2017년 졸업작품

팀 편돌이

프로젝트 편돌이

작성자 : 신창섭

변동사항

2016년 8월 16일 : 초안 작성. 플레이어 조작 관련 기획서 추가

2016년 8월 18일 : 플레이어 스탯 관련 기획서 작성중

1. 목차

[1 플레이어 조작 및 상태 4](#_Toc459317846)

[2 플레이어 설정 및 공식 9](#_Toc459317847)

# **플레이어 조작 및 상태**

* 1. 기획의도  
     키보드에 최적화된 플레이어 조작방식 작성
  2. 개요  
     플레이어가 플레이를 할 때 기본적으로 사용하게되는 기본조작 / 상태는 **[대기/걷기/뛰기/공격/피해 입음/회피/점프/공격/아이템 사용/상호작용]**으로 나눌 수 있다. 각 상태에 따른 조작 방법 및 애니메이션 출력은 아래와 같다.
     1. 대기
        1. 플레이어가 아무런 조작을 하고 있지 않을 경우 기본적으로 idle 애니메이션을 재생한다.
        2. 플레이어가 전투 모드에 돌입한 후에는 fightidle 애니메이션을 재생한다. 전투 모드로 변경되는 경우는 다음과 같다. fightidle 애니메이션 재생 이후 5초이상 추가로 fightidle 애니메이션을 재생하기 위한 조건을 충족시키지 못하면 다시 idle애니메이션을 재생한다.
           1. 직접 몬스터 / NPC / 사물 / 허공 등 모든 경우에서 공격행위를 할 경우
           2. 플레이어가 낙하 데미지 / 몬스터, 플레이어의 공격 등으로 피해를 입은 경우
           3. 플레이어가 피해를 입지 않더라도 플레이어의 위치를 강제로 이동시키는 모든 행위를 당할 경우
     2. 걷기
        1. 기본적으로 키보드의 방향키를 사용해 플레이어를 조작할 수 있다.
        2. 이동 시에는 캐릭터가 진행방향을 바라보며 walk 애니메이션을 재생한다. 이동속도는 player 테이블의 moveSpeed값을 사용한다.
        3. 두 개의 키를 조합하여 대각선 이동이 가능하다. 동시에 3개 이상의 키를 누를 경우엔 1, 2번쨰 키를 제외한 나머지 키의 입력을 무시한다.
        4. 특정 방향으로 하나의 방향키를 누르다가 반대 방향의 키를 입력할 경우 입력을 무시한다. (예를 들어 왼쪽을 누르고 있을 때 오른쪽을 누르면 입력을 무시한다.)
        5. 1.2.1.2에서 정의한대로 fightidle에 돌입한 이후, 플레이어가 이동조작을 하면 fightwalk 애니메이션을 재생한다. Fightidle과 마찬가지로 5초 이상 fightwalk을 유지할 조건이 충족되지 않을 경우 walk 애니메이션을 재생한다.
           1. 예를 들어 만약 fightwalk 애니메이션 12프레임에 walk 애니메이션으로 전환될 조건을 충족했을 경우, walk 애니메이션의 12프레임부터 애니메이션이 시작된다.
        6. 방향키를 이용해 이동하다가 멈출 경우 가속도에 관계 없이 즉시 행동을 멈춘다.
        7. 이동조작이 불가능한 경우는 다음과 같다.
           1. 앉아있을 경우
           2. 점프 중일 경우
           3. 회피 동작 중인 경우
           4. 공격 / 스킬 사용 중인 경우

이 경우 예외로 공격 중 ‘방향’을 바꾸는 것은 허용된다.

