🎮 Staking Booster

Tier-based Multiplier System for SAIL's Experience Points

In the game SAIL, players have the opportunity to enhance their gaming experience by staking $SOPH tokens. This staking mechanism is stratified into three main tiers: Bronze, Gold, and Diamond. Each tier requires a minimum amount of $SOPH tokens to be staked and has associated daily bonus multipliers that increase based on the length of the staking period.

- **Bronze Tier**: By staking 1,000+ $SOPH tokens, players enter the Bronze tier, which starts with a 1.2x daily bonus multiplier for a 3-month staking period. The multiplier can increase to 1.3x and 1.5x for staking periods of 6 and 12 months, respectively.
- **Gold Tier**: A higher commitment of 5,000+ $SOPH tokens places a player in the Gold tier. The starting daily bonus multiplier is 1.3x for 3 months, increasing to 1.4x for 6 months, and 1.75x for a full year.
- **Diamond Tier**: The most premium tier, Diamond, requires a stake of 10,000+ $SOPH tokens. It rewards players with a 1.4x multiplier for a 3-month period, a 1.5x multiplier for 6 months, and a substantial 2x multiplier for 12 months.
- **Transcendence Tier**: multiplier (2x across the board for 3, 6, 12 months), minimum 50K ;
- **Singularity Tier**: minimum 100K multiplier (2x across the board for 3,6, 12 months) ;

Additionally, there is a concept of "LP Tokens," which seems to offer a 1.1x daily bonus multiplier, possibly as a separate or additional incentive mechanism.

```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.19;

import {StakeBooster} from "../StakeBooster.sol";

contract BoosterBronze is StakeBooster {

    constructor(address _soph, string memory _uri3, string memory _uri6, string memory _uri12)
    StakeBooster(_soph, 1000 ether, _uri3, _uri6, _uri12){}
}
```
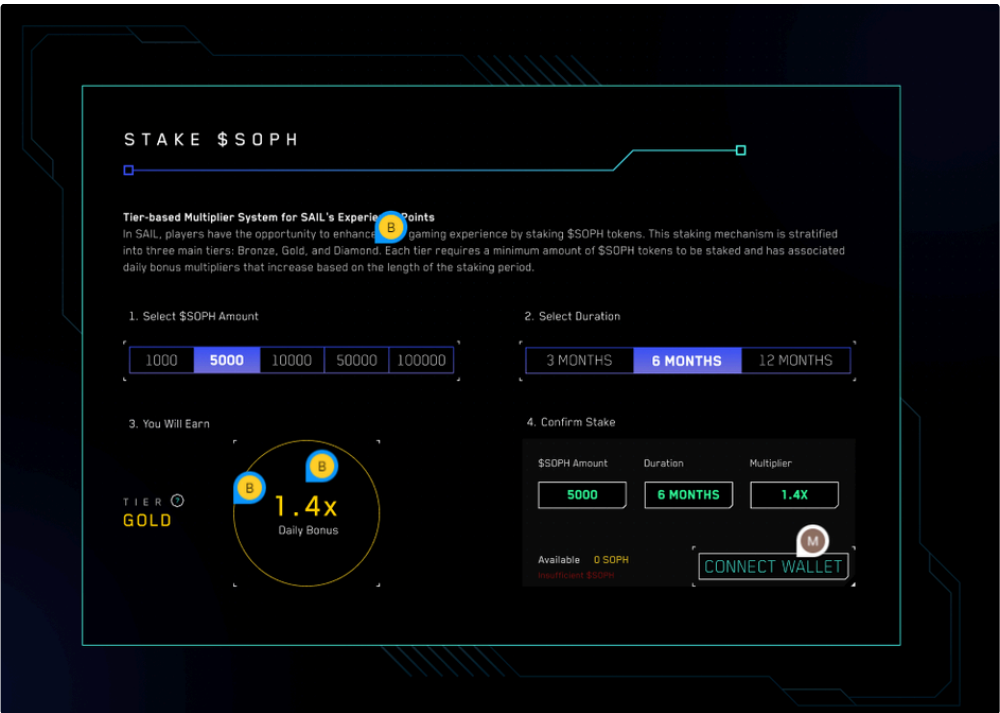
To create a new staking contract , deployer must specify the URI for each of the staking durations!

Actions a user can perform:

- Choose a staking minimum contract,

each "select soph amount" is a different contract address

- **Stake** (for a period of 3 / 6 / 12 months) ;
- **Restake** (stake function again for different period, after the chosen period is over) ;
- **Unstake** (after the chosen period) and receive an account bound NFT;

Smart Contracts

Ethereum	Testnet Goerli

Staker Contract Functions

```
function stake(uint256 _timeType)
```

- approve `amount_to_stake` first the chosen contract;

```
//approve
token.approve(address(stakeBoosterA), 100);
token.approve(address(stakeBoosterB), 100);
token.approve(address(stakeBoosterC), 100);
```

- `uint256 _timeType` : 7,958,400 = 3m , 15,916,800 = 6m , 31,536,000 = 12m;

function upgradeStake(uint256 _timeType)

- `uint256 _timeType` : 15,916,800 = 6m , 31,536,000 = 12m;
- reverts if trying to upgrade to a lesser **_timeType** than current

Note:

- when doing **upgrade** stake time, time is incremented not reset.
- If wants to increase tokens, user must a new staking period in a different tier contract.

function unstake(address _receiver)

- **_receiver is address of sentience module,** although this is not a rule in the contract, frontend must pass a valid sentience module to claim this.
- when *stakes* to the end of period and proceeds to *unstake* user gets NFT (they get forever & soul-bounded aka non-transferable)
- user still can choose to upgrade stake

Query Information:

→ To get end timer of a user staking : **wallet_stakeEndTimer(address _wallet)** , returns unix timestamp

→ To get user stake type, i.e 3m or 6m or 12m: **wallet_stakeTimeType(address _wallet)** , returns unix timestamp

→ To get minimum amount use, `minimumAmount` ()

→ To get token uri `function uri(uint256 _id)` , _id must be 3, 6 or 12 as these are the only valid ids