In-Game Currency

Using the Sui Closed-Loop Token standard, you can create in-game currency (such as gems or diamonds in mobile games) that you can grant to players for their actions or make available to purchase. You mint the tokens on Sui, but players can only use the tokens within the economy of the game itself. These types of tokens are usually not transferrable and you would typically mint them in predefined amounts to maintain scarcity and game balance.

The following example creates an in-game currency called a GEM, which represents a certain number of SUI. In the example, the user can buy fungible GEMs using SUI, which can then be used as currency within the game. Use the code comments to follow the logic of the example.

The Sui repo hosts a basic example of creating in-game currency. The Move modules that create the economy of the example are in the gems.move source file.

The examples::sword module creates one of the objects, a sword , that has in-game value. The module assigns a value in GEMs (the other valuable in-game item) to the sword. The module also provides the logic for trading GEMs to receive a sword.

The examples::gem module creates the in-game currency, GEMs. Users spend SUI to purchase GEMs, which can then be traded for swords. The module defines three groups of GEMs (small, medium, and large), with each group representing a different in-game value. Constants hold both the value of each package and the actual number of GEMs the groups contain.

The module's init function uses coin::create_currency to create the GEM. The init function, which runs only the one time when the module publishes, also sets the policies for the in-game currency, freezes the metadata for the coin, and transfers the policy capability to the publisher of the package.

The module handles the purchase of GEMs with the buy_gems function.

Use the following toggle to control the display of the complete module.

examples::gem module in gems.move

Toggle display of the complete source for this example, including comments, or use the link in the Related links section to view the project source on GitHub.

gems.move

Example

The Sui repo hosts a basic example of creating in-game currency. The Move modules that create the economy of the example are in the gems.move source file.

The examples::sword module creates one of the objects, a sword , that has in-game value. The module assigns a value in GEMs (the other valuable in-game item) to the sword. The module also provides the logic for trading GEMs to receive a sword.

The examples::gem module creates the in-game currency, GEMs. Users spend SUI to purchase GEMs, which can then be traded for swords. The module defines three groups of GEMs (small, medium, and large), with each group representing a different in-game value. Constants hold both the value of each package and the actual number of GEMs the groups contain.

The module's init function uses coin::create_currency to create the GEM. The init function, which runs only the one time when the module publishes, also sets the policies for the in-game currency, freezes the metadata for the coin, and transfers the policy capability to the publisher of the package.

The module handles the purchase of GEMs with the buy_gems function.

Use the following toggle to control the display of the complete module.

examples::gem module in gems.move

Toggle display of the complete source for this example, including comments, or use the link in the Related links section to view the project source on GitHub.

gems.move

Related links