Intro to Sui Objects

Speaker: Henry Duong, Developer Relations Engineer

MystenLabs

Workshop Outline

- Sui Object Basics
- Environment Setup and Tools
- Basic NFT Contract Demo
- Parent Child Objects Demo
- Access Control and Paid Mint Demo

Main Repo for the Workshop

- https://github.com/hyd628/sui_intro_workshop

Sui Object Basics

In Sui, Every Object is an NFT

Sui Modules and Immutable Objects

- A module in Sui is an immutable object and will have an object ID
- Immutable objects can be used by anyone, so all modules in Sui are public/can be used by anyone
- To use the Sui CLI to publish a module:
    - `sui client publish --path [filepath to the module] --gas-budget [sufficient amount of gas]`

Parent and Child Objects

- An object in Sui can own other objects, those are known as parent objects and child objects
- Using children objects requires passing in its parent object(s)
- References to children objects are represented with the Option type
- A parent object cannot be frozen, wrapped or deleted while it has children objects

Initializer Function

- A special function that’s run exactly once when the module is published
- Serves to initialize the module state and create singleton objects
- Must be named init, no parameters other than TxtContext, no return values, private

Shared Object

- An object in Sui can become shared if it’s transferred using the `sui::transfer::share_object` method
- Anyone can read or write to a shared object
- Shared objects require consensus (Narwhal & Bullshark) to sequence reads and writes
- In other blockchains, every entity is shared by default. In Sui, developers have the choice to make something shared or non-shared.

Capability

- A design pattern that allows authorizing actions with an object
- Serves a similar purpose as role based access control in Solidity smart contracts
- A capability object generally only has to have one ability: key
- Methods can require passing in the capability object (and consuming it immediately) to enforce access control

Environment Setup and Tools

Environment Setup

Package/OS	Linux	macOS	Windows 11
Curl	X	X	X
Rust	X	X	X
Git CLI	X	X	X
CMake	X	X	X
libssl-dev	X		
libclang-dev	X		
Brew		X	
C++ build tools			X
LLVM Compiler			X
Sui	X	X	X

Install Sui Binaries and CLI

- git clone https://github.com/MystenLabs/sui.git --branch devnet
- git fetch upstream
- git checkout devnet-0.10.0
- cargo install --locked --git https://github.com/MystenLabs/sui.git --branch "devnet" sui sui-gateway
- sui client (use default options to connect to devnet)

Tools Used

- Sui CLI (installed with Sui binaries)
- Sui Explorer
  - https://explorer.devnet.sui.io/
- VS Code or any other editor

Demo

Sui Dino NFT Smart Contract (basic version)

- https://github.com/hyd628/sui_intro_workshop/blob/main/sources/sui_dinos.move_basic

Sui Dino NFT Smart Contract (child object version)

- https://github.com/hyd628/sui_intro_workshop/blob/main/sources/sui_dinos.move_childobject

Sui Dino NFT Smart Contract (access control version)

- https://github.com/hyd628/sui_intro_workshop/blob/main/sources/sui_dinos.move

Get in Touch

- Discord: https://discord.gg/GcFNX4WMrB
- Twitter: @Mysten_Labs
- Developer Docs: https://docs.sui.io/
- GitHub:
  - https://github.com/MystenLabs/sui
  - https://github.com/MystenLabs/awesome-move
  - https://github.com/move-language/move
- Contact: Email: henry@mystenlabs.com

  Twitter/Telegram: @henrydevrel

Workshop Repo and Slides

Breakout Coding Session

Instructions

https://github.com/hyd628/sui_intro_workshop#breakout-session-instructions

Challenge One

Create a new method in the `sui_intro_workshop::dino_nft` module for the owner of the NFT contract to withdraw the balance collected in `MintingTreasury` to their own account.

Hint: the contract already has most of the infrastructure needed to do this. What are we using to mark the owner or minter account of the NFT contract?

Challenge Two

Currently, each dino NFT object can only have one child NFT object (try to mint a new dino NFT object to a dino NFT that already has a child and see what error you get).

Make the changes necessary to the dino_nft module to allow each parent Dino NFT to own many child NFTs.

Hint: You will need to utilize the vector data structure: https://move-book.com/advanced-topics/managing-collections-with-vectors.html