Unity 3D RPG

01. 월드 맵, 캐릭터 생성 및 이동

2016-09-19

Created By. ChoA.

Index

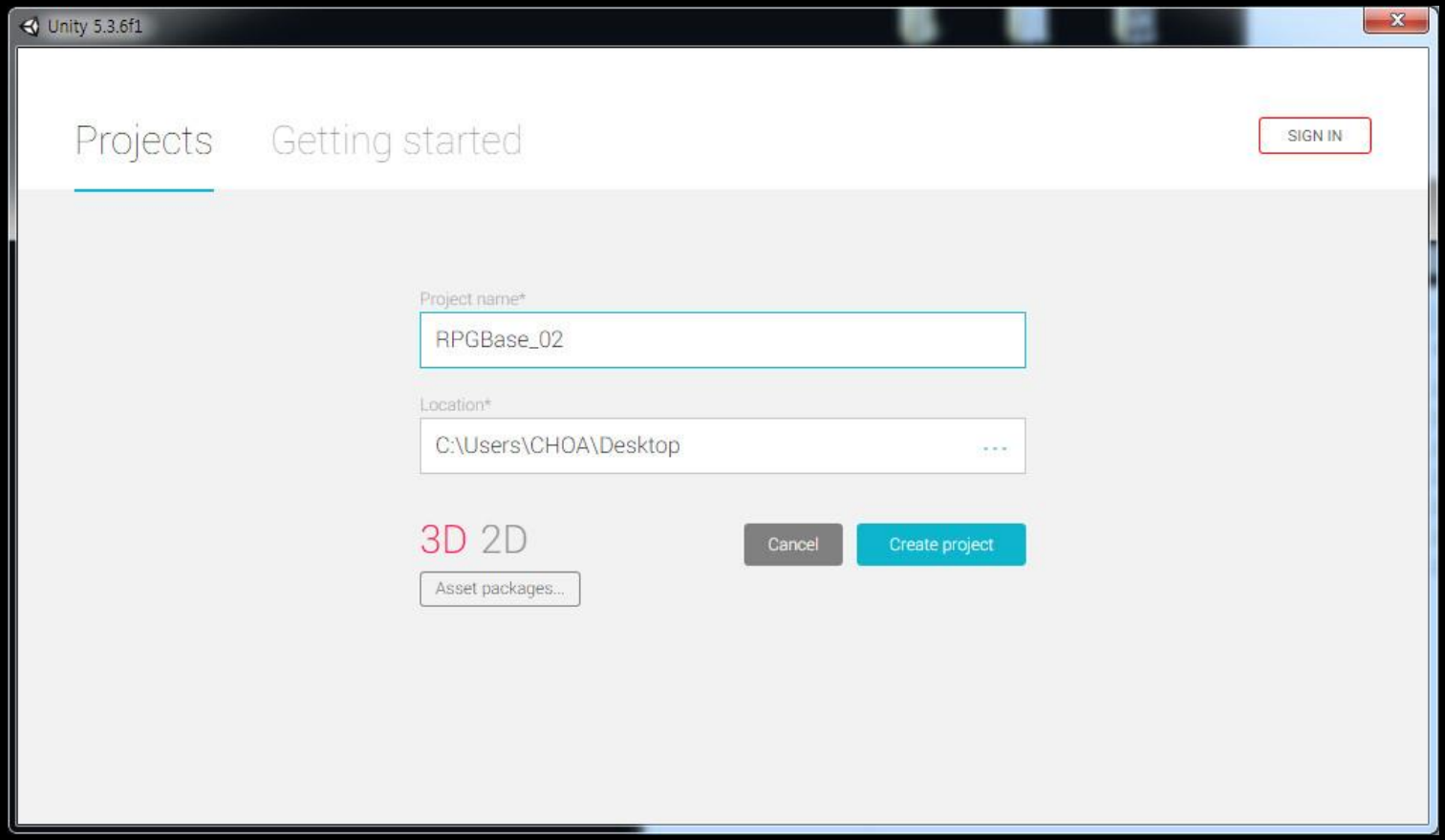
- 게임 제작 준비
- World Map 제작(Cube)
- 캐릭터 생성 및 기본 설정
- 캐릭터 이동

게임 제작 준비

- 프로젝트 생성
- 게임화면 설정
- 카메라 설정
- 조명 설정

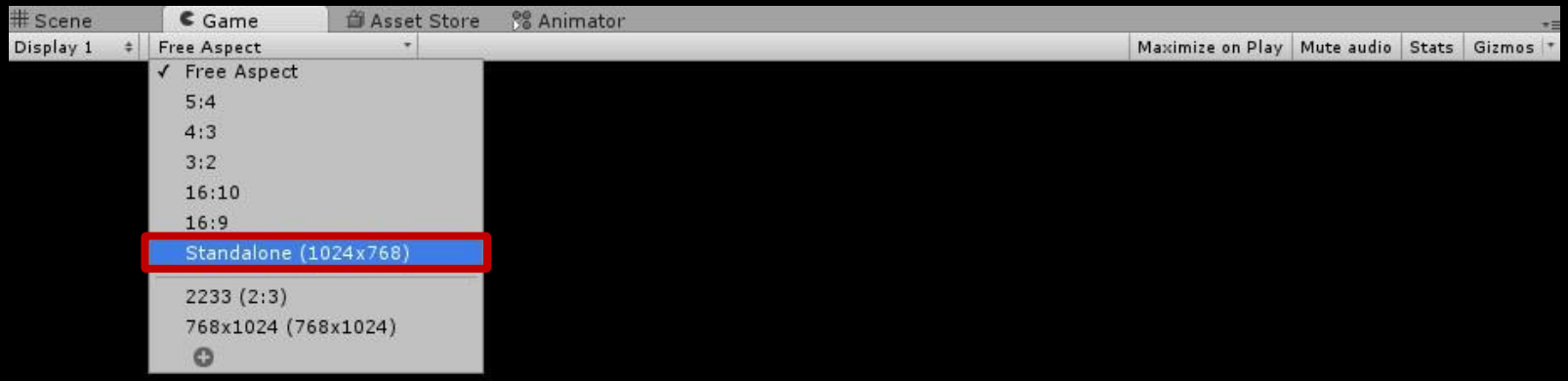
게임 제작 준비

- **프로젝트 생성**
  - 그림과 같이 프로젝트 이름, 위치 등 설정 후 프로젝트 생성



게임 제작 준비

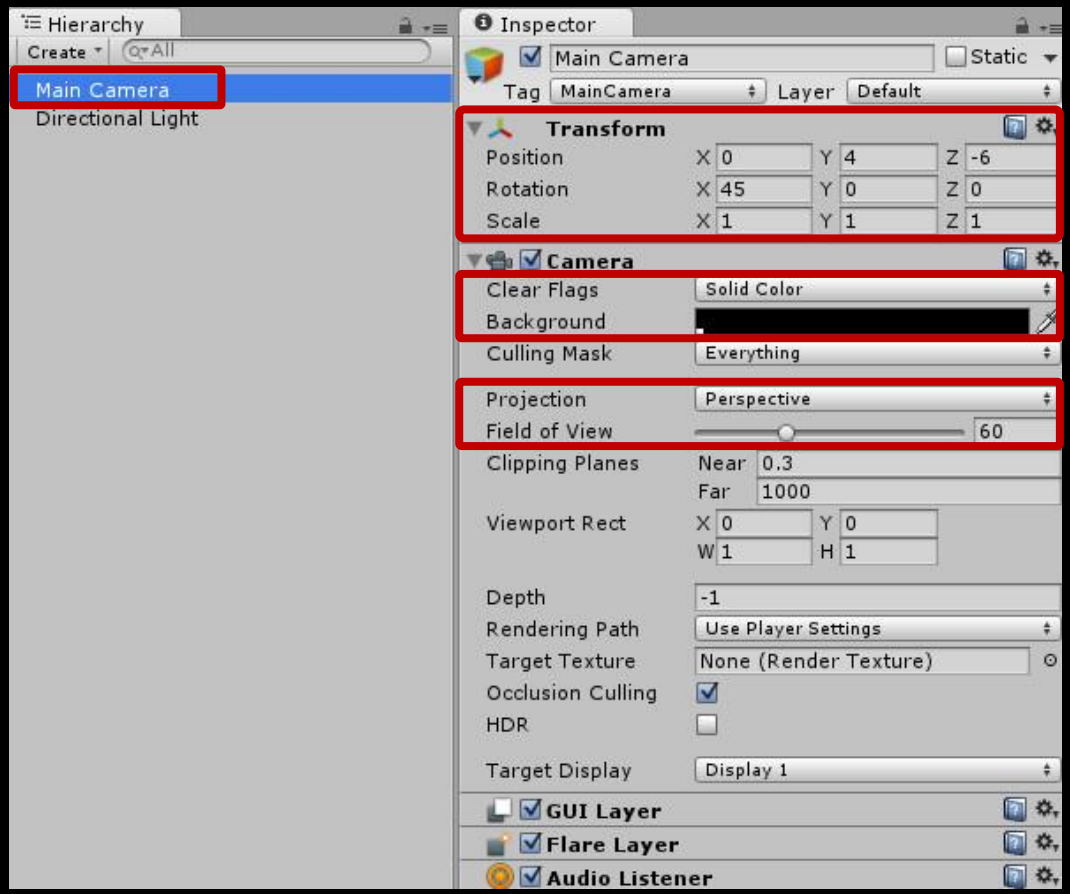
- **게임화면 설정**
  - Game View 바로 아래 Drop List 클릭
  - Standalone (1024x768) 크기의 게임 화면 생성
  - **"+" 버튼을 누르면 원하는 사이즈의 게임화면 추가 가능**



게임 제작 준비

카메라 설정

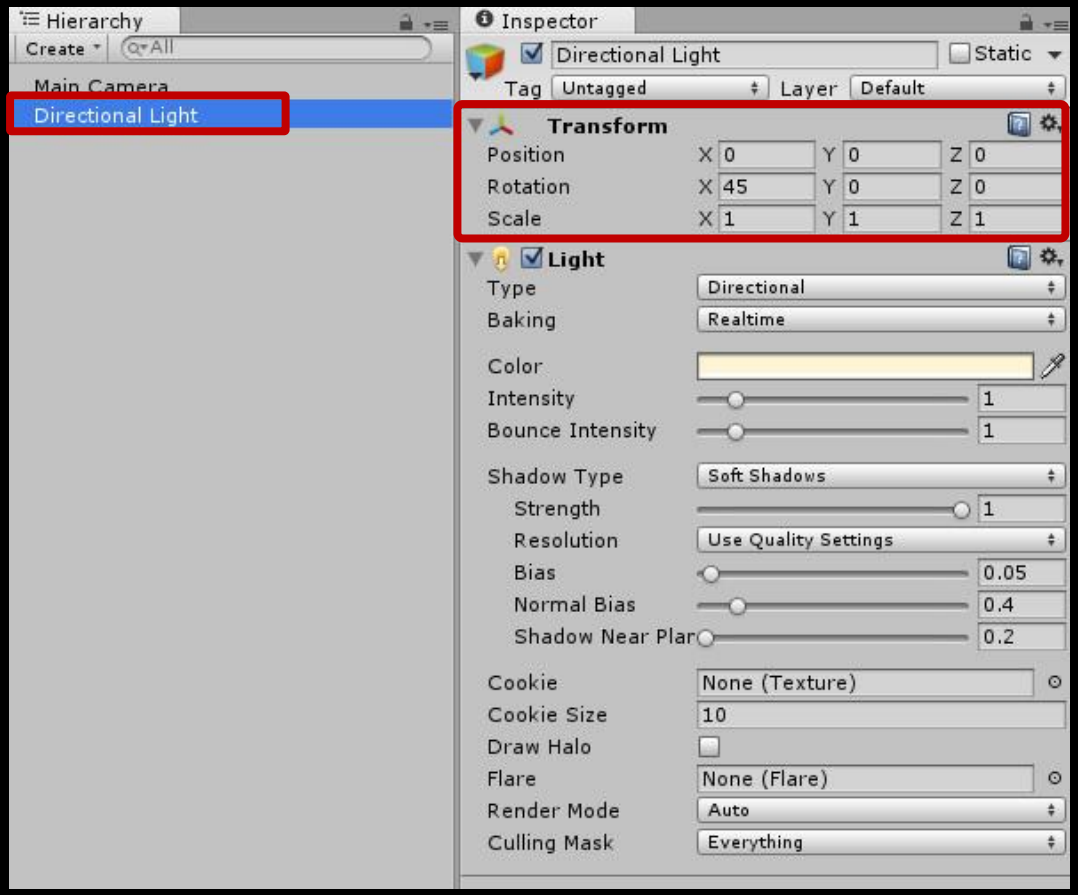
- Transform
  - 위치 값 : Position(0, 4, -6) / 회전 값 : Rotation(45, 0, 0)
- Camera
  - 배경색 설정
    - Clear Flags : Solid Color
    - Background : 0, 0, 0, 255
  - 카메라 모드 3D
    - Projection : Perspective
    - Field of View : 60



게임 제작 준비

조명 설정

- Transform
  - 회전 값 : Rotation(45, 0, 0)

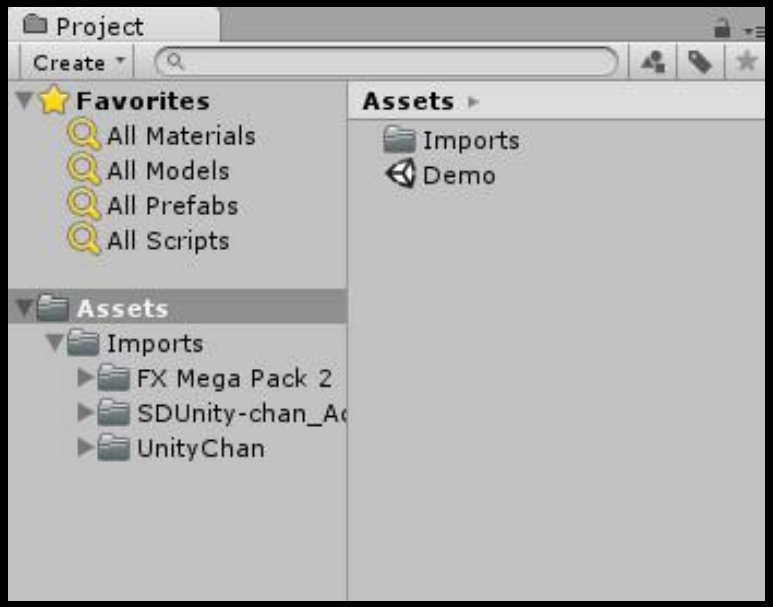


World Map 제작(Cube)

- Package Import
- Map 오브젝트 생성
- Map Material 생성 및 설정
- Map 오브젝트 설정
- 포탈(Portal) 설치
- 결과 화면

World Map 제작(Cube)

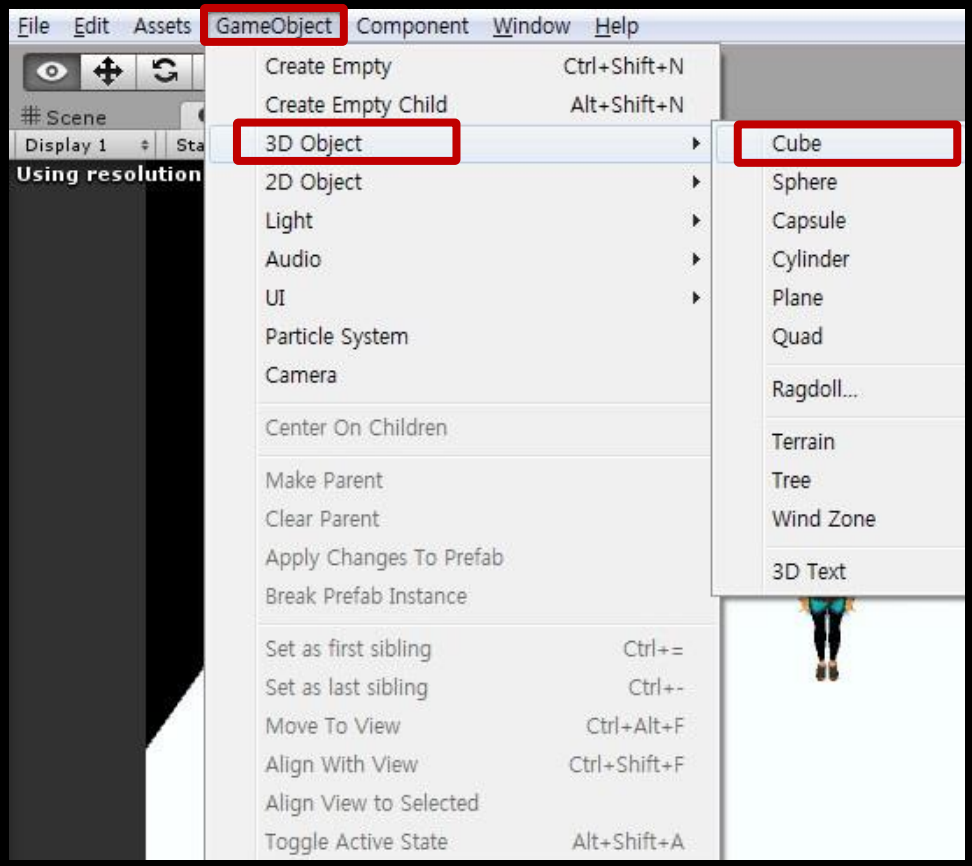
- **Package Import**
  - Project View - 마우스 오른쪽 클릭 - Import Package - Custom Package
  - 제공한 Package를 모두 Import
    - SD_UnityChan-1
    - SD_KohakuChanz-1
    - SDUnity_Action_Pack
    - FX Mega Pack 2

