1.Write Earn Money Procedure

```
public static class PlayerMoneyUseCase
{
   public static void EarnMoney(this IContext playerContext, in int money)
   {
       playerContext.GetMoney().Value += money;
   }
}
```

2. Write Destroy Coin Procedure

```
public static class DestroyCoinUseCase
{
   public static void DestroyCoin(this IContext gameContext, IEntity coin)
   {
       gameContext.GetCoinSystemData().pool.Return(coin);
   }
}
```

3. Write Collect Coin Procedure

```
public static class CollectCoinUseCase
{
   public static bool CollectCoin(this IContext playerContext, in IEntity coin)
   {
       if (!coin.HasCoinTag())
       {
           return false;
       }

       playerContext.EarnMoney(coin.GetMoney().Value);

       IContext gameContext = playerContext.Parent;
       gameContext.DestroyCoin(coin);
       return true;
   }
}
```

4. Write CollectCoinSystem

```
public sealed class CoinCollectSystem : IContextInit, IContextEnable,
IContextDisable
{
   private IContext _playerContext;
   private IValue<IEntity> _character;

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

   public void Enable(IContext context)
   {
       _character.Value.GetTriggerEventReceiver().OnEntered +=
this.OnTriggerEntered;
   }

   public void Disable(IContext context)
   {
       _character.Value.GetTriggerEventReceiver().OnEntered -=
this.OnTriggerEntered;
   }

   private void OnTriggerEntered(Collider other)
   {
       if (other.TryGetComponent(out IEntity target))
       {
           _playerContext.CollectCoin(target);
       }
   }
}
```

4. Write Collect Coin Installer for Player Context

```
public sealed class CoinCollectInstaller : SceneContextInstallerBase
{
   public override void Install(IContext context)
   {
       context.AddMoney(new ReactiveVariable<int>());
       context.AddSystem<CoinCollectSystem>();
   }
}
```