

RPPIECES

INSTRUCTION BOOKLET

Remember, all characters, places, and items portrayed in this game are purely from our caffeine-fueled imaginations. Any resemblance to actual persons, living or dead, or actual events is purely coincidental. And by "coincidental," we mean "we totally made this up."

Heads up: Choking Hazard - Small parts. Not for kids under 3 years or adults who still put random things in their mouths.

This game is intended for ages 12 and up. Or ages 8 to 80 if you're cool like that.

Printed in Japan

First Edition: May 2023

ISBN: 978-0-WeLoveThe80s-1

For a good time, check out more Soroban and its sweet documentation at our website,
<https://soroban-stellar.org>



SHH. KEEP IT DOWN!

We're glad you're here, we desperately need your help. We're currently engaged in a week-long siege of the Soroban RPC. The RPCiege! Over the course of the next five days, we'll be tackling five challenges. There are only three critically important pieces of information you'll need to remember.



1. The first is always use our official RPC endpoint when enacting your attacks
<https://futurenet.rpciege.com:443> along with the **FUTURENET** network passphrase
Test SDF Future Network ; October 2022
2. The second is that for each gameplay contract invocation you'll want to include as the function's final argument the **Address** of your own mainnet Stellar public key where you'd like to receive your pack of commemorative NFT cards. We will issue NFTs as claimable balances to this address only if it exists as a funded account on the **PUBLIC** network
3. You can claim your NFT cards wherever you wish however we've built a simple claim page over at rpciege.com/claim to aid in the claiming process



BEWARE

Cards will be issued as rare animated assets for three weeks after the release of each Skirmish at which point they will begin being issued as common static assets.

Getting Started with Soroban

Before you begin your journey, you need only take your first step. If you're new here - these tips will be your guidestones.



Setup

Install and configure Rust and the Soroban CLI.



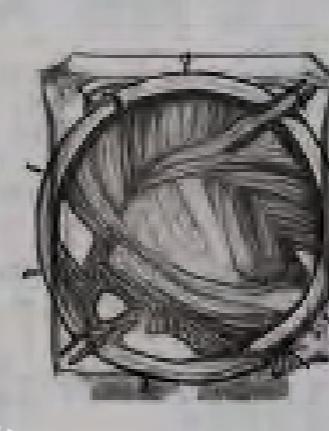
Soroban RPC

The RPC service allows you to communicate directly with Soroban via a JSON RPC interface.



High-Level Overview

Descriptions of key Soroban Concepts.



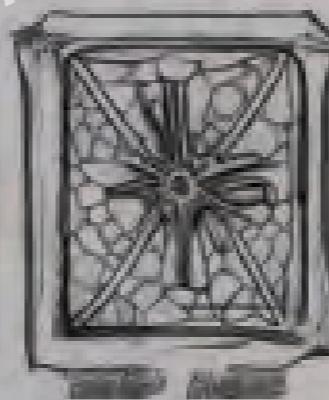
Hello World

Create your first Soroban contract.



Soroban CLI

Soroban CLI is the command line interface to Soroban.



Deploy to Futurenet

Deploy and invoke your contracts on the Futurenet testnet.

If you're prepared, continue on. **TO BATTLE!**

Skirmish I

For battle 1 all you've got to do is submit a contract invocation. That's it. Remember it needs to include as its final argument an **Address** of your mainnet Stellar public key where you'd like to receive your card pack for today's skirmish, but that's the only requirement.