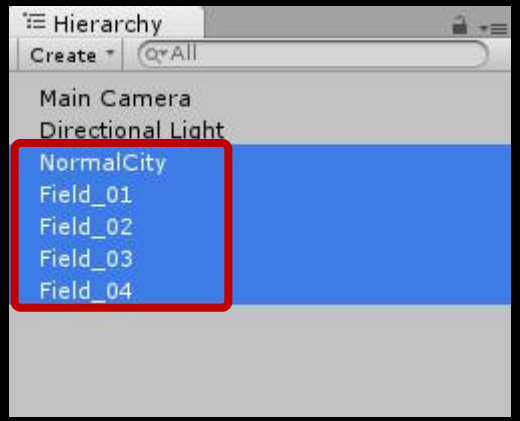


  - Import한 Package들은 따로 폴더에 모아 저장하면 관리가 편리

World Map 제작(Cube)

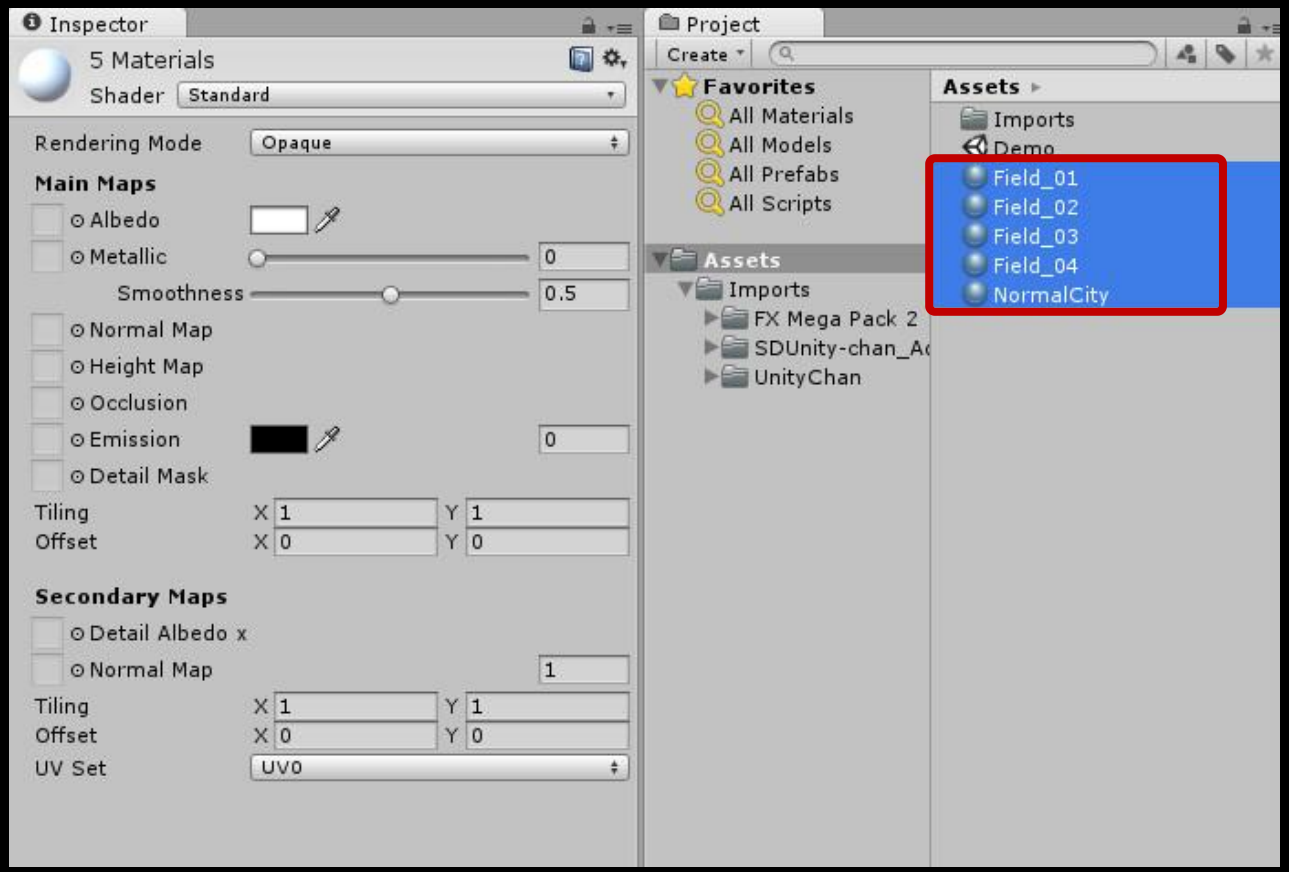
- **Map 오브젝트 생성**
  - GameObject - 3D Object - Cube 생성(5개)
  - Cube 오브젝트의 그림과 같이 변경





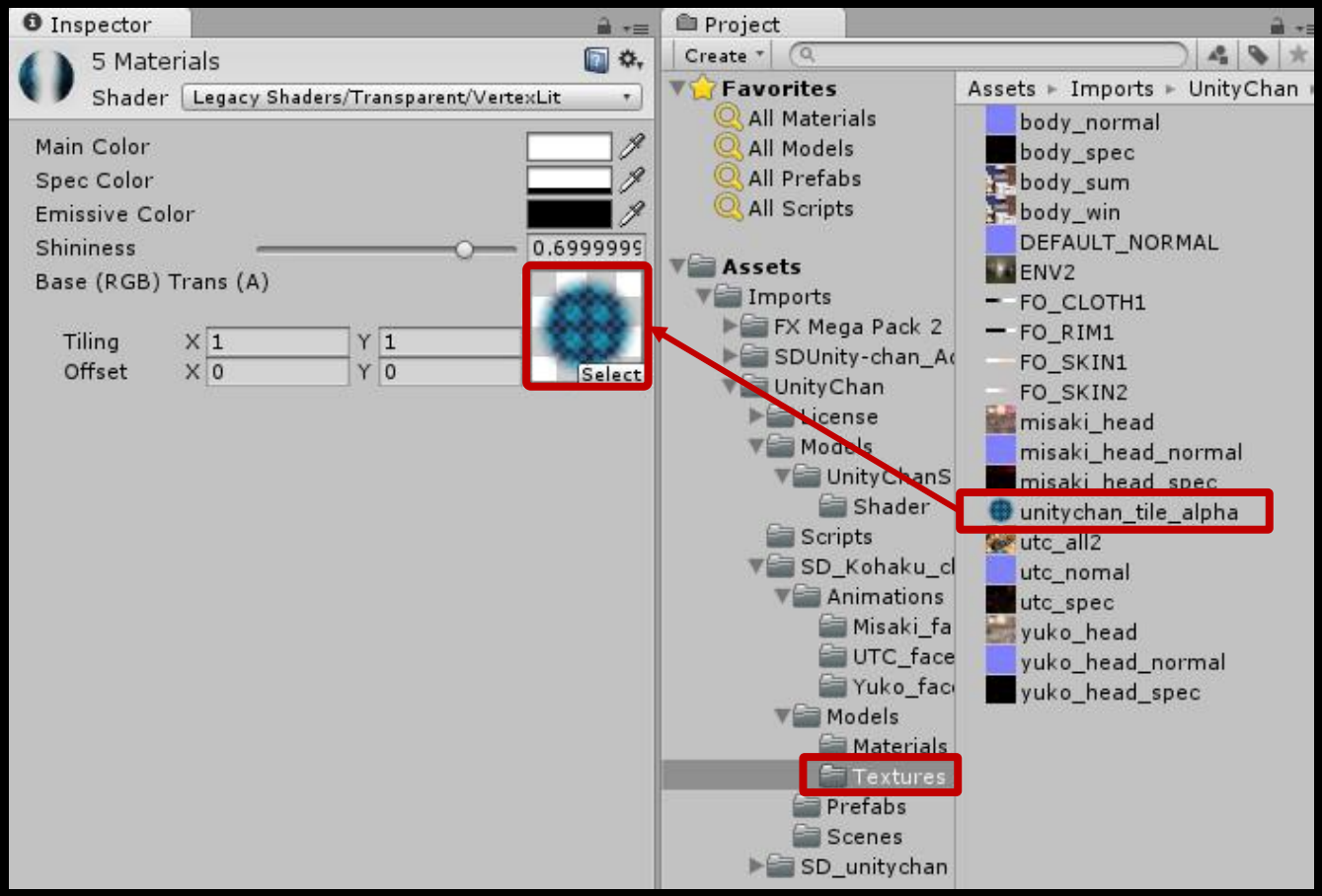
World Map 제작(Cube)

- **Material 생성 및 설정**
  - Project View - 마우스 오른쪽 클릭 - Create - Material(5개)
  - 생성한 5개의 Material의 이름을 그림과 같이 설정



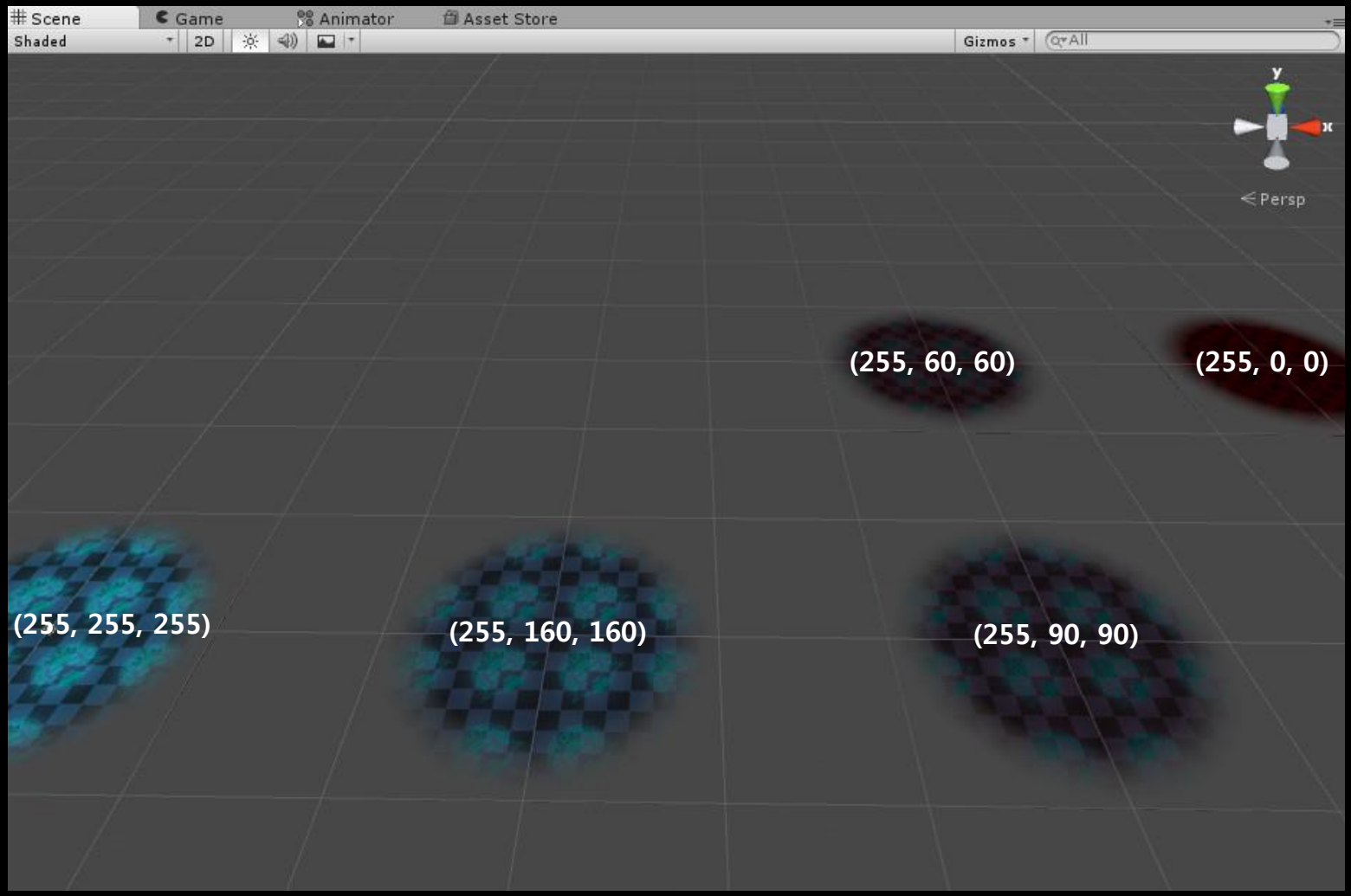
World Map 제작(Cube)

- Shader를 Legacy Shaders - Transparent - VertexLit으로 설정
- UnityChan 모델의 Textures 폴더에 있는 unitychan_tile_alpha를 5개 Material의 Texture로 설정



World Map 제작(Cube)

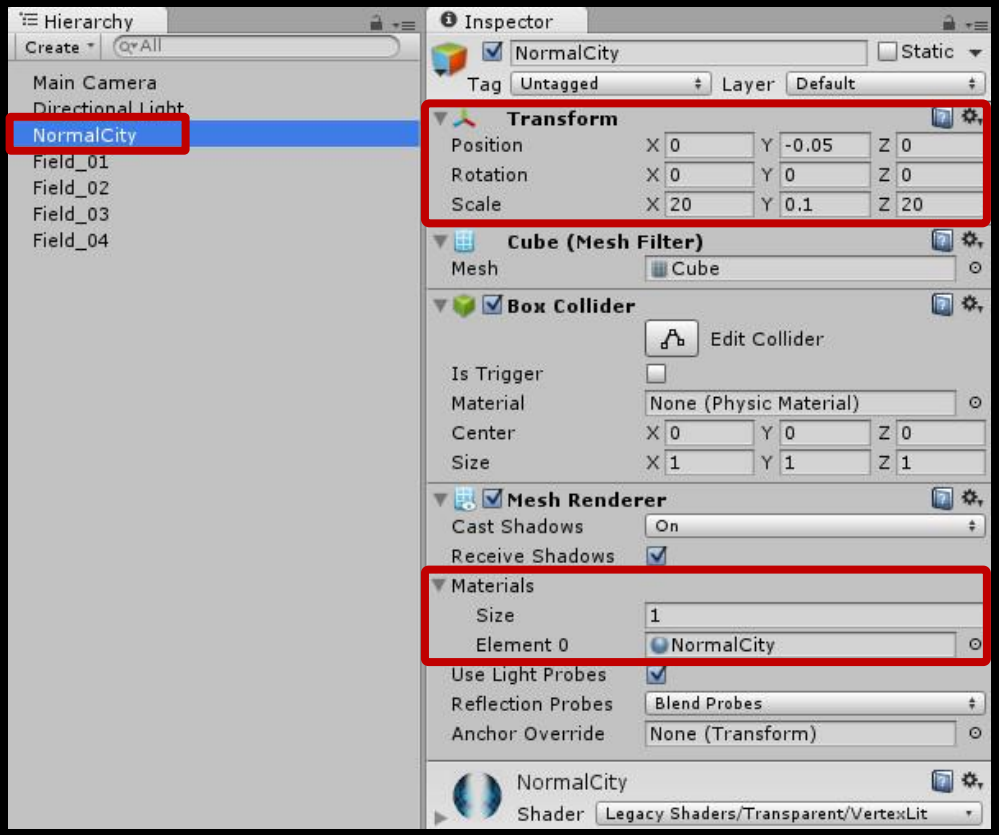
- Field_01 ~ 04 Material의 색깔을 적절히 다르게 적용



World Map 제작(Cube)

맵 오브젝트 설정

- "NormalCity" 오브젝트 설정
  - Transform : Position(0, -0.05, 0) / Scale(20, 0.1, 20)
  - Material을 "NormalCity"로 변경

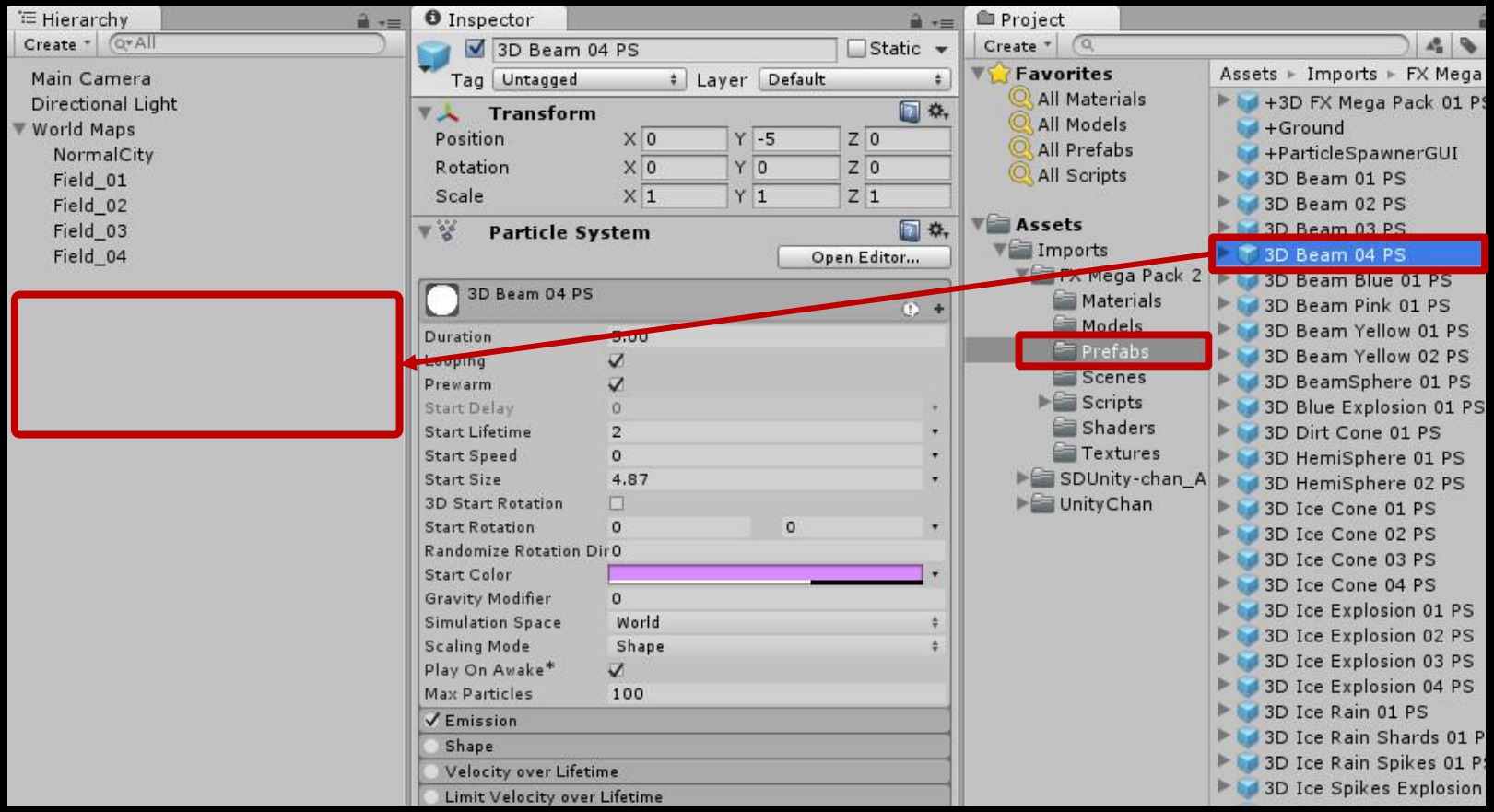


World Map 제작(Cube)

