Solidity Programming

Part 1: Enhancing Hero Game 1

Enhance the game by allowing funds to be won. When a Hero is created some funds (ERC20 token) are transferred from the Hero account to the Game contract. And whoever wins the battle gets to win the tokens transferred earlier.

To make the coding simple, you can use the simple random number generator below but don't use this in live deployments.

To test:

1. Deploy an ERC20 contract
2. Send ERC20 to all players
3. Use the ERC20 address in the Hero contract constructor
4. Test createHero, check token is transferred
5. Test Attack, check token is transferred

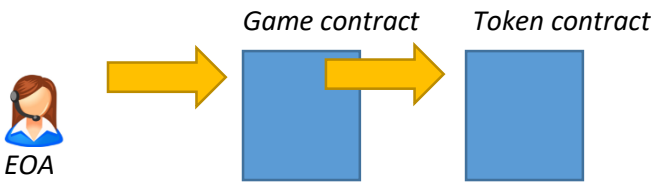
Simple Random Number Generator
We can use a simple random number generator:

```
uint randomResult = uint(keccak256(abi.encodePacked(block.timestamp, msg.sender, nonce)))
% 100;
nonce++;
```

Note: **This random generator is not recommended for use in actual situations.**

Notes on the use of transferFrom

Recap



msg.sender=EOA msg.sender=game-contract

If it is necessary for game-contract to transfer tokens on behalf on an EOA, then the EOA needs to submit to approve the game-contract before the transferFrom can completed by game-contract on the EOA behalf.

Part 2: Enhancing Hero Game 2

This adds advanced features into the game.

Features:

- 3 stages of play: registration, tournament, payout
- Players register and send token for each tournament round
- Each player gets to do one battle per tournament
- The player with the most wins is the tournament winner
- Tournament ends when all players get a turn
- If there is no single winner, tokens are not distributed, tournament is repeated
- Winner takes % the rest goes to contract.

Contract TournamentPlay

Functions Description

```
Register ()
if Hero= msg.sender have not been created  then
        createHero
        Store address to heroID
        Transfer funds from Hero to contract
        emit

Attack (attackerId, targetId)
if battle is not done then
        Generate random number
        If random > attackerProbability then
                attacker win
                emit
        Update data

checkWinner()
if all Hero have battled then
        return Hero with most wins
        transfer funds from Contract to winner
        emit
        reset battle records to zero
```