利用Remix编译智能合约

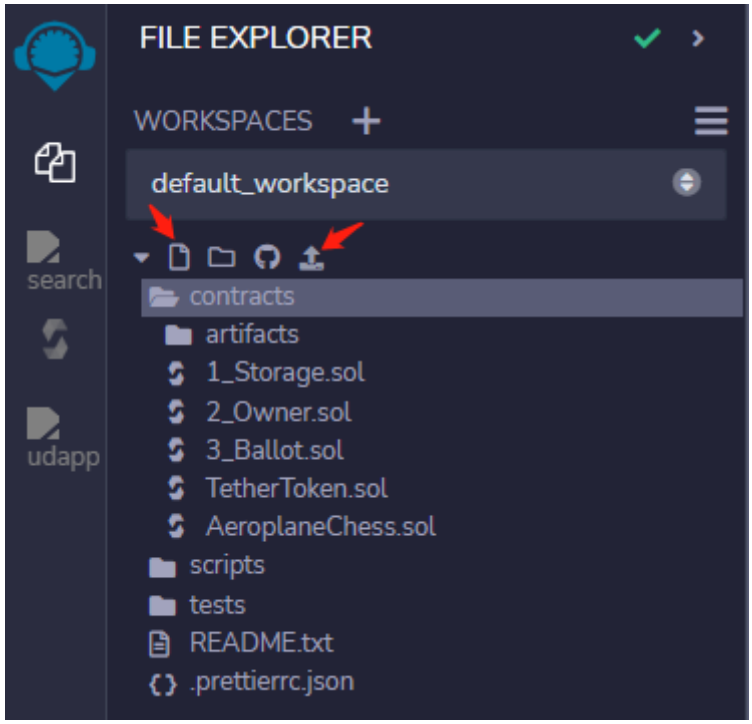
一、介绍

Remix集成开发环境（Remix IDE）可供各种知识水平的用户用于智能合约的整个开发过程。它不需要设置，促进了快速开发周期，并拥有丰富的插件和直观的GUI。该IDE有两种形式（网络应用程序或桌面应用程序）和VSCode扩展。