- "Field_0*" 오브젝트들
  - Field_01
    - Transform : Position(30, -0.05, 0) / Scale(20, 0.1, 20)
    - Material을 "Field_01"로 변경
  - Field_02
    - Transform : Position(60, -0.05, 0) / Scale(20, 0.1, 20)
    - Material을 "Field_02"로 변경
  - Field_03
    - Transform : Position(60, -0.05, 30) / Scale(20, 0.1, 20)
    - Material을 "Field_03"로 변경
  - Field_04
    - Transform : Position(90, -0.05, 30) / Scale(20, 0.1, 20)
    - Material을 "Field_04"로 변경

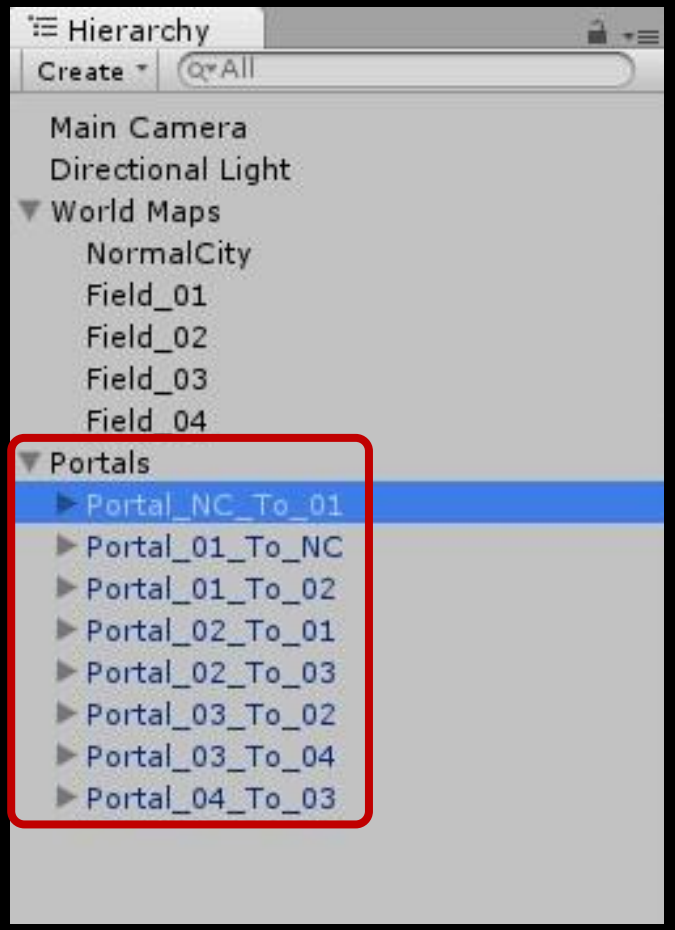
World Map 제작(Cube)

- **포탈(Portal) 설치**
  - FX Mega Pack 2 - Prefabs - 3D Beam 04 PS 생성(8개)



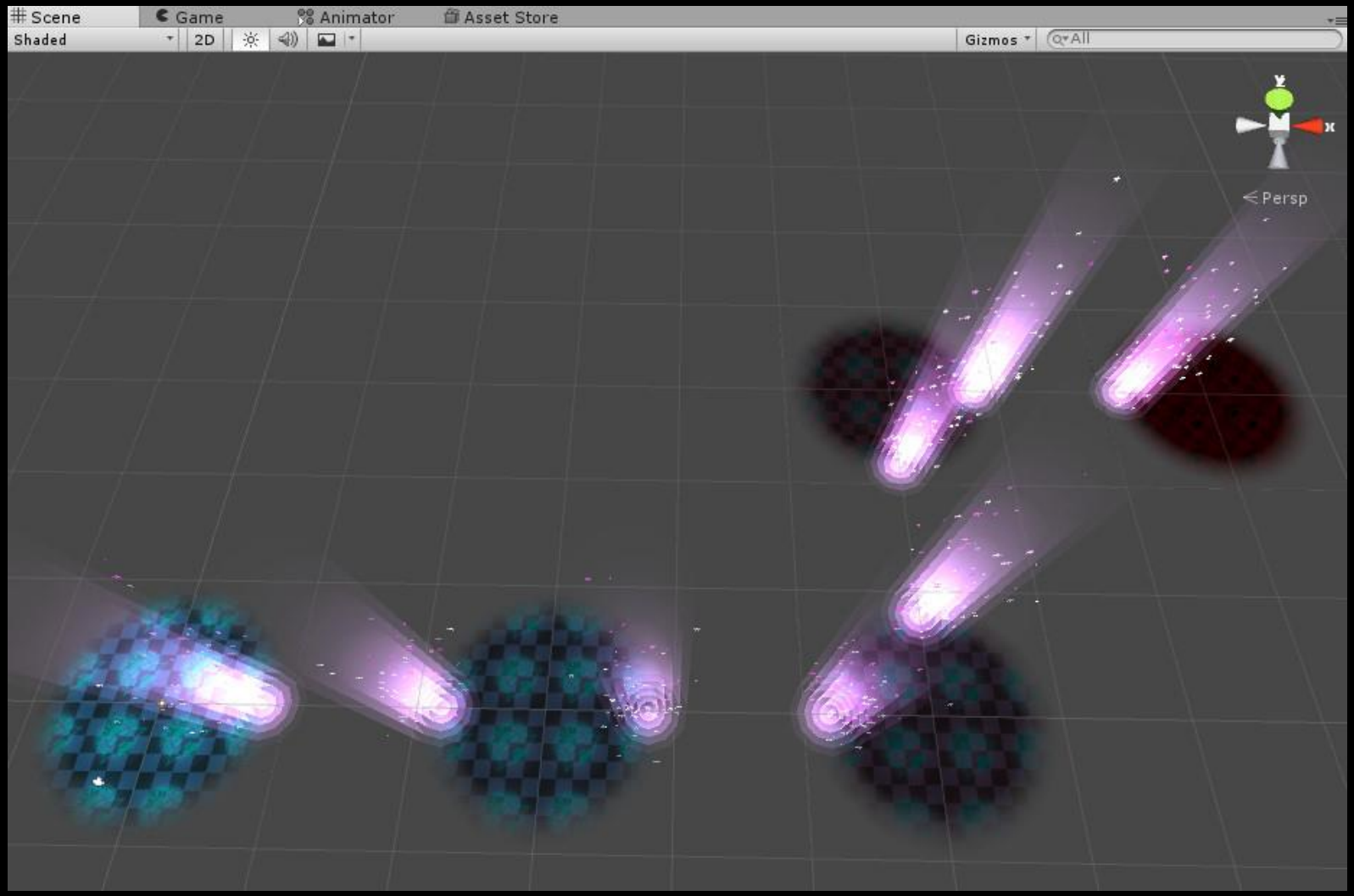
World Map 제작(Cube)

- 각 포탈의 이름을 그림과 같이 설정
  - Portal_NC_To_01 Position(8, 0, 0)
  - Portal_01_To_NC Position(22, 0, 0)
  - Portal_01_To_02 Position(38, 0, 0)
  - Portal_02_To_01 Position(52, 0, 0)
  - Portal_02_To_03 Position(60, 0, 8)
  - Portal_03_To_02 Position(60, 0, 22)
  - Portal_03_To_04 Position(68, 0, 30)
  - Portal_04_To_03 Position(82, 0, 30)



World Map 제작(Cube)

- 결과 화면

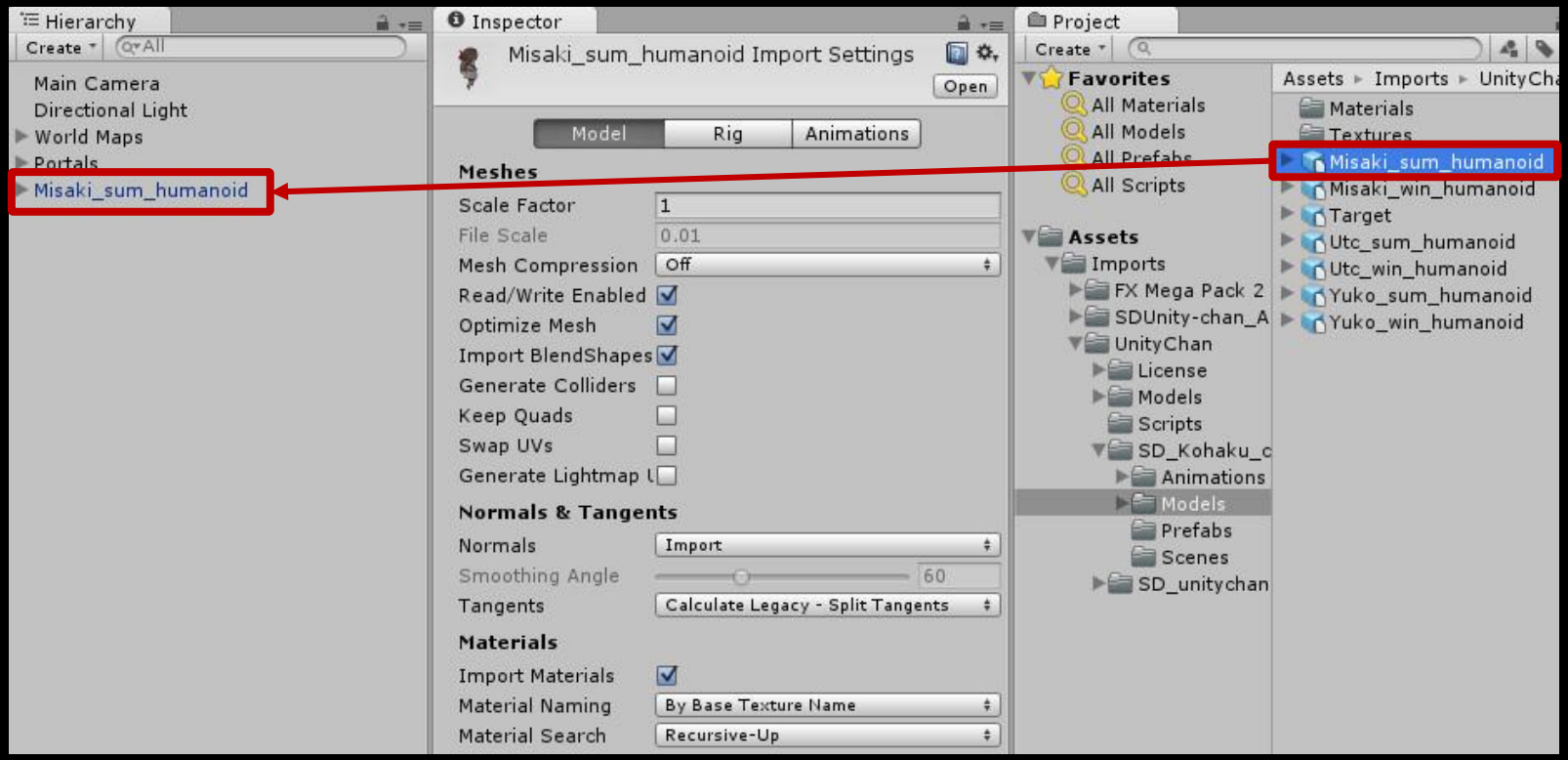


플레이어 캐릭터 생성 및 기본 설정

- 플레이어 Model 생성
- 대기 상태 애니메이션 등록
- Player Class 작성
- 결과 화면

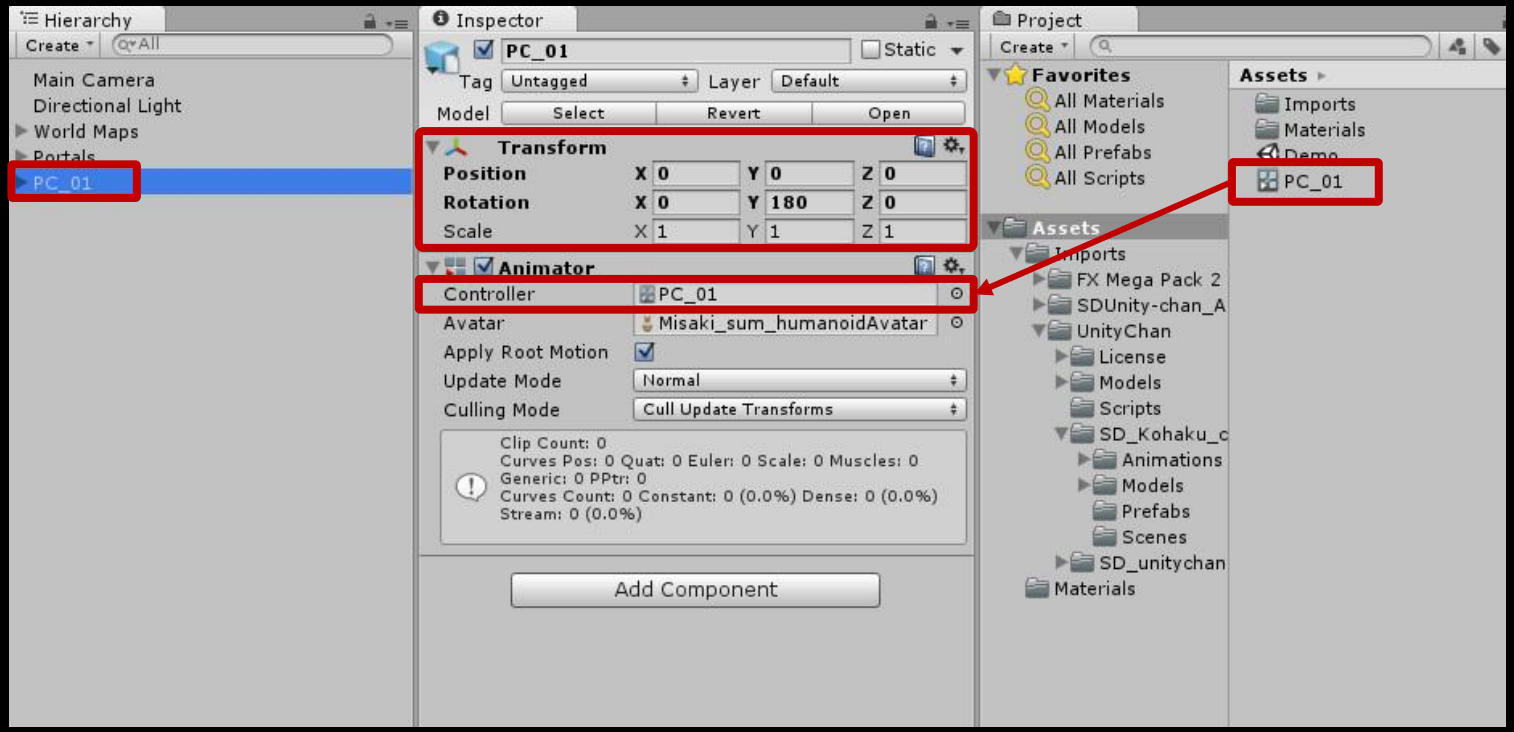
플레이어 캐릭터 생성 및 기본 설정

- **플레이어 Model 생성**
  - UnityChan - SD_Kohaku_chanz - Models – Misaki_sum_humanoid 생성
  - 생성된 오브젝트의 이름을 "PC_01"로 변경



