Project contains such scripts and controllers.

1. Player Controller.
   It contains player movement logic, taking advantage of Unity’s new Input system, catching InputValue and using this as movement direction.

2. ResourceItemType, ShopElementDescriptor
   Represents Enum with resources that can be bought in shop. Some of them can be equipped by player. ShopElementDescriptor contains data about items in shop - type, price, icon, count.

3. BaseShopElementView, ShopElementModel, ShopWindow
   They control UI initialization of the shop and its elements. Helper script (AssetProvider) loads from resources item’s SO that are available for display by corresponding ResourceItemTypes referenced in the shop window. Then instances of BaseShopElementView are created, and this happens once during initialization. Other times the script just enables created instances. All instances are destroyed with the shop window.

4. ShopManager, IShopCustomer, ShopWindowTrigger
   ShopManager is responsible for a purchase process - adds resources, decreases player in-game currency, checks if player already has such equipment to decide whether to invoke a callback using IShopCustomer interface. When player enters the shop trigger area, ShopWindowTrigger opens the shop window and passes IShopCustomer as a parameter to init.

5. PlayerAnimationController, PlayerStyleSetupHelper
   PlayerStyleSetupHelper initializes bought equipment and sets it to a player prefab. PlayerAnimationController controls character animation transition between idle state and moving state by setting boolean parameter tied to transitions. It also set such flag for all of the equipped clothes via style setup helper.

6. GameData, PlayerDataPersist
   Wanted to use this as a save&load system to store bought items, player resources and position to have the ability to play the latest game and start a new one. For that purpose, my initial thought was to create a main screen and have these options there.