Here's an example contract, just because I like you ❤️

```
#![no_std]

use soroban_sdk::{contractimpl, Address, Env};

pub struct Contract;

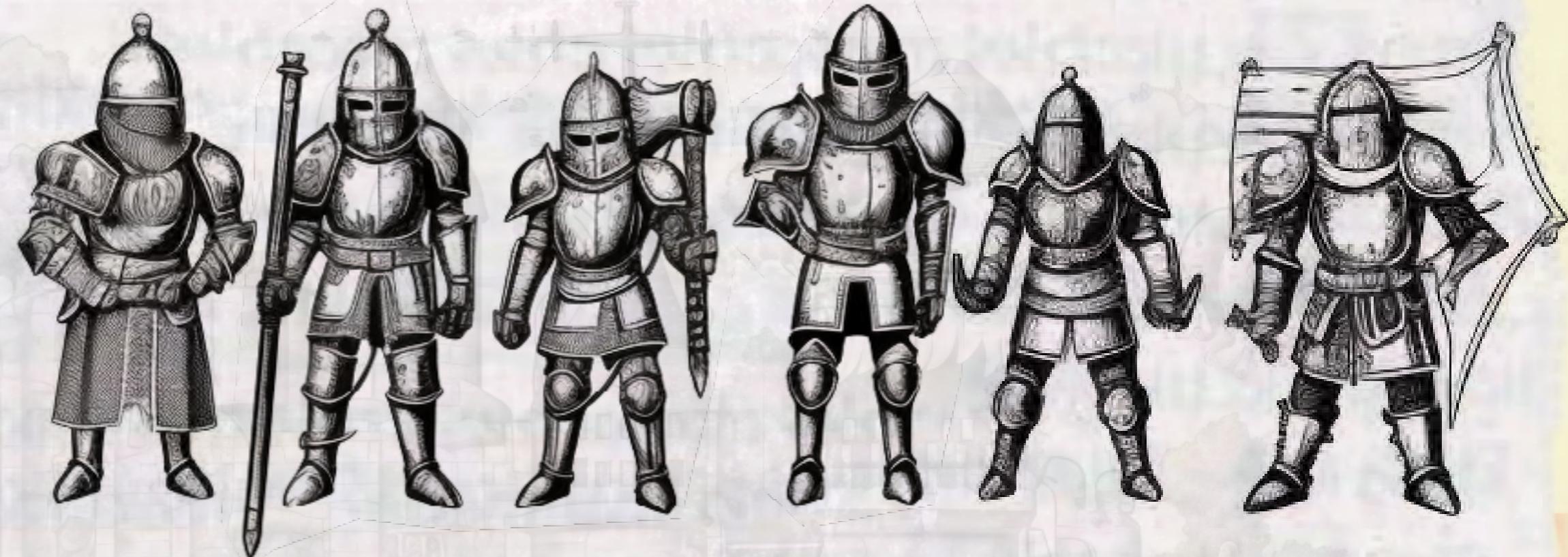
#[contractimpl]
impl Contract {
    pub fn game_1(_env: Env, _nft_dest: Address) {}
}
```



Skirmish II

Welcome to day 2 of the legendary RPCiege! For today's skirmish, your goal is to create and invoke a contract that returns the **String 1694-1727**.

Don't forget the three rules but other than that good luck and go nuts!



No contract cheat codes today but here's a link to some documentation you might find useful:

String in soroban_sdk - Rust

String is a contiguous growable array type containing u8s.

https://docs.rs/soroban-sdk/latest/soroban_sdk/struct.String.html



Skirmish III

Morning dawns on day 3 of the RPCiege. For today's quarrel we're enacting a new tactic: brute force. To win the battle you must submit an invocation to the `game_3` function of the [a9f195600907c6204a55831ce0347b03a3409c7194e3c944c604cf7c16f495de](#) contract.