* + - * 1. 상태이상 스킬에 걸려있는 경우(스턴 등)
        2. 강한 공격에 의해 경직된 경우
        3. 스킬 공격에 의해 넘어진 경우
    1. 달리기
       1. 키보드의 A키를 누름과 동시에 이동 조작을 할 경우(방향키) 플레이어 캐릭터가 달릴 수 있다. 달리는 중에는 진행 방향을 바라보며 run 애니메이션을 호출하고, 초당 기력(mp)를 plyaerSkill 테이블에 정의된 값만큼 소모한다.
          1. 예를 들어 만약 walk 애니메이션 12프레임에 run 애니메이션으로 전환될 조건을 충족했을 경우, run 애니메이션의 12프레임부터 애니메이션이 시작된다.
       2. 달리는 도중 A키를 손에서 뗄 경우, 달리기를 취소하며 진행 방향으로 walk 애니메이션을 재생한다.
          1. 1.2.3.1.1과 같이 같은 애니메이션 프레임에서 시작된다.
    2. 공격
       1. 플레이어의 공격 수단은 [약한 공격 / 강한 공격 / 스킬공격]으로 구성된다.
          1. 각 공격에 따른 애니메이션이 종료되면 fightidle로 돌아간다.
       2. 약한 공격(S)를 누르면 플레이어가 보고 있는 방향으로 공격을 실행한다.
          1. 약한 공격에 대한 설정은 playerSkill 테이블에서 정의한다.
          2. 각 공격 애니메이션이 끝난 후 0.7초 이내에 같은 공격을 실행하면 다음 약한 공격으로 이어진다. 예를 들어 약한 공격1을 실행한 상태에서 연속으로 S버튼을 누를 경우 약한 공격2, 3, 4로 이어진다.

0.7초의 값을 confing.xml 테이블의 attackTiming에서 정의한다.

* + - * 1. 각 공격 이후 아무런 추가 조작을 하지 않을 경우 fightIdle 애니메이션으로 되돌아간다.
        2. 공격 도중 방향키를 입력할 경우, 다음 공격은 해당 방향으로 진행된다.
      1. 강한 공격(D)를 누르면 플레이어가 보고 있는 방향으로 공격을 실행한다.
         1. 강한 공격에 대한 설정은 playerSkill 테이블에서 정의한다.
         2. 각각의 강한 공격은 약한 공격 이후에만 실행이 가능하다.

약한 공격1 -> 강한 공격 1

약한 공격2 -> 강한 공격 2

약한 공격3 -> 강한 공격 3

약한 공격4 -> 강한 공격 4

* + - * 1. 각각의 강한 공격이 종료된 이후에는 fightIdle 애니메이션으로 되돌아간다.
      1. 각각의 스킬버튼(Q, W, E, R, T)을 누르면 플레이어가 보고 있는 방향으로 playerSkill 테이블에 정의된 각각의 스킬을 실행한다.
         1. 각각의 모든 스킬은 약한 공격, 강한 공격의 애니메이션이 진행중이라도 즉시 사용할 수 있다. 이 경우 해당 공격을 즉시 중지하고 스킬을 시전한다.
         2. 각각의 스킬 공격은 애니메이션이 끝나기 전까지 아무런 행동을 취할 수 없다.
      2. 공격 중 강한 공격 이상의 공격을 당했을 경우, 즉시 공격을 중지하고 피격 애니메이션을 실행한다. **이후 이와 관련된 설정은 플레이어 테이블 / 스킬 테이블에 power / resiPower 값을 설정해주는 것으로 정의한다.**
    1. 회피
       1. 플레이어는 기본적으로 회피 스킬(A)을 사용해 적의 공격을 회피하거나, 순간적으로 먼 거리를 이동해야 할 때 등에 활용할 수 있다.
       2. 회피 스킬은 기본적으로 A키를 누르면 플레이어가 보고 있는 방향으로 사용되며, 이동 거리 / 소모 기력 등의 설정은 playerSkill 테이블에 정의된 값을 사용한다.
       3. 회피는 어떤 공격(약한 공격 / 강한 공격 / 스킬 공격) 중에도 해당 애니메이션을 취소하고 바로 사용할 수 있다.
       4. 회피 도중 모든 아이템을 사용할 수 없다.
       5. 플레이어가 회피 스킬 사용 도중 공격을 당했을 경우, 스킬 공격이 아니면 피해만 입히고 power 값에 따른 피격 애니메이션을 재생하지 않는다. 하지만 power 값에 따른 스킬공격 이상의 공격을 받으면 즉시 회피 애니메이션을 취소하고, 회피 완료 지점에 이동하면서 hardDam 애니메이션을 재생한다.
    2. 피해 입음  
       플레이어가 피해를 입었을 때에 대한 설정이다.
       1. 플레이어의 피격 애니메이션은 받은 공격의 skill 테이블에 정의된 power 값에 따라 다르게 표출된다.
          1. 강한 공격에 피해를 당했을 때는 normalDam1, normalDam2 중 임의의 애니메이션이 표출된다.