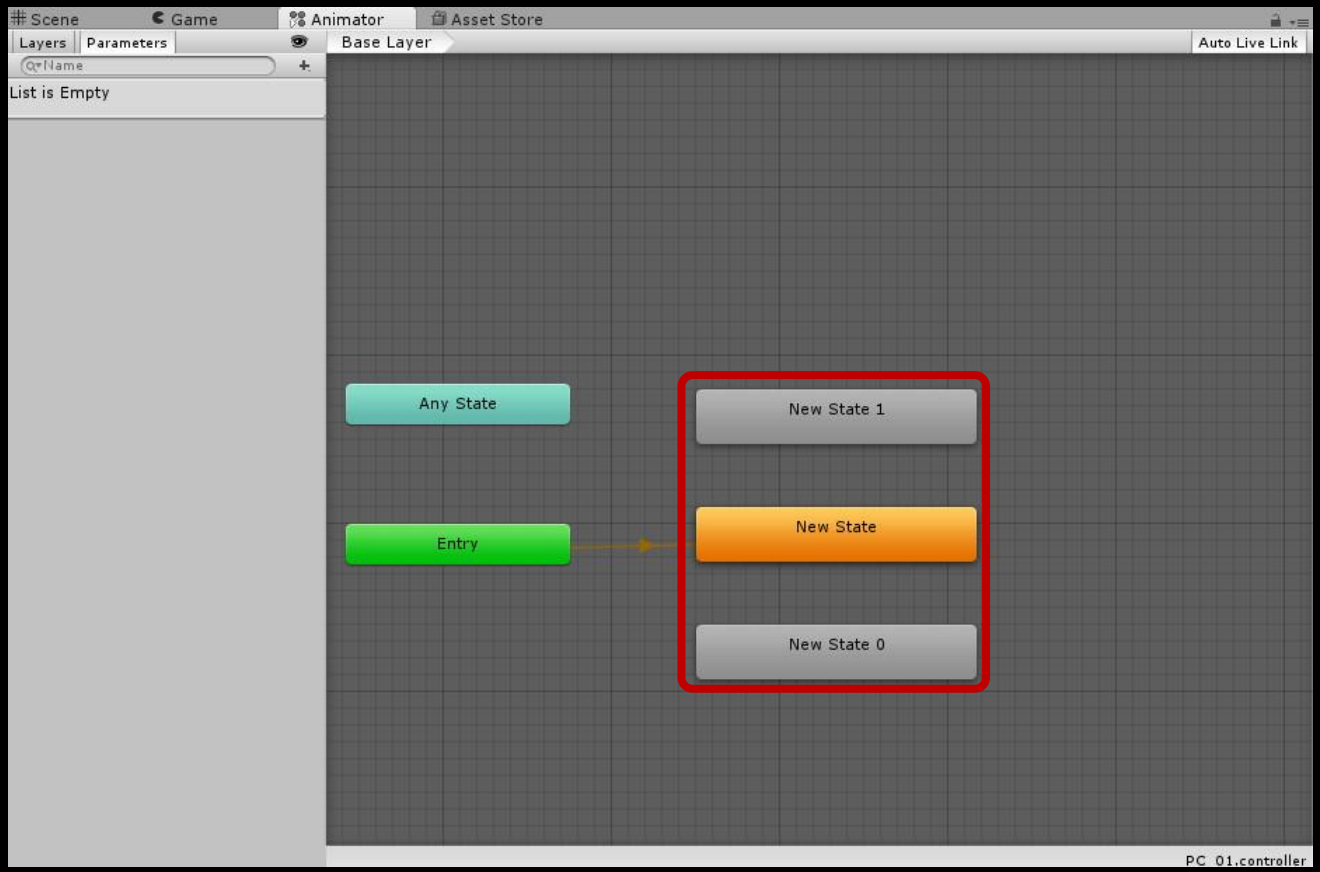
플레이어 캐릭터 생성 및 기본 설정

- Animator Controller 등록
  - Project View - 마우스 오른쪽 클릭 - Create - Animator Controller 생성
  - "PC_01" 오브젝트에 생성한 컨트롤러를 등록



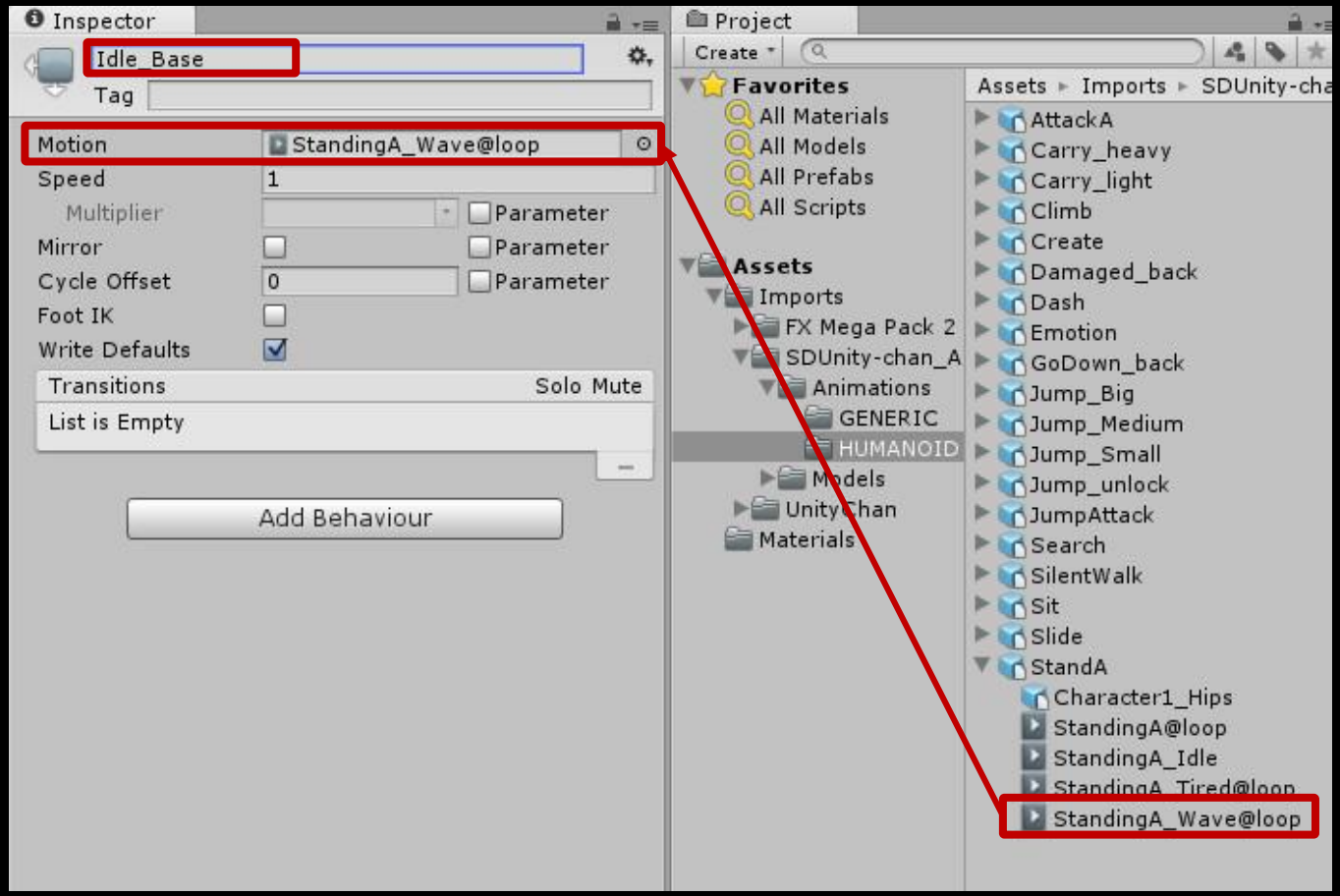
플레이어 캐릭터 생성 및 기본 설정

- 대기 상태 애니메이션 등록
  - 상태 생성
    - Animator View 내부에서 마우스 오른쪽 클릭 - Create State - Empty(3개 생성)