Here's that contract's Rust code.

```
#![no_std]

use soroban_sdk::{
    contracterror, contractimpl, panic_with_error, xdr::ToXdr, Address, Env, Symbol,
};

pub struct Contract;

#[contracterror]
#[derive(Copy, Clone, Debug, Eq, PartialEq, PartialOrd, Ord)]
#[repr(u32)]
pub enum Error {
    MissingPew = 1,
}

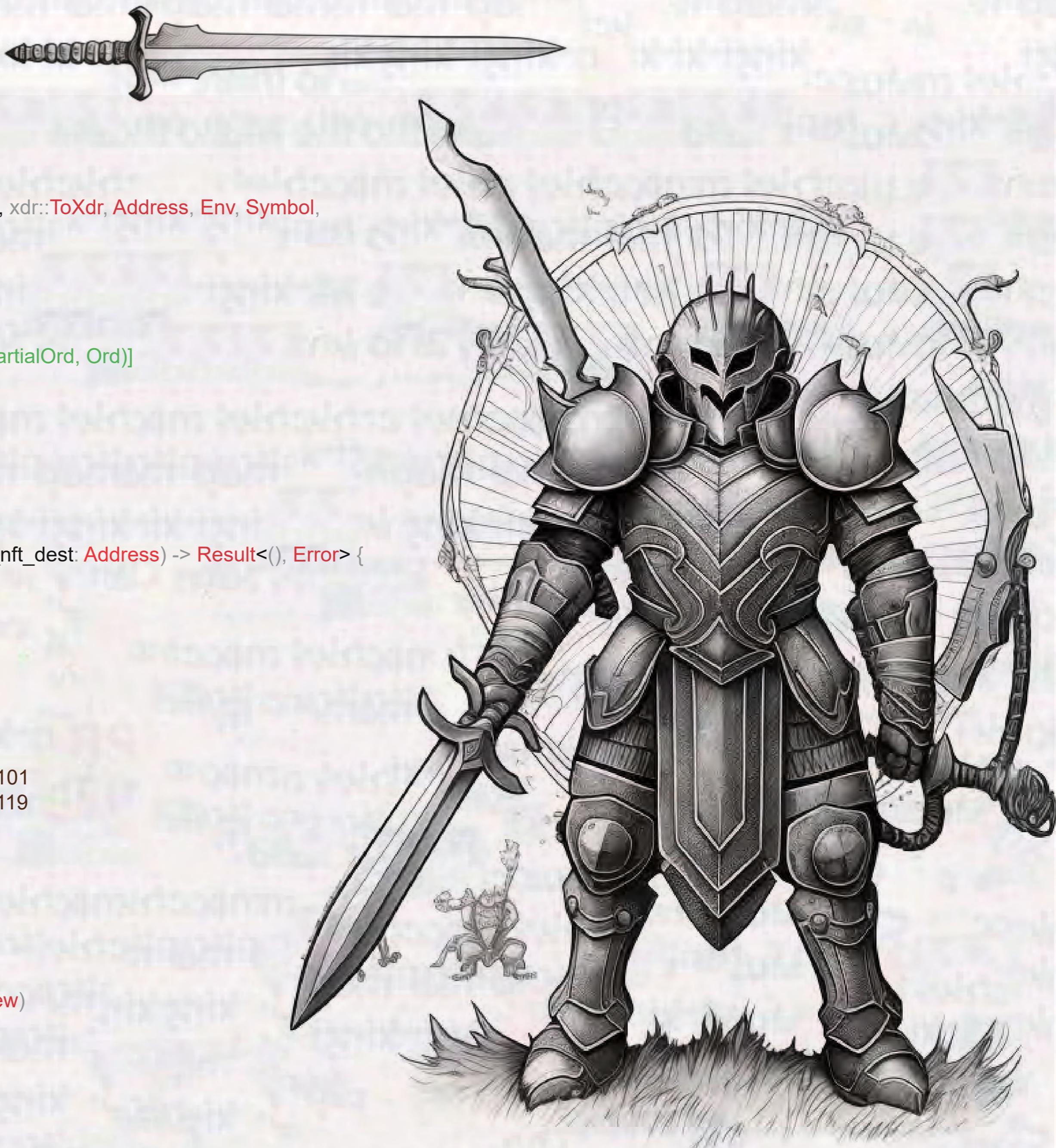
#[contractimpl]
impl Contract {
    pub fn game_3(env: Env, symbol: Symbol, _nft_dest: Address) -> Result<(), Error> {
        let bytes = symbol.to_xdr(&env);
        let hash = env.crypto().sha256(&bytes);

        let mut i = 0;
        let mut has_pew = false;

        for v in hash.clone().into_iter() {
            if v == 112
                && hash.get(i + 1).unwrap_or(0) == 101