단, 연속으로 강한 공격을 당할 경우 하나의 normalDam 애니메이션이 끝나기 전까지 다른 normalDam 애니메이션을 재생하지 않는다.

* + - * 1. 스킬 공격 이상의 강한 공격을 받을 경우 hardDam 애니메이션을 재생한다.
    1. 점프
       1. 플레이어가 스페이스바(spacebar)를 누르면 제자리 혹은 최초 입력 방향으로 점프를 할 수 있다.
          1. 점프에 따른 체공 시간은 config.xml의 jump에서 정의하며, 정의된 시간(초)만큼 동안 0.5m만큼 상승하며 jump 애니메이션을 재생한다.
       2. 점프는 기본적으로 어떤 자원도 소모하지 않는다.
       3. 점프 도중 다른 방향으로 이동은 불가능하지만, 방향키를 누를 시 점프의 이동 방향은 유지 하면서 플레이어가 보고 있는 방향만 방향키의 방향으로 바뀐다.
       4. 점프 도중에는 playerSkill에 정의된 일부 스킬 이외에는 모든 행동이 불가능하다.
       5. 점프 도중에 공격을 당할 경우, 즉시 해당 공격 power에 맞는 애니메이션을 재생한다. 단, 원래 점프 착지 지점을 이동하면서 애니메이션을 재생한다.
       6. 만약 점프를 할 때 플레이어의 위치에 천장 등이 있어 0.5m 만큼 점프를 하지 못할 경우, 애니메이션을 더 빠르게 재생해서 즉시 점프 동작을 마친다.
       7. 실제 공중 체공시간보다 애니메이션 시간을 약간 더 길게 설정하여, 바닥에 닿았을 때 착지 동작과 동시에 약간 미끄러지듯이 이동한다.
    2. 아이템 사용
       1. 플레이어는 1 ~ 6 번호키를 사용해 등록된 아이템을 사용할 수 있다.
       2. 아이템은 플레이어가 idle, fightIdle 상태일 때만 사용 가능하다. Walk, run, fightIdle, , fightWalk, fightRun상태에서 아이템을 사용할 경우 즉시 자리에 멈춰서 아이템에 따른 애니메이션을 재생한다. 이후 idle / fightIdle 애니메이션으로 돌아간다.
       3. 아이템의 효과는 아이템에 따른 애니메이션이 완료 되었을 때 발휘된다. 아이템 사용 도중 회피를 사용해 아이템 사용을 취소했을 경우, 아이템은 소모되지 않는다.
    3. 상호작용
       1. 플레이어는 상호작용이 가능한 대상이 근처에 있을 때 Spacebar를 누르면 해당 상호작용 대상을 바라보며 chat 애니메이션을 재생한다.

# **플레이어 설정 및 공식**

* 1. 기획의도

플레이어의 각종 수치에 대한 테이블화로 간편한 수정 및 추가가 가능하도록 한다.

같은 내용의 테이블을 서버에도 올려 서버에서 값을 내려 받도록 한다.

* 1. 개요
     1. 플레이어의 **[클래스 / 성별 / 무기 / 스탯]** 등에 대해 테이블에서 정의할 수 있도록 하여, 추후 새로운 클래스가 필요하더라도 작업 소요를 적게 한다.
     2. 플레이어에게 보여질 일부 스탯 값은 테이블의 값들을 사용해 별도 계산을 통해 노출된다.
  2. 플레이어 테이블의 구성요소
     1. 플레이어의 테이블은 다음과 같이 구성된다. **[번호 / 클래스 / 진영 / 캐릭터 이름 / 성별 / 무기타입 / 기본HP / 기본MP / 방어력 / 힘 / 민첩 / 지능 / 건강 / 이동속도 / 일반공격 스킯값 / 달리기 스킬 값 / 회피에서 스킬 값 / 스킬1 값 / 스킬2 값 / 스킬3 값 / 스킬4값 / 스킬5 값 / 플레이어 모델]** 이중 플레이어에게 노출되는 부분은 하단에서 다시 다룬다.
     2. 플레이어의 각 테이블 값은 stat 테이블에서 다음과 같이 정의된다.

