1. Create input map for movement

```
[Serializable]
public sealed class InputMap
{
   public KeyCode forward;
   public KeyCode back;
   public KeyCode left;
   public KeyCode right;
}
```

2. Write function that return input move direction

```
public static class InputUseCases
{
   public static float3 GetMoveDirection(this InputMap inputMap)
   {
       float3 moveDirection = float3.zero;

       if (Input.GetKey(inputMap.left))
       {
           moveDirection.x = -1;
       }
       else if (Input.GetKey(inputMap.right))
       {
           moveDirection.x = 1;
       }

       if (Input.GetKey(inputMap.back))
       {
           moveDirection.z = -1;
       }
       else if (Input.GetKey(inputMap.forward))
       {
           moveDirection.z = 1;
       }

       return moveDirection;
   }
}
```

3. Create Movement System

```
public sealed class CharacterMovementSystem : IContextInit, IContextUpdate
{
   private IValue<IEntity> _character;
   private InputMap _inputMap;

   public void Init(IContext context)
   {
       _character = context.GetCharacter();
       _inputMap = context.GetInputMap();
   }

   public void Update(IContext context, float deltaTime)
   {
       _character.Value.GetMoveDirection().Value = _inputMap.GetMoveDirection();
   }
}
```

4. Install Movement Components to Player Context

```
public sealed class CharacterMovementInstaller : SceneContextInstallerBase
{
   [SerializeField]
   private InputMap inputMap;

   public override void Install(IContext context)
   {
       context.AddInputMap(this.inputMap);
       context.AddSystem<CharacterMovementSystem>();
   }
}
```