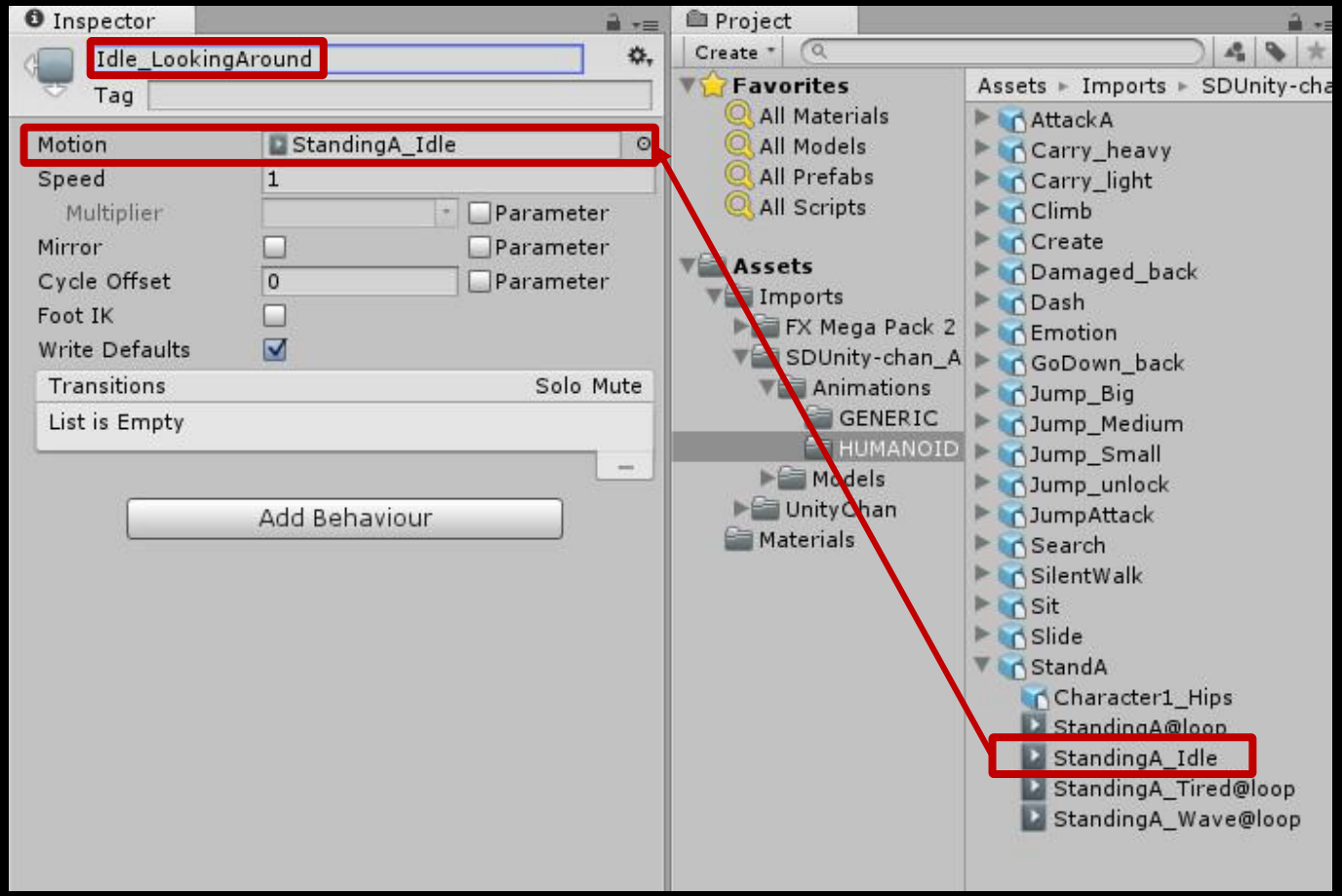
플레이어 캐릭터 생성 및 기본 설정

- 상태 설정 - New State
  - Name : Idle_Base
  - Motion : StandingA_Wave@loop



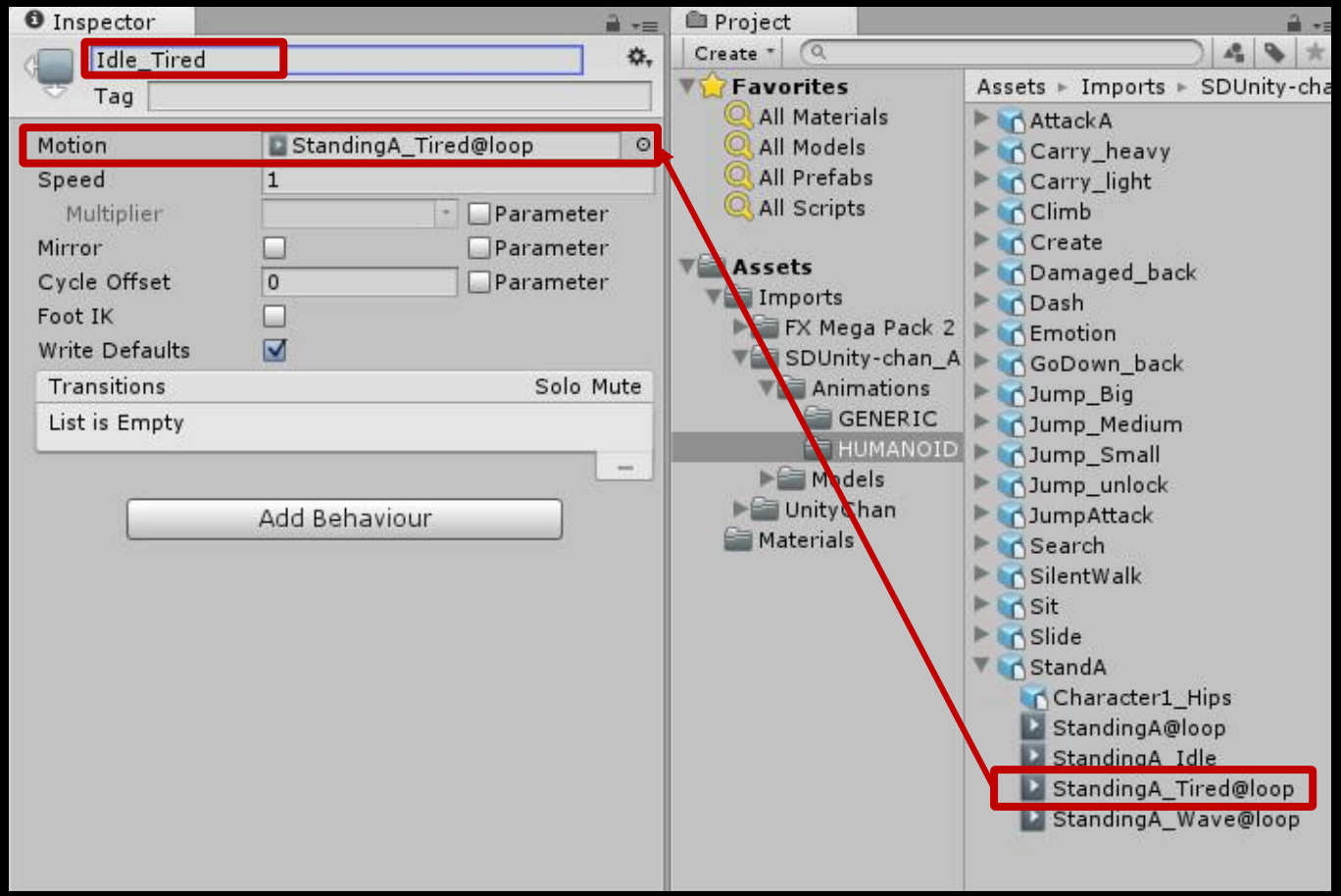
플레이어 캐릭터 생성 및 기본 설정

- 상태 설정 - New State 1
  - Name : Idle_LookingAround
  - Motion : StandingA_Idle



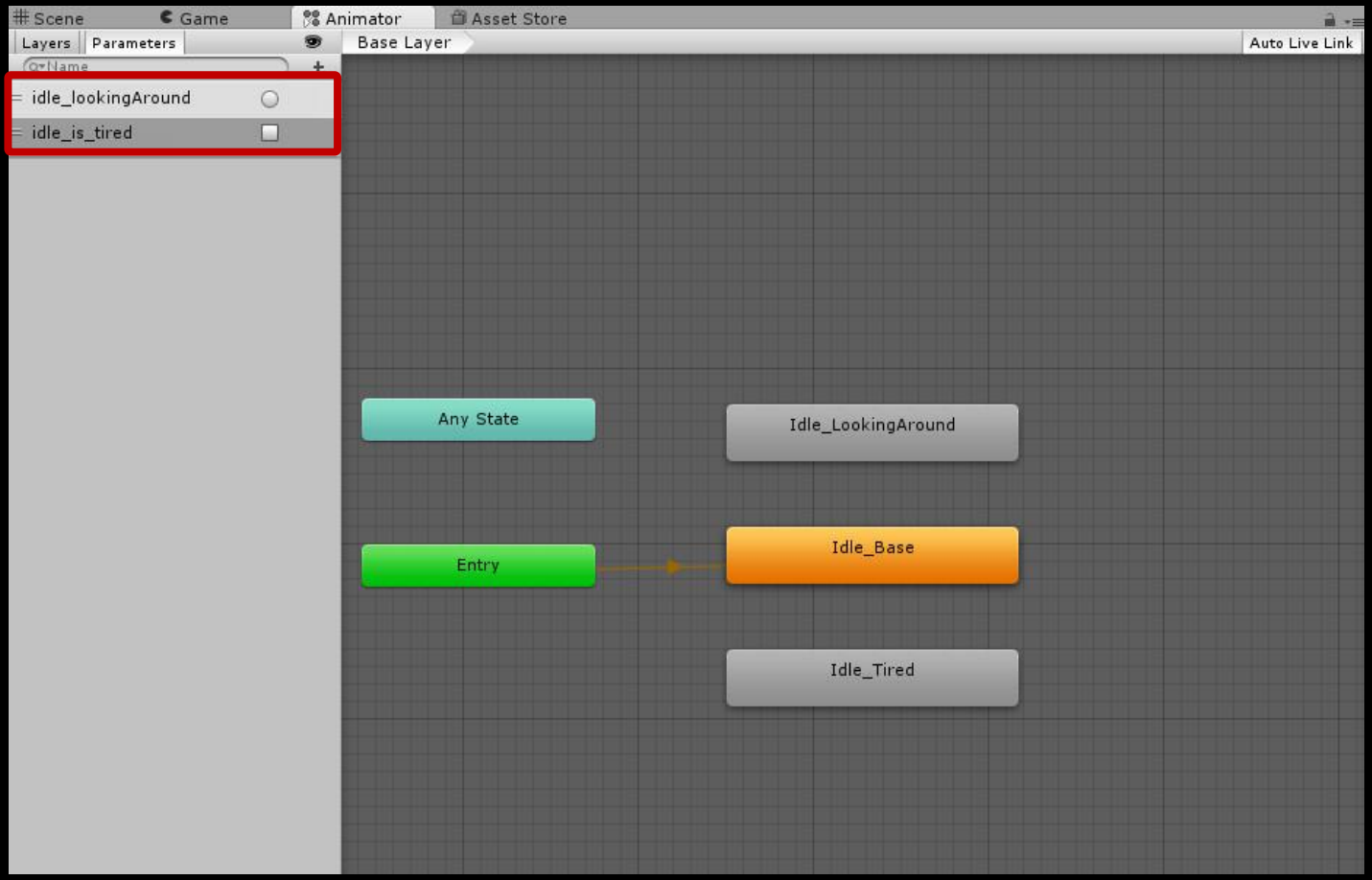
플레이어 캐릭터 생성 및 기본 설정

- 상태 설정 - New State 0
  - Name : Idle_Tired
  - Motion : StandingA_Tired@loop



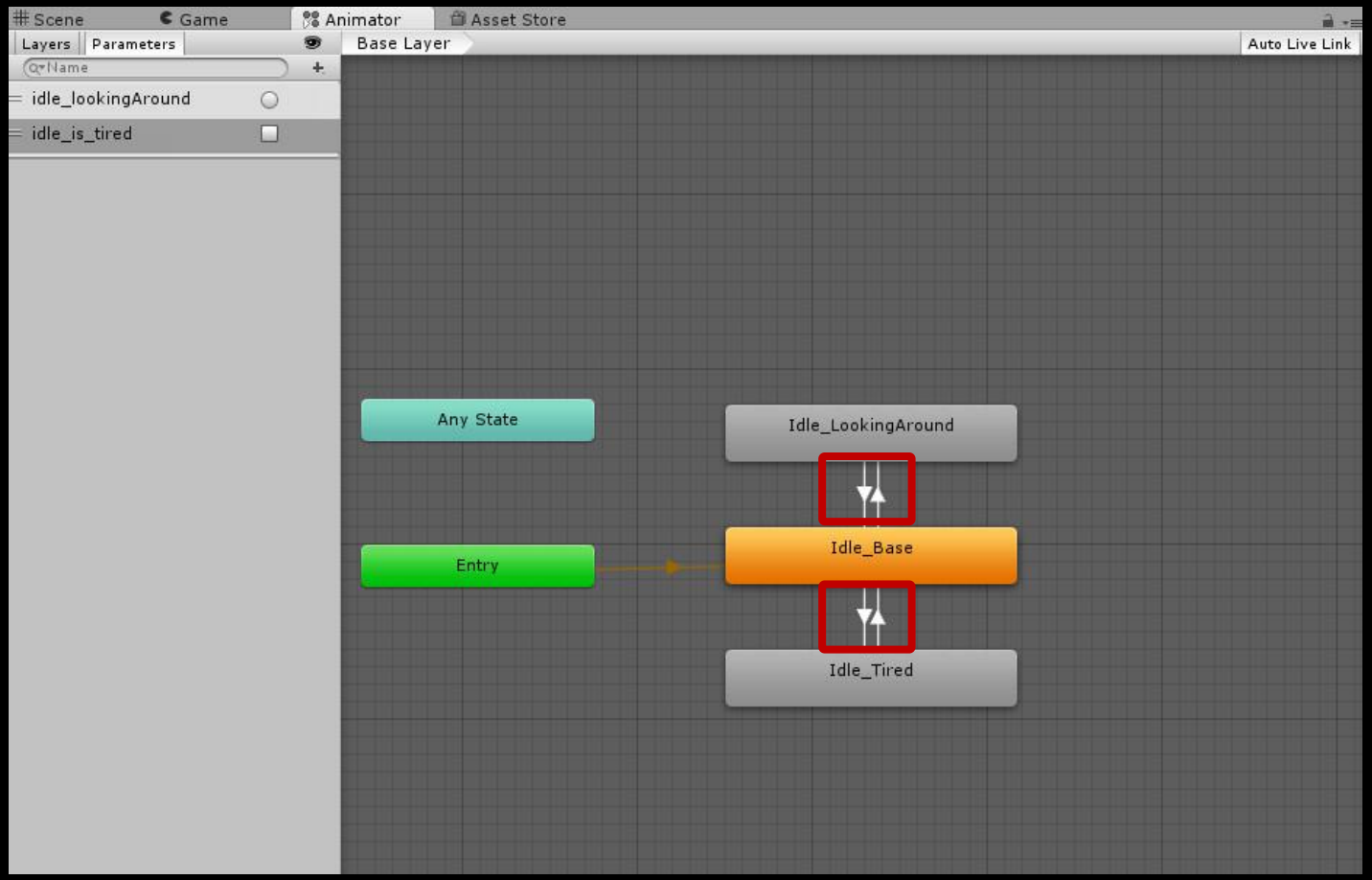
플레이어 캐릭터 생성 및 기본 설정

- Parameter 생성
  - Trigger Type : idle_lookingAround
  - Bool Type : idle_is_tired



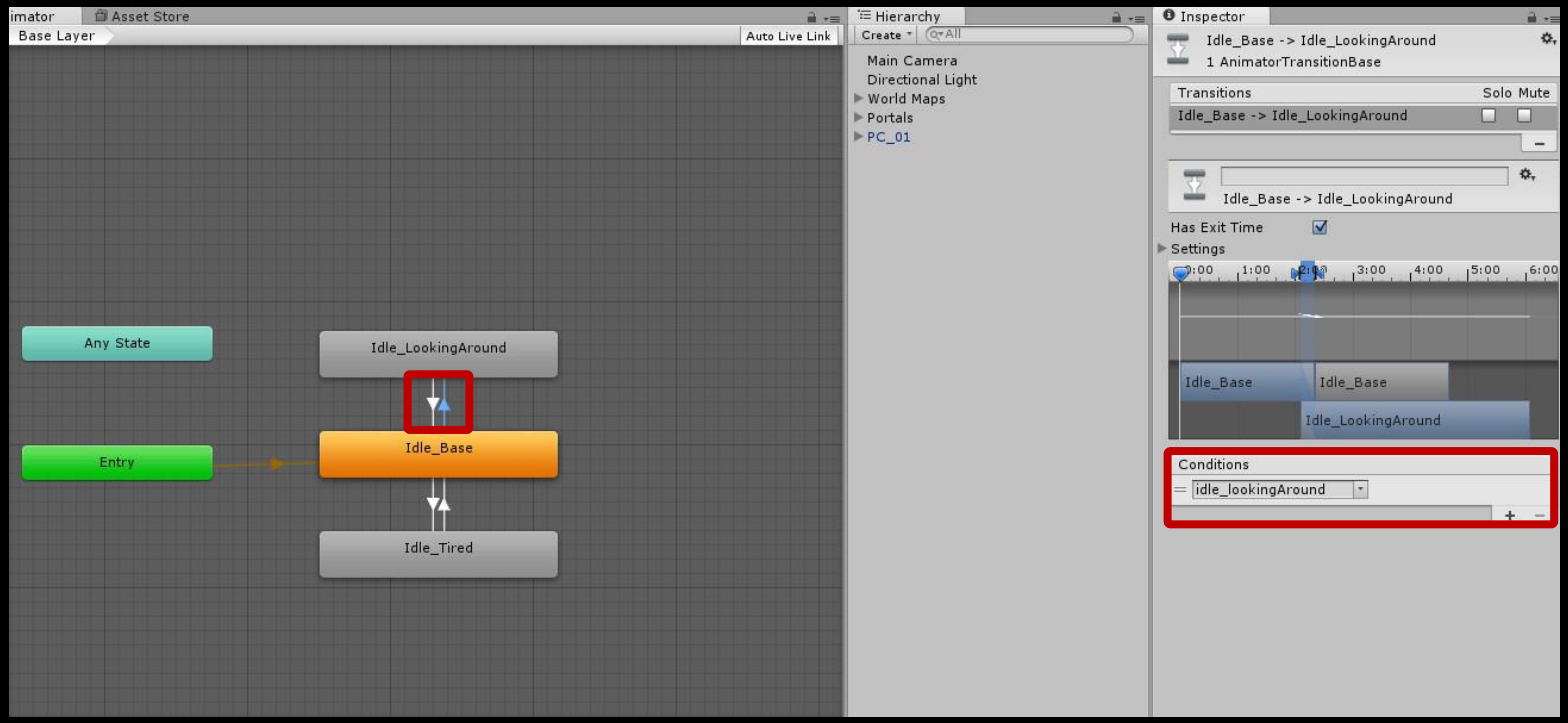
플레이어 캐릭터 생성 및 기본 설정

- Transition 생성
  - Idle_Base ↔ Idle_LookingAround 간 상태를 변경할 수 있는 Transition 생성
  - Idle_Base ↔ Idle_is_tired 간 상태를 변경할 수 있는 Transition 생성



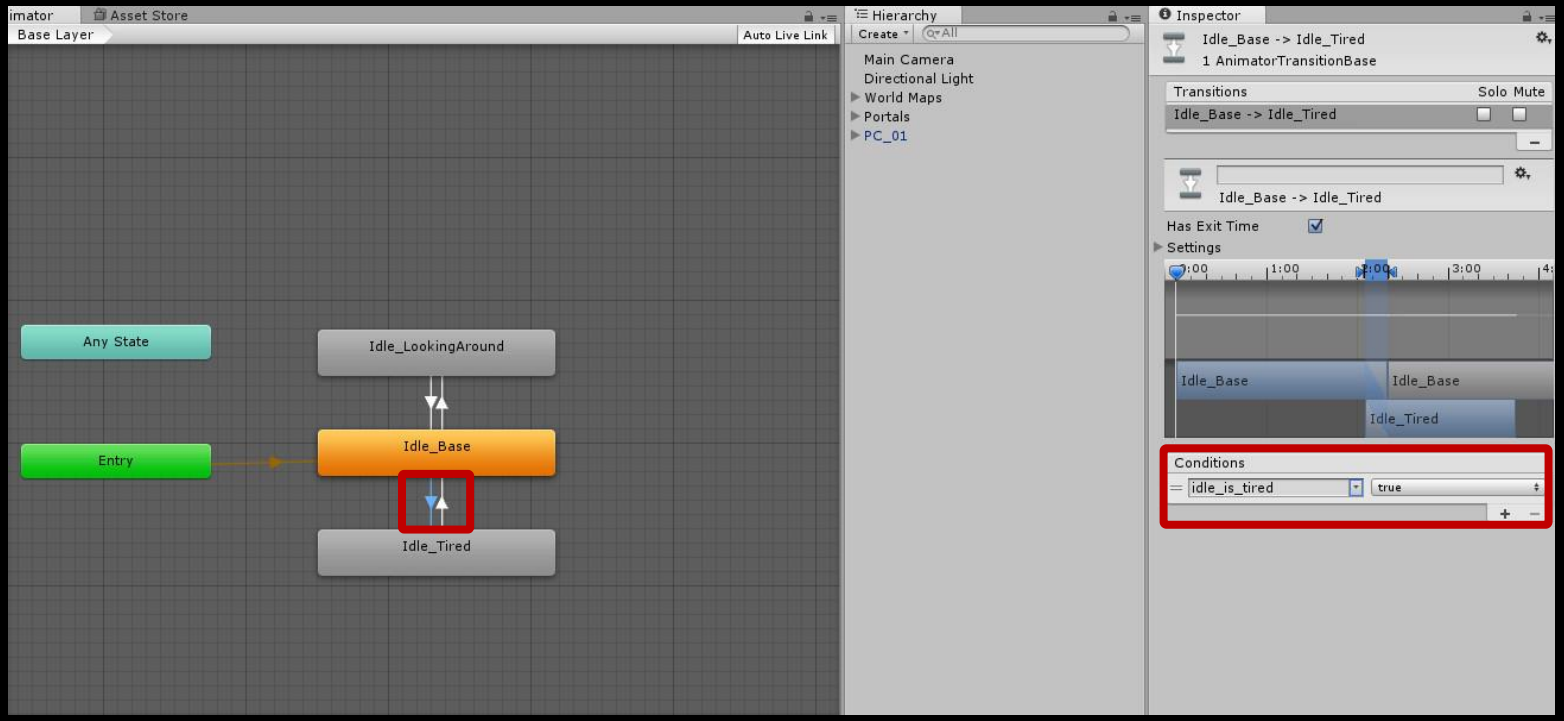
플레이어 캐릭터 생성 및 기본 설정

- Transition 설정
  - Idle_Base → Idle_LookingAround 상태로 변경하는 조건으로 idle_lookingAround Trigger Parameter 사용
  - Idle_LookingAround는 1회 재생하는 상태이기 때문에 Idle_Base로 돌아가는 Transition은 아무런 설정도 하지 않음



플레이어 캐릭터 생성 및 기본 설정

- Idle_Base → Idle_Tired 상태로 변경하는 조건은
  "idle_is_tired" Parameter가 true 일 때
- Idle_Tired → Idle_Base 상태로 변경하는 조건은
  "idle_is_tired" Parameter가 false 일 때



플레이어 캐릭터 생성 및 기본 설정

- Player Class 작성

```
1  using UnityEngine;
2  using System.Collections;
3
4  public enum PLAYER_STATE { IDLE=0, }
5
6  public class Player : MonoBehaviour
7  {
8      private Animator        anim;
9      private PLAYER_STATE    player_state;
10     private float           idle_time;
11
12     void Awake()
13     {
14         anim            = GetComponent<Animator>();
15         player_state    = PLAYER_STATE.IDLE;
16         idle_time       = .0f;
17     }
18     void Update()
19     {
20         Update_Actions();
21     }
22     void Update_Actions()...
44 }
```

플레이어 캐릭터 생성 및 기본 설정

- Player Class - Update_Actions()

```
22      void Update_Actions()
23      {
24          switch ( player_state )
25          {
26              case PLAYER_STATE.IDLE:
27                  if ( anim.GetCurrentAnimatorStateInfo(0).IsName("Idle_Base") )
28                  {
29                      if ( idle_time < 5.0f ) idle_time += Time.deltaTime;
30                      else
31                      {
32                          idle_time = .0f;
33                          anim.SetTrigger("idle_lookingAround");
34                      }
35                  }
36
37                  if ( Input.GetKeyDown(KeyCode.P) )
38                  {
39                      anim.SetBool("idle_is_tired", !anim.GetBool("idle_is_tired"));
40                  }
41                  break;
42          }
43      }
44  }
```

기본 상태(Idle_Base)일 때만 일정 시간마다
두리번 거리는 상태를 1회씩 재생

현재는 체력 시스템이 없기 때문에 P 키를 이용해
기본 상태/피곤한 상태를 변경

플레이어 캐릭터 생성 및 기본 설정

- 결과 화면

캐릭터 이동

- 캐릭터 이동
- 이동 방향 바라보기
- 이동 애니메이션 처리
- 카메라 이동 및 회전
- 포탈을 이용한 필드 이동

캐릭터 이동

캐릭터 이동

- Player Class - 마우스 오른쪽 클릭 시 Mouse Picking을 이용해 이동

```
using UnityEngine;
using System.Collections;

public enum PLAYER_STATE { IDLE=0, MOVE }

public class Player : MonoBehaviour
{
    private Animator        anim;
    private PLAYER_STATE    player_state;
    private float           idle_time;

    private float           move_speed;
    private Vector3         goal_pos;

    void Awake()
    {
        anim            = GetComponent<Animator>();
        player_state    = PLAYER_STATE.IDLE;
        idle_time       = .0f;

        move_speed      = 3.0f;
        goal_pos        = Vector3.zero;
    }
    void Update()
    {
        Update_Inputs();
        Update_Actions();
    }
    void Update_Inputs()...
    void Update_Actions()...
    public void     Add_Pos(Vector3 p)  { transform.position += p; }
    public void     Set_Pos(Vector3 p)  { transform.position = p; }
    public Vector3  Get_Pos()           { return transform.position; }
}
```

캐릭터 이동

- Player Class - Update_Inputs()

```
29      void Update_Inputs()
30      {
31          Ray         ray;
32          RaycastHit  hit;
33
34          if ( Input.GetMouseButtonDown(1) )
35          {
36              ray = Camera.main.ScreenPointToRay(Input.mousePosition);
37              if ( Physics.Raycast(ray, out hit) )
38              {
39                  player_state    = PLAYER_STATE.MOVE;
40                  goal_pos        = hit.point;
41              }
42          }
43      }
```

마우스 오른쪽 클릭을 하게 되면 카메라로부터 마우스 좌표로 광선을 쏴 부딪힌 오브젝트의 세부 좌표를 hit.point를 통해 받아옴

캐릭터 이동

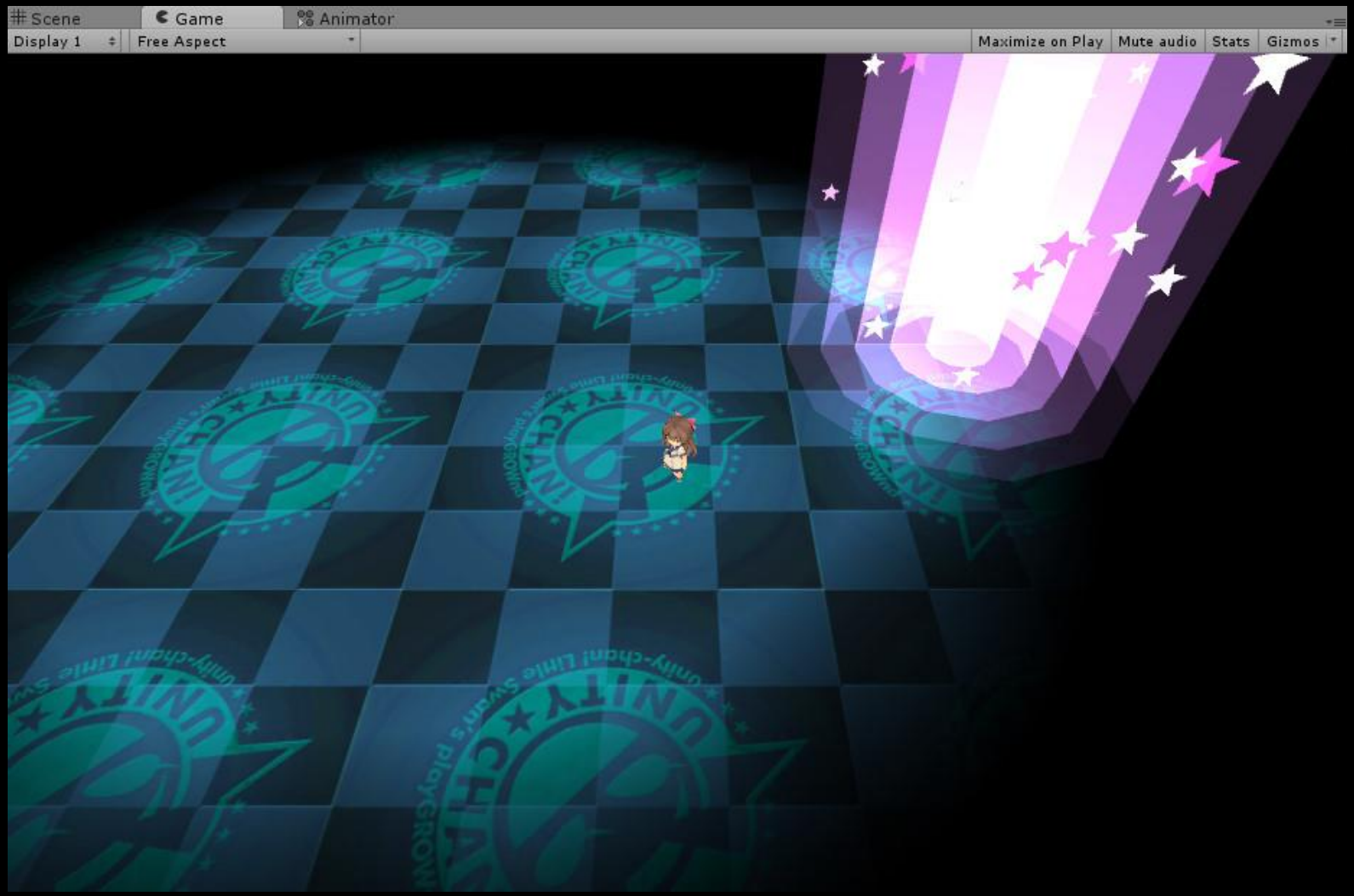
- Player Class - Update_Actions()

```
44  void Update_Actions()
45  {
46      switch ( player_state )
47      {
48          PLAYER_STATE.IDLE
66          #region PLAYER_STATE.MOVE
67          case PLAYER_STATE.MOVE:
68              Vector3 move_pos = Vector3.zero;
69              if ( Vector3.Distance(goal_pos, Get_Pos()) > .1f )
70                  move_pos = Vector3.Normalize(goal_pos-Get_Pos());
71              else
72              {
73                  Set_Pos(goal_pos);
74                  player_state = PLAYER_STATE.IDLE;
75              }
76              Add_Pos(move_pos * move_speed * Time.deltaTime);
77              break;
78          #endregion
79      }
80  }
```

현재 좌표와 목표 좌표 사이의 거리가 멀면 이동하고, 가까우면 상태를 대기(Idle)로 변경

캐릭터 이동

- 결과 화면


Scene
Game
Animator
Display 1
Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos

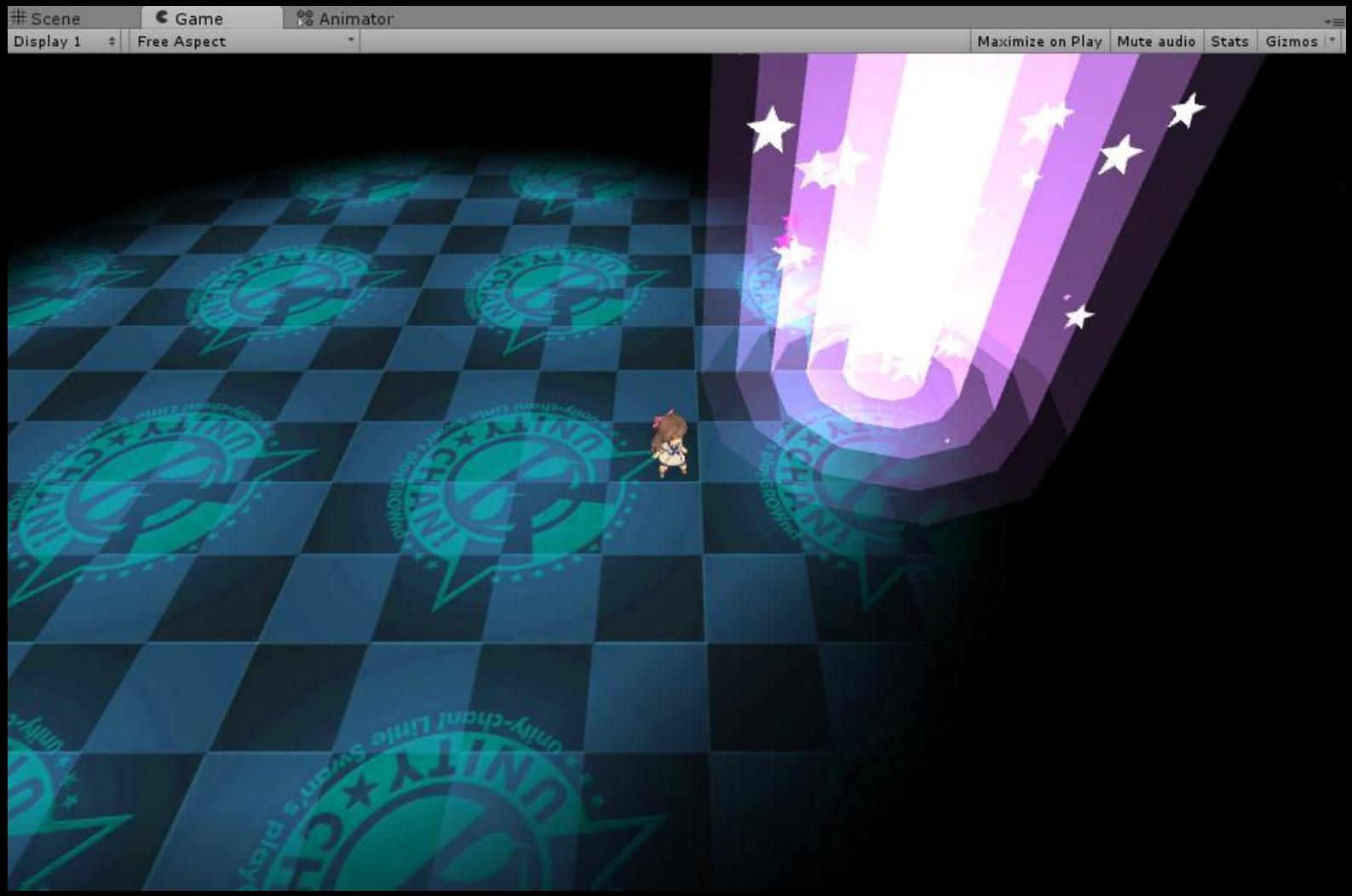
캐릭터 이동

- 이동 방향 바라보기
  - Player Class - Update_Inputs()

```
29      void Update_Inputs()
30      {
31          Ray         ray;
32          RaycastHit  hit;
33
34          if ( Input.GetMouseButtonDown(1) )
35          {
36              ray = Camera.main.ScreenPointToRay(Input.mousePosition);
37              if ( Physics.Raycast(ray, out hit) )
38              {
39                  player_state    = PLAYER_STATE.MOVE;
40                  goal_pos        = hit.point;
41
42                  transform.localRotation = Quaternion.LookRotation(goal_pos-Get_Pos());
43              }
44          }
45      }
```

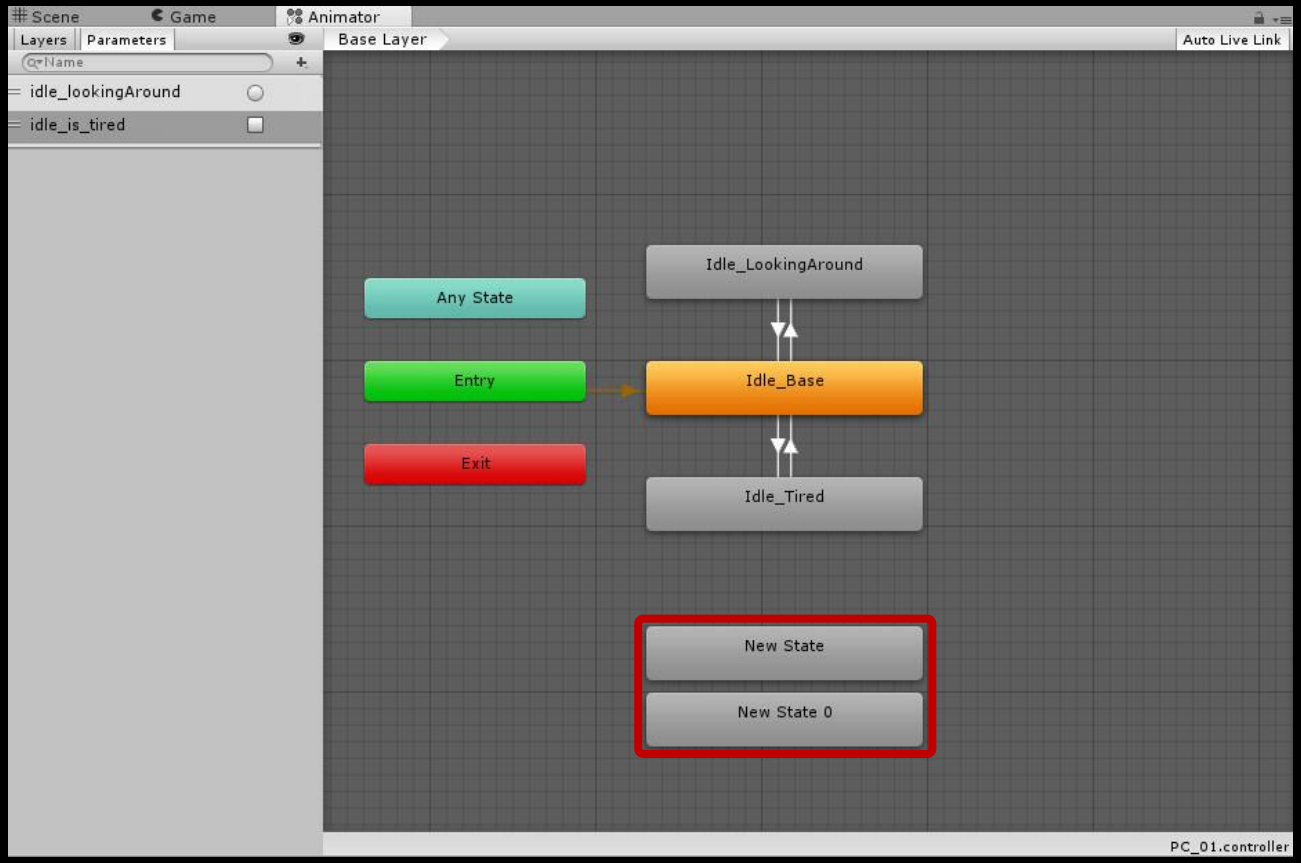
캐릭터 이동

- 결과 화면


Scene
Game
Animator
Display 1
Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos

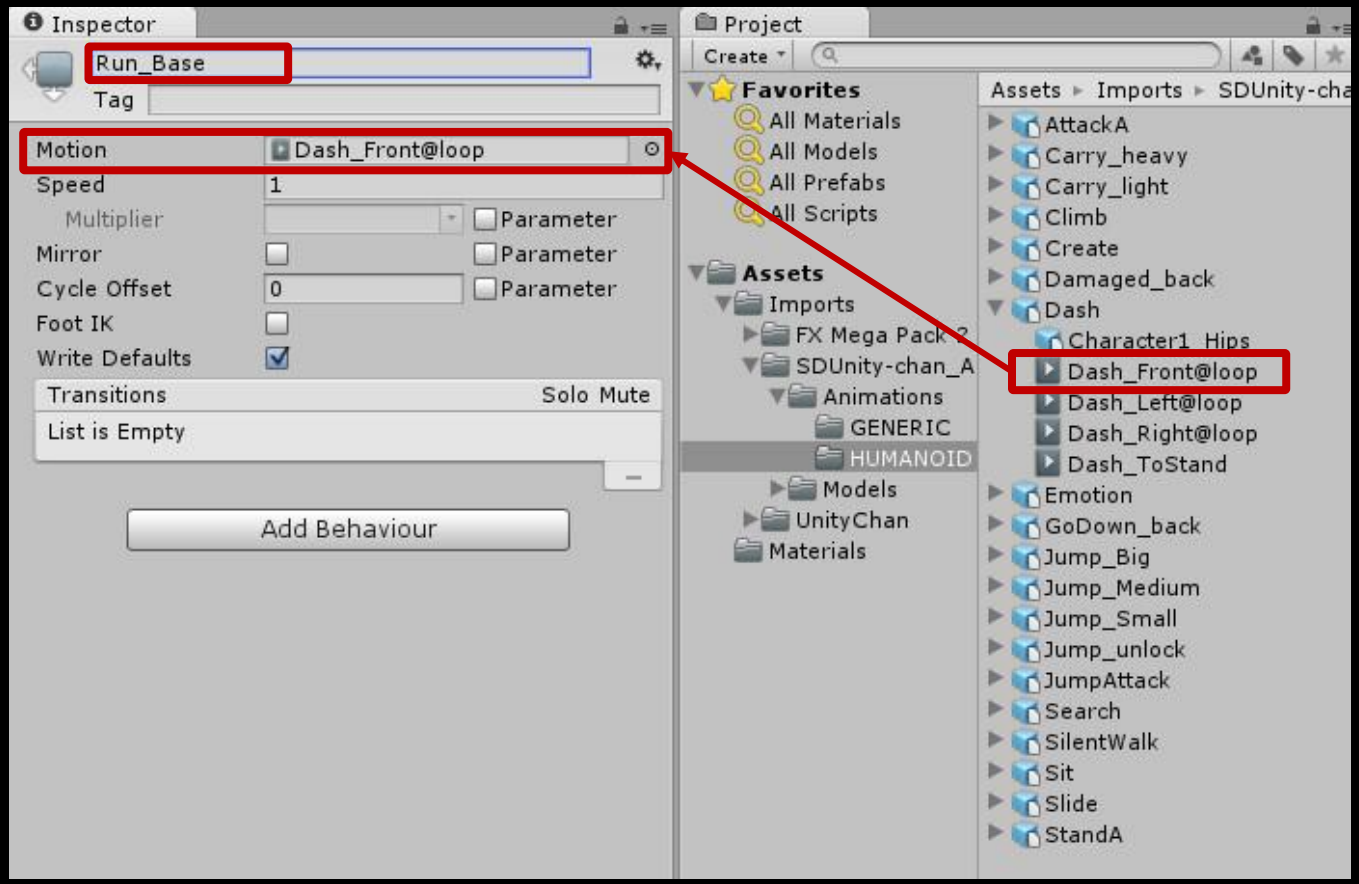
캐릭터 이동

- 이동 애니메이션 처리
  - 상태 생성
    - Animator View - 마우스 오른쪽 클릭 - Create State - Empty로 2개의 상태 생성



