name**: 호출할 함수의 명칭. (Toggle Trigger)

* + 1. D\_G\_008\_Crossing: 건널목 장애물

부착된 스크립트는 MovingRoute, InterpolatingObject 두가지입니다.

**MovingRoute**

3.1.1 가시 참조.

**InterpolatingObject**

**Damage**: 건널목 장애물이 주는 데미지

**Knockback Dist**: 플레이어가 데미지를 입었을시 밀려나는 값

**Max Loop**: 건널목 장애물의 이동 횟수. (0일 경우 무한 반복)

**Time Between Nodes**: 노드간 이동시 소요되야할 시간. (건널목 장애물은 속도 값이 존재하지 않고, movingRoute에서 설정한 노드에서 노드간에 걸려야할 시간만 입력해주면 다음 노드까지 가속하며 설정한 시간내에 도착합니다)

**Is PingPong**: 핑퐁방식의 반복 알고리즘을 가져갈것인지를 설정. (노드가 3개 이상일때만 유효)

**Move Wait Time:** 노드에 도착했을시 다음 노드로 이동하기까지 대기하는 시간 (초)

* + 1. D\_G\_009\_Jumppad: 점프대

부착된 스크립트는 JumpPaddle 입니다.

**JumpPaddle**

**Bounce Height**: 점프 높이

**Forward Force**: (점프대가 설치된 방향으로) 플레이어를 앞으로 날려주는 세기

정확한 거리 추산을 위해 공식을 R&D중입니다. 기다려주세요!

이상입니다. 배치시 자세한 사항은 저에게 문의해주세요.

감사합니다.