|  |  |
| --- | --- |
| no | 캐릭터의 값(정수) |
| class | 캐릭터의 클래스 구분(정수)  0 : 기본 전사 클래스 |
| region | 캐릭터의 진영 구분  0 : 레지스탕스  1 : 얼라이언스 |
| name | 캐릭터의 기본 캐릭터 이름. 우리 게임은 졸작용임으로 그냥 완성형 한글 때려 박을 겁니다. |
| gender | 캐릭터의 성별을 결정합니다. 우리 게임은 기본적으로 하나의 모델링만 사용할 계획임으로 0만 사용하지만 일단 옵션은 만들어 둡니다.  0 : 남자  1 : 여자 |
| Hp | 캐릭터의 기본 체력(정수) |
| Mp | 캐릭터의 기본 기력(정수) |
| Def | 캐릭터의 기본 방어력(정수) |
| Str | 캐릭터의 기본 힘(정수) |
| Agi | 캐릭터의 기본 민첩(정수) |
| Int | 캐릭터의 기본 지능(정수) |
| Health | 캐릭터의 기본 건강(정수) |
| moveSpeed | 캐릭터의 기본 이동 속도(정수)  게임내 1m를 이동할 때 걸리는 시간을 기본으로 하며, 100을 입력하면 1m를 이동하는데 1초가 걸리는 것으로 계산 |
| normalAttack | 캐릭터가 사용하는 기본 약한공격1의 스킬번호. playerSkill 테이블의 No값을 입력한다. |
| runSkill | 캐릭터가 사용하는 기본 달리기 스킬의 스킬번호. playerSkill 테이블의 No 값을 입력한다. |
| dodgeSkill | 캐릭터가 사용하는 기본 회피 스킬의 스킬번호. playerSkill 테이블의 No 값을 입력한다. |
| skill1 | 캐릭터가 사용하는 스킬의 스킬번호. playerSkill 테이블의 no값을 입력한다. |
| skill2 | 캐릭터가 사용하는 스킬의 스킬번호. playerSkill 테이블의 no값을 입력한다. |
| skill3 | 캐릭터가 사용하는 스킬의 스킬번호. playerSkill 테이블의 no값을 입력한다. |
| skill4 | 캐릭터가 사용하는 스킬의 스킬번호. playerSkill 테이블의 no값을 입력한다. |
| skill5 | 캐릭터가 사용하는 스킬의 스킬번호. playerSkill 테이블의 no값을 입력한다. |
| model | 캐릭터의 모델링 파일 경로를 입력한다. |

* 1. 각종 계산 공식
     1. 공격력(다른 모든 것에 피해를 입힐 때)
     2. 방어력(다른 모든 것에 피해를 입을 때 및 유저 노출)
     3. 스탯에 따른 HP 상승량
        1. 플레이어의 체력은 기본적으로 주어지는 HP에 str, health의 값을 종합해서 추가되게 된다 str, health 값에 따른 추가 HP의 값은 다음과 같다.

**HP = HP + (str \* 10) + health \* 10)**

* + 1. 크리티컬 발생 확률
    2. HP 자동 회복
    3. 공격 시 피해량
  1. 플레이어의 레벨업 및 레벨에 따른 플레이어 스탯의 상승
     1. 레벨업에 따른 스탯의 상승량은 playerLevelUp 테이블에서 정의한다. 각각의 값들은 기존 값에 더해지는 형태가 된다.
     2. 플레이어의 경험치 양은 기본적으로 0으로 시작된다. 그리고 현재 레벨 + 1의 요구 exp양을 달성하면 레벨업을 하며, 레벨업을 한 후에 남은 경험치는 다음 레벨의 경험치에 남는다.
     3. 레벨업 시에 플레이어의 체력이 모두 회복된다.

|  |  |
| --- | --- |
| No | 번호 |
| Class | 플레이어 클래스의 구분. 우리 게임은 졸작용이라 기본적으로 전사만 사용될 예정이지만, 일단 클래스 구분은 만들어둠(정수)  0 : 기본 전사  1 : 마법사 |
| Level | 해당 레벨 달성 시(정수) |
| Exp | 해당 level을 달성하기 위해 필요한 경험치의 양. |
| Def |  |
| Str |  |
| Agi |  |
| Int |  |
| Health |  |
| hp |  |

* 1. 플레이어에 사용되는 애니메이션 종류