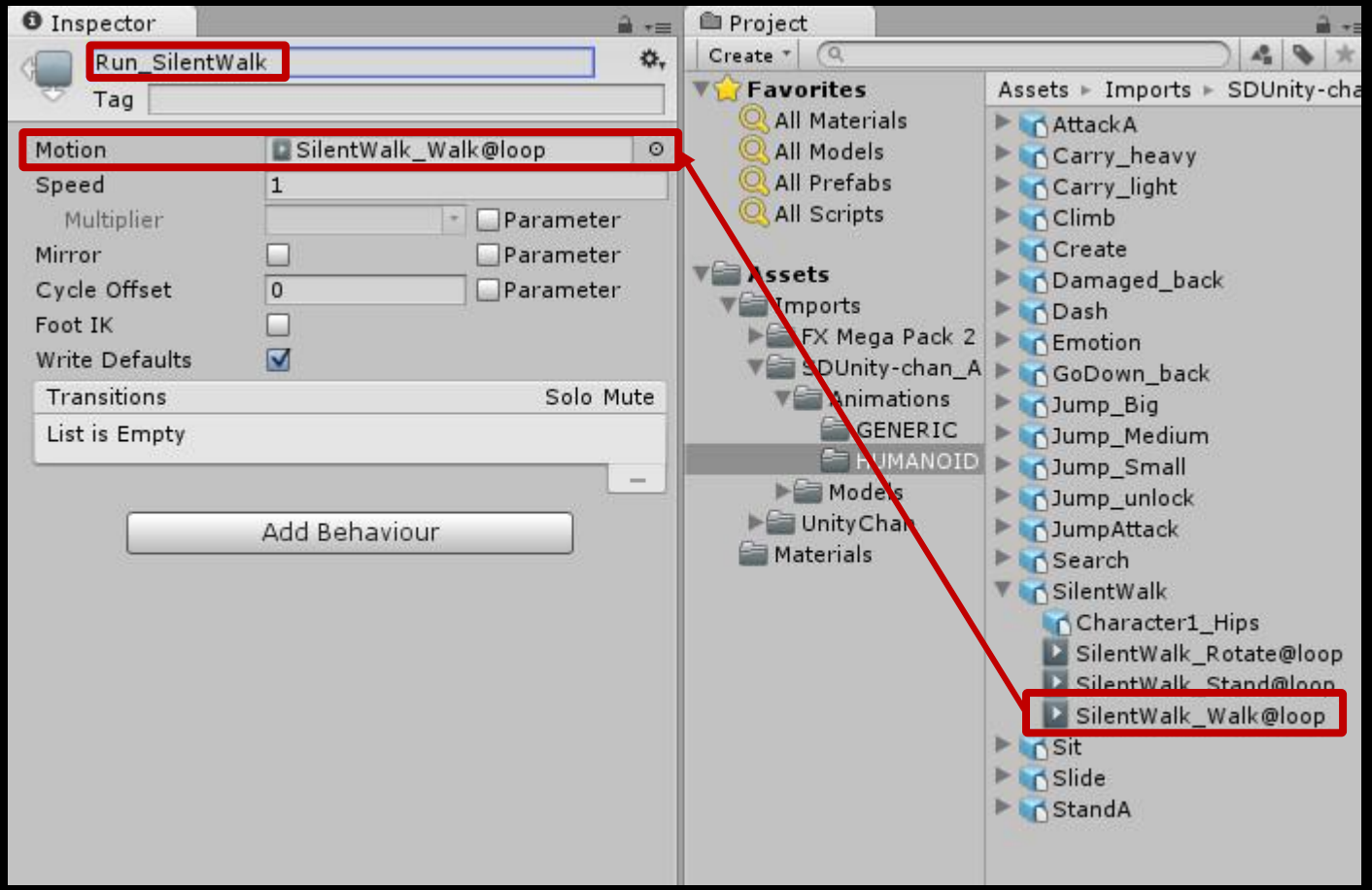
캐릭터 이동

- 상태 설정
  - New State의 이름을 "Run_Base"로 변경
  - Motion은 "Dash_Front@loop"로 변경



캐릭터 이동

- New State 0의 이름을 "Run_SilentWalk"로 변경
- Motion은 "SilentWalk_Walk@loop"로 변경



캐릭터 이동

- Player Class - Update_Inputs()

왼쪽 Shift키를 누르고 있는 상태면 속도 0.5에 걷기 모션을 실행
그렇지 않으면 속도 3.0에 뛰기 모션을 실행

```
29      void Update_Inputs()
30      {
31          Ray         ray;
32          RaycastHit  hit;
33
34          if ( Input.GetMouseButtonDown(1) )
35          {
36              ray = Camera.main.ScreenPointToRay(Input.mousePosition);
37              if ( Physics.Raycast(ray, out hit) )
38              {
39                  if ( Input.GetKey(KeyCode.LeftShift) )
40                  {
41                      move_speed = .5f;
42                      anim.Play("Run_SilentWalk");
43                  }
44                  else
45                  {
46                      move_speed = 3.0f;
47                      anim.Play("Run_Base");
48                  }
49
50                  player_state    = PLAYER_STATE.MOVE;
51                  goal_pos        = hit.point;
52
53                  transform.localRotation = Quaternion.LookRotation(goal_pos-Get_Pos());
54              }
55          }
56      }
```

캐릭터 이동

- Player Class - Update_Actions()

```
57  void Update_Actions()
58  {
59      switch ( player_state )
60      {
61          PLAYER_STATE.IDLE
79          #region PLAYER_STATE.MOVE
80          case PLAYER_STATE.MOVE:
81              if ( Input.GetKey(KeyCode.LeftShift) )
82              {
83                  move_speed = .5f;
84                  anim.Play("Run_SilentWalk");
85              }
86              else
87              {
88                  move_speed = 3.0f;
89                  anim.Play("Run_Base");
90              }
91
92              Vector3 move_pos = Vector3.zero;
93              if ( Vector3.Distance(goal_pos, Get_Pos()) > .1f )
94                  move_pos = Vector3.Normalize(goal_pos-Get_Pos());
95              else
96              {
97                  anim.Play("Idle_Base");
98
99                  Set_Pos(goal_pos);
100                 player_state = PLAYER_STATE.IDLE;
101             }
102             Add_Pos(move_pos * move_speed * Time.deltaTime);
103             break;
104         #endregion
105     }
106 }
```

왼쪽 Shift키를 누르면 속도 0.5에 걷기 모션을 실행

왼쪽 Shift키에서 손을 떼면 속도 3.0에 뛰기 모션을 실행

목표지점에 도달하면 대기(Idle) 상태로 변경

캐릭터 이동

- 결과 화면


Scene
Game
Animator
Display 1
Free Aspect
www.Bandicam.co.kr
Maximize on Play
Mute audio
Stats
Gizmos

캐릭터 이동

카메라 이동 및 회전

- CameraManager Class
  - Main Camera의 Component로 적용

```
using UnityEngine;
using System.Collections;

public class CameraManager : MonoBehaviour
{
    [SerializeField]
    private Transform   target;
    private float       distance;
    private float       xSpeed, ySpeed;
    private float       yMinLimit, yMaxLimit;
    private float       x, y;
    private Vector3     position;
    private Quaternion  rotation;

    void Awake()
    {
        distance    = 10.0f;
        xSpeed      = 250.0f;
        ySpeed      = 120.0f;
        yMinLimit   = 5.0f;
        yMaxLimit   = 80.0f;

        Vector3 angles = transform.eulerAngles;
        x = angles.y;
        y = angles.x;
        rotation = Quaternion.Euler(y, x, .0f);
    }
    void Update()...
    float ClampAngle(float angle, float min, float max)...
}
```

캐릭터 이동

- CameraManager Class - Update()

```
28      void Update()
29      {
30          if ( !target ) return;
31
32          position = rotation * new Vector3(.0f, .0f, -distance) + target.position;
33
34          transform.rotation = rotation;
35          transform.position = position;
36
37          if ( Input.GetAxis("Mouse ScrollWheel") > .0f )
38          {
39              if ( distance > 2.0f ) distance -= .2f;
40              else distance = 2.0f;
41          }
42          if ( Input.GetAxis("Mouse ScrollWheel") < .0f )
43          {
44              if ( distance < 10.0f ) distance += .2f;
45              else distance = 10.0f;
46          }
47
48          if ( Input.GetMouseButton(1) )
49          {
50              if ( !Input.GetKey(KeyCode.LeftControl) ) return;
51
52              x += Input.GetAxis("Mouse X") * xSpeed * 0.02f;
53              y -= Input.GetAxis("Mouse Y") * ySpeed * 0.02f;
54
55              y = ClampAngle(y, yMinLimit, yMaxLimit);
56
57              rotation = Quaternion.Euler(y, x, .0f);
58          }
59      }
```

마우스 휠을 이용해 캐릭터와의 거리 값을 변경

마우스 오른쪽 클릭이 캐릭터 이동에도 사용되기 때문에 Ctrl키를 눌렀을 때만 반응하게 함

이동은 x, y -> 회전은 y, x

캐릭터 이동

- CameraManager Class - ClampAngle()

```
60      float ClampAngle(float angle, float min, float max)
61      {
62          if ( angle < -360 ) angle += 360;
63          if ( angle > 360 )  angle -= 360;
64
65          return Mathf.Clamp(angle, min, max);
66      }
```

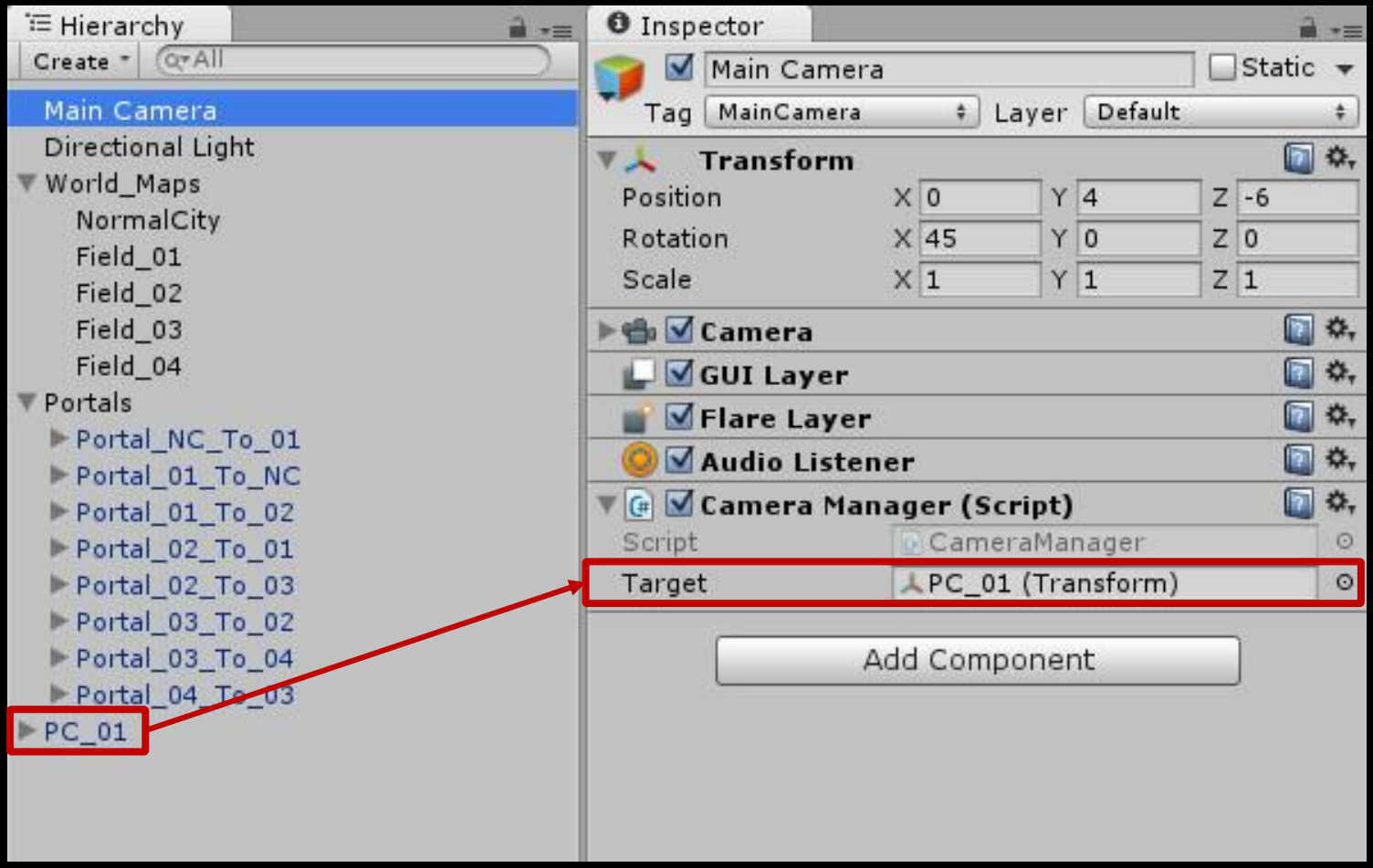
캐릭터 이동

- Character Class - Update_Inputs()

```
29      void Update_Inputs()
30      {
31          Ray         ray;
32          RaycastHit  hit;
33
34          if ( Input.GetMouseButtonDown(1) )
35          {
36              if ( Input.GetKey(KeyCode.LeftControl) ) return;
37
38              ray = Camera.main.ScreenPointToRay(Input.mousePosition);
39              if ( Physics.Raycast(ray, out hit) )
40              {
41                  if ( Input.GetKey(KeyCode.LeftShift) )
42                  {
43                      move_speed = .5f;
44                      anim.Play("Run_SilentWalk");
45                  }
46                  else
47                  {
48                      move_speed = 3.0f;
49                      anim.Play("Run_Base");
50                  }
51
52                  player_state    = PLAYER_STATE.MOVE;
53                  goal_pos        = hit.point;
54
55                  transform.localRotation = Quaternion.LookRotation(goal_pos-Get_Pos());
56              }
57          }
58      }
```

캐릭터 이동

- CameraManager Script 변수 설정
  - Target 변수에 카메라가 따라다닐 "PC_01" 오브젝트를 저장



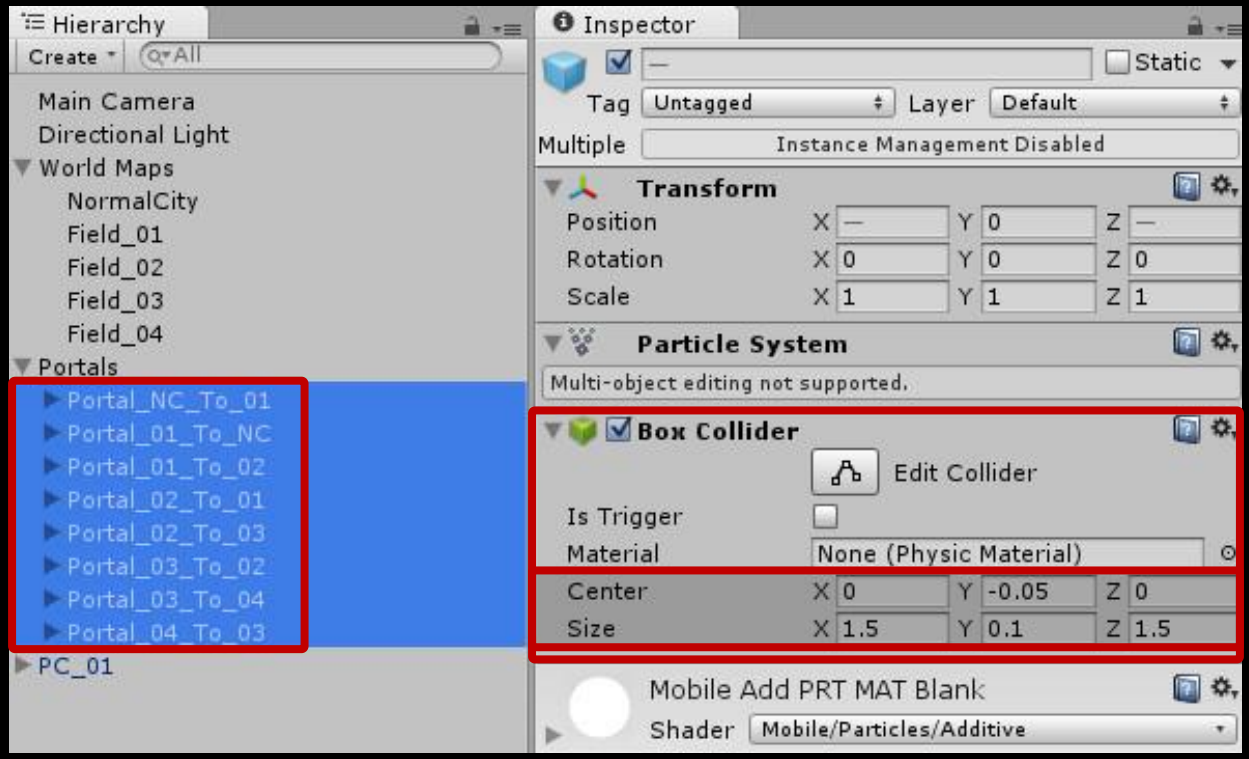
캐릭터 이동

- 결과 화면


Scene
Game
Animator
www.Bandicam.co.kr
Display 1
Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos

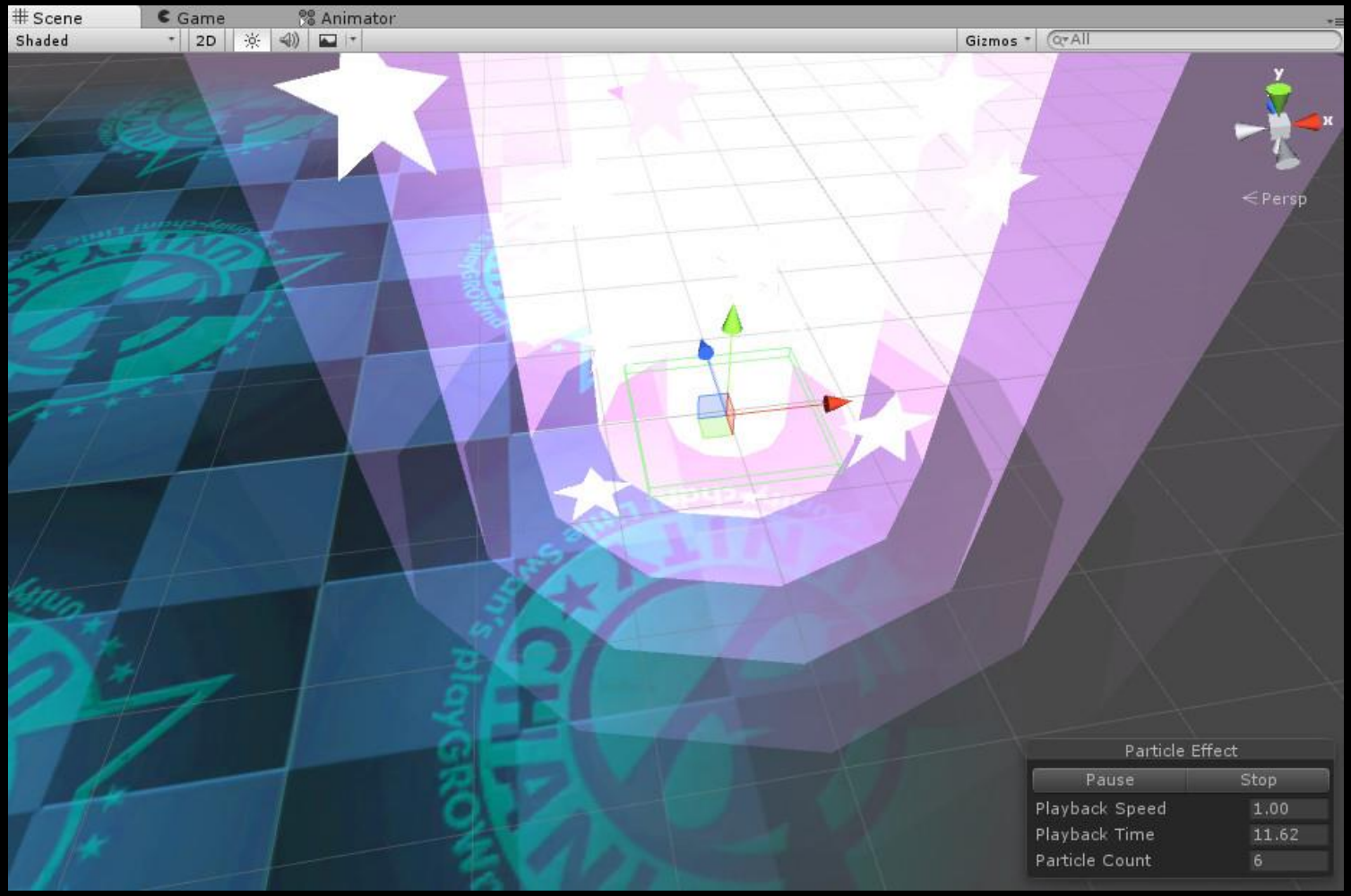
캐릭터 이동

- 포탈을 이용한 필드 이동
  - 캐릭터가 포탈과 충돌할 수 있도록 모든 포탈에 “Box Collider” 적용
    - Component - Physics - Box Collider
  - 포탈의 중심부에 Collider를 배치 - Center(0, -0.05, 0) / Size(1.5, 0.1, 1.5)