                && hash.get(i + 2).unwrap_or(0) == 119
            {
                has_pew = true;
            }
            i += 1;
        }

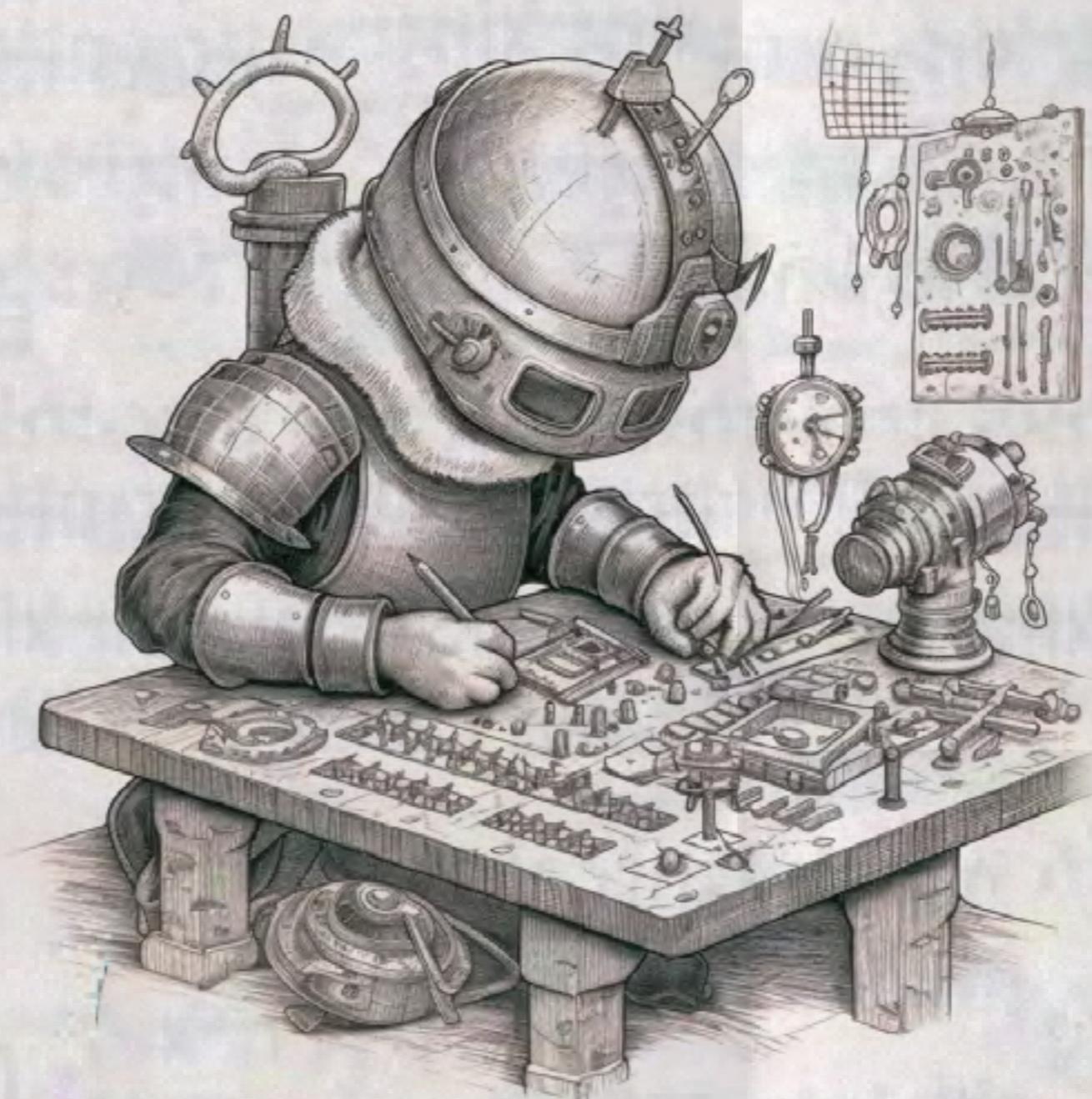
        if !has_pew {
            panic_with_error!(env, Error::MissingPew)
        }

        Ok(())
    }
}
```



No spoilers but I'm telling you now that writing a test against this contract will be your best ally.

Prepare yourself.



Tune back in tomorrow for **Skirmish IV**

Need some help in the meantime? Check out the [Soroban documentation](#), join the [developer Discord](#), and [follow Soroban on Twitter](#).



soroban.stellar.org

2023 The Stellar Development Foundation

PRINTED IN JAPAN