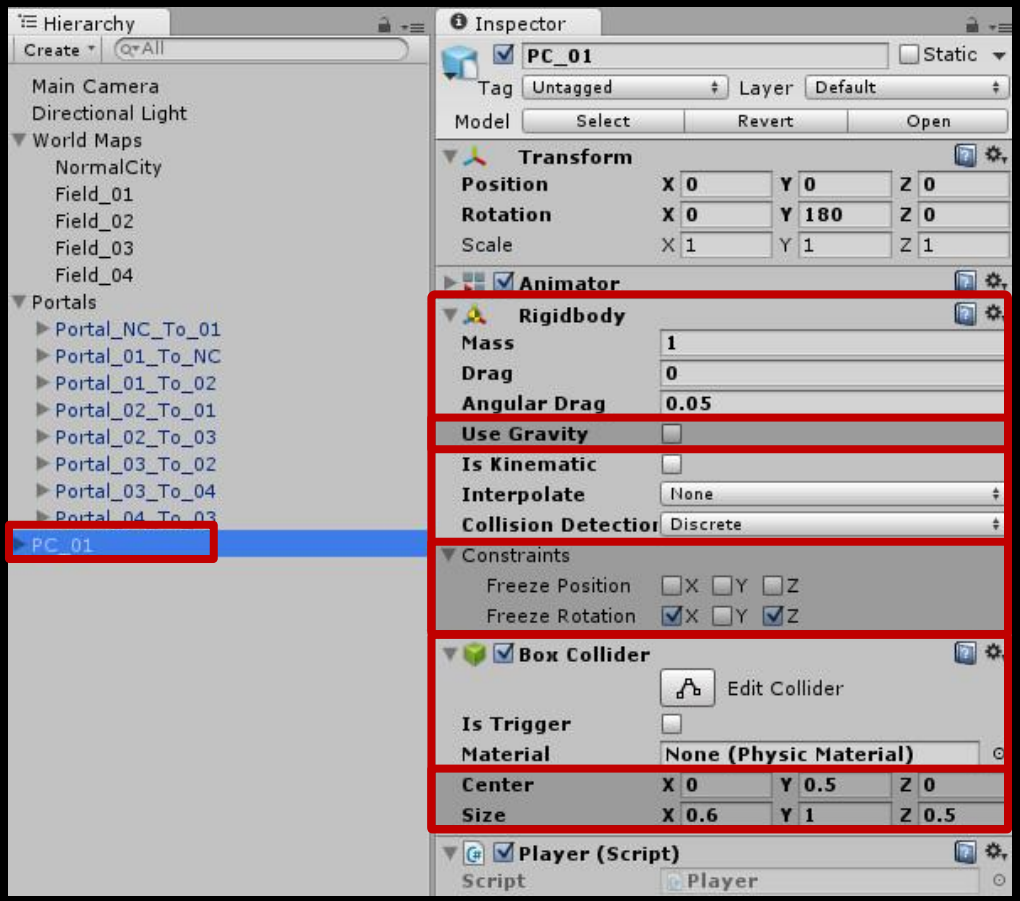
캐릭터 이동

- 포탈 중심부에 배치된 Box Collider



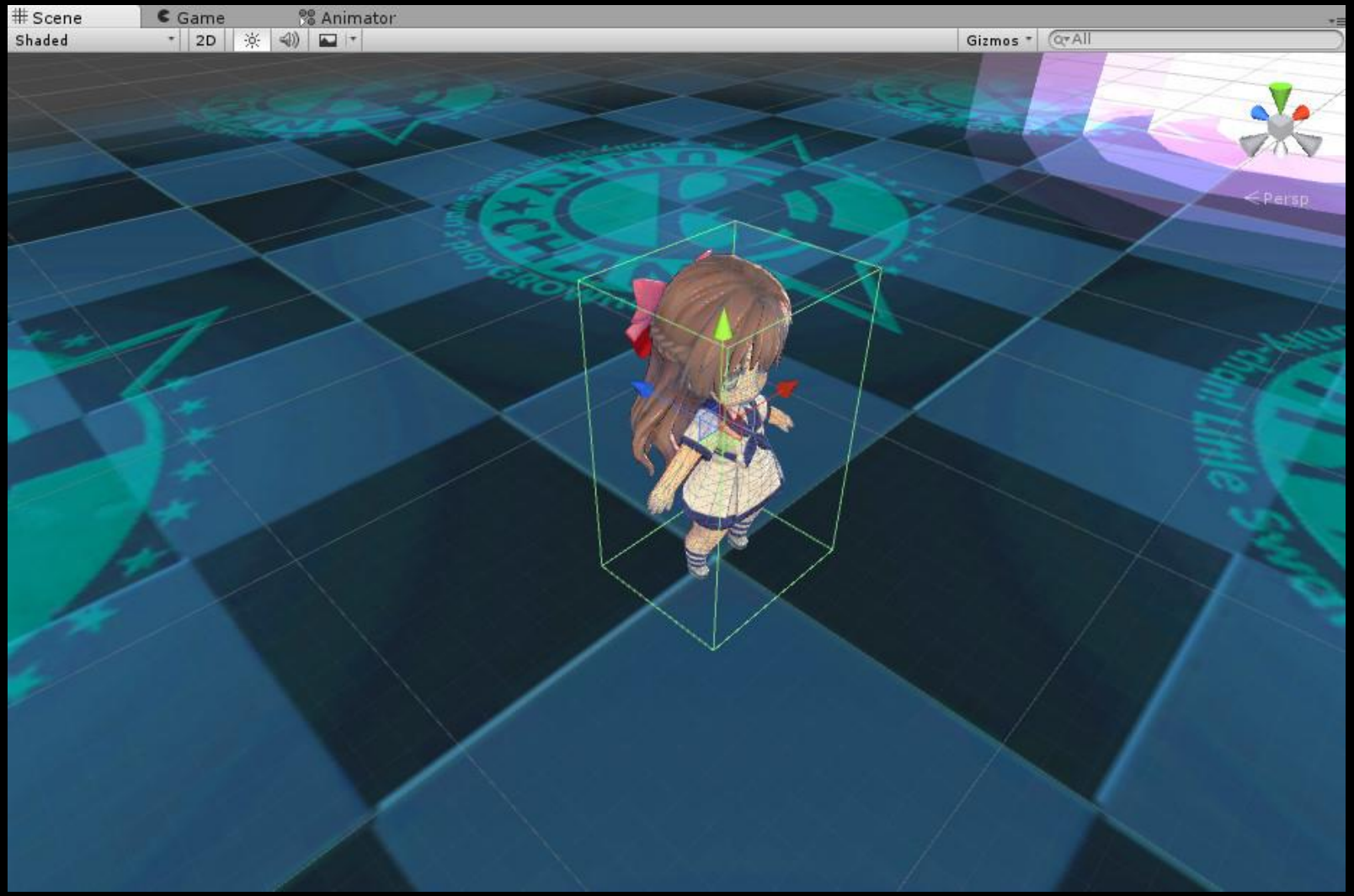
캐릭터 이동

- 캐릭터에게 "Rigidbody"와 "Box Collider" 컴포넌트 적용
  - Compoent - Physics - Rigidbody/Box Collider
- Collider를 캐릭터 모델에 맞게 설정 - Center(0, 0.5, 0) / Size(0.6, 1, 0.5)



캐릭터 이동

- 캐릭터 모델의 Box Collider



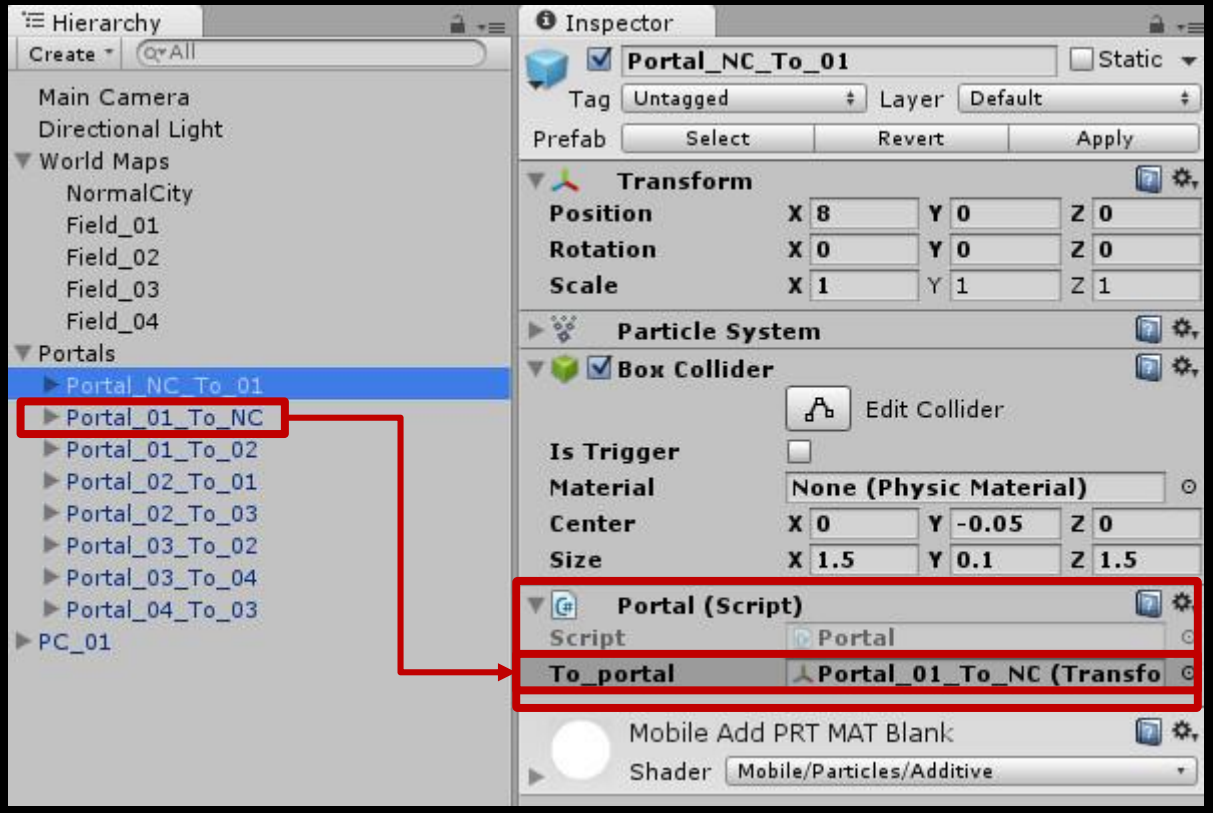
캐릭터 이동

- Portal Class
  - 모든 포탈에게 Portal Class를 컴포넌트로 적용

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Portal : MonoBehaviour
5  {
6      [SerializeField]
7      private Transform   to_portal;
8
9      public void Move_To(Player p)
10     {
11         p.Set_Pos(to_portal.position);
12     }
13 }
```

캐릭터 이동

- Portal Class의 변수 To_portal 설정
  - 현재 포탈을 통해 이동할 포탈의 오브젝트를 To_portal에 저장
  - 포탈 (NC ↔ 01) / (01 ↔ 02) / (02 ↔ 03) / (03 ↔ 04)



캐릭터 이동

- Player Class

```
1   using UnityEngine;
2   using System.Collections;
3
4   public enum PLAYER_STATE { IDLE=0, MOVE }
5
6   public class Player : MonoBehaviour
7   {
8       private Animator        anim;
9       private PLAYER_STATE    player_state;
10      private float           idle_time;
11
12      private float           move_speed;
13      private Vector3         goal_pos;
14
15      private GameObject      target_portal;
16      private Rigidbody       _rigid;
17
18      void OnCollisionEnter(Collision col)...
31      void OnCollisionExit(Collision col)...
35      void Awake()...
47      void Update()...
52      void Update_Inputs()...
89      void Update_Actions()...
142     public void     Add_Pos(Vector3 p)  { transform.position += p; }
143     public void     Set_Pos(Vector3 p)  { transform.position = p; }
144     public Vector3  Get_Pos()           { return transform.position; }
145 }
```

캐릭터 이동

- Player Class - OnCollisionEnter(), OnCollisionExit()

target이 없거나 현재 부딪힌 오브젝트의 이름에 "Portal"이 포함되어 있지 않을 땐 return

```
18      void OnCollisionEnter(Collision col)
19      {
20          if ( target_portal == null )                    return;
21          if ( !col.gameObject.name.Contains("Portal") )  return;
22
23          Portal portal = col.gameObject.GetComponent<Portal>() as Portal;
24          portal.Move_To(this);
25
26          target_portal = null;
27
28          anim.Play("Idle_Base");
29          player_state = PLAYER_STATE.IDLE;
30      }
31      void OnCollisionExit(Collision col)
32      {
33          _rigid.isKinematic = true;
34      }
```

Portal 클래스의 Move_To() 메소드를 이용해 캐릭터를 이동시키고, target을 해제

이동(Move) 상태로 포탈에 도착하였기 때문에 이동 후엔 대기(Idle) 상태로 변경해 주어야 함

오브젝트와의 충돌 후 물리력을 제거하기 위한 isKinematic = true

캐릭터 이동

- Player Class - Awake()

```
35      void Awake()
36      {
37          anim            = GetComponent<Animator>();
38          player_state    = PLAYER_STATE.IDLE;
39          idle_time       = .0f;
40
41          move_speed      = 3.0f;
42          goal_pos        = Vector3.zero;
43
44          target_portal   = null;
45          _rigid          = GetComponent<Rigidbody>();
46      }
```

캐릭터 이동

- Player Class - Update_Inputs()

```
52  void Update_Inputs()
53  {
54      Ray         ray;
55      RaycastHit  hit;
56
57      if ( Input.GetMouseButtonDown(1) )
58      {
59          if ( Input.GetKey(KeyCode.LeftControl) ) return;
60
61          ray = Camera.main.ScreenPointToRay(Input.mousePosition);
62          if ( Physics.Raycast(ray, out hit) )
63          {
64              if ( hit.transform.name.Equals("PC_01") ) return;
65
66              if ( hit.transform.name.Contains("Portal") )
67                  target_portal = hit.transform.gameObject;
68              else
69                  target_portal = null;
70
71              if ( Input.GetKey(KeyCode.LeftShift) )
72              {
73                  move_speed = .5f;
74                  anim.Play("Run_SilentWalk");
75              }
76              else
77              {
78                  move_speed = 3.0f;
79                  anim.Play("Run_Base");
80              }
81
82              player_state    = PLAYER_STATE.MOVE;
83              goal_pos        = hit.point;
84
85              transform.localRotation = Quaternion.LookRotation(goal_pos-Get_Pos());
86          }
87      }
88  }
```

캐릭터를 클릭했을 땐 반응하지 않음

마우스 클릭으로 선택한 오브젝트(hit)의 이름에 "Portal"이 들어가면 target으로 설정

캐릭터 이동

- Player Class - Update_Actions()

```
89  void Update_Actions()
90  {
91      switch ( player_state )
92      {
93          PLAYER_STATE.IDLE
111         #region PLAYER_STATE.MOVE
112         case PLAYER_STATE.MOVE:
113             if ( Input.GetKey(KeyCode.LeftShift) )
114             {
115                 move_speed = .5f;
116                 anim.Play("Run_SilentWalk");
117             }
118             else
119             {
120                 move_speed = 3.0f;
121                 anim.Play("Run_Base");
122             }
123
124             Vector3 move_pos = Vector3.zero;
125             if ( Vector3.Distance(goal_pos, Get_Pos()) > .1f )
126                 move_pos = Vector3.Normalize(goal_pos-Get_Pos());
127             else
128             {
129                 anim.Play("Idle_Base");
130
131                 Set_Pos(goal_pos);
132                 player_state = PLAYER_STATE.IDLE;
133             }
134             Add_Pos(move_pos * move_speed * Time.deltaTime);
135
136             if ( _rigid.isKinematic == true )
137                 _rigid.isKinematic = false;
138             break;
139         #endregion
140     }
141 }
```

OnCollisionExit()에서 해제한 물리를 다시 적용

캐릭터 이동

- 결과 화면


Scene
Game
Animator
www.Bandicam.co.kr
Display 1
Free Aspect
Maximize on Play
Mute audio
Stats
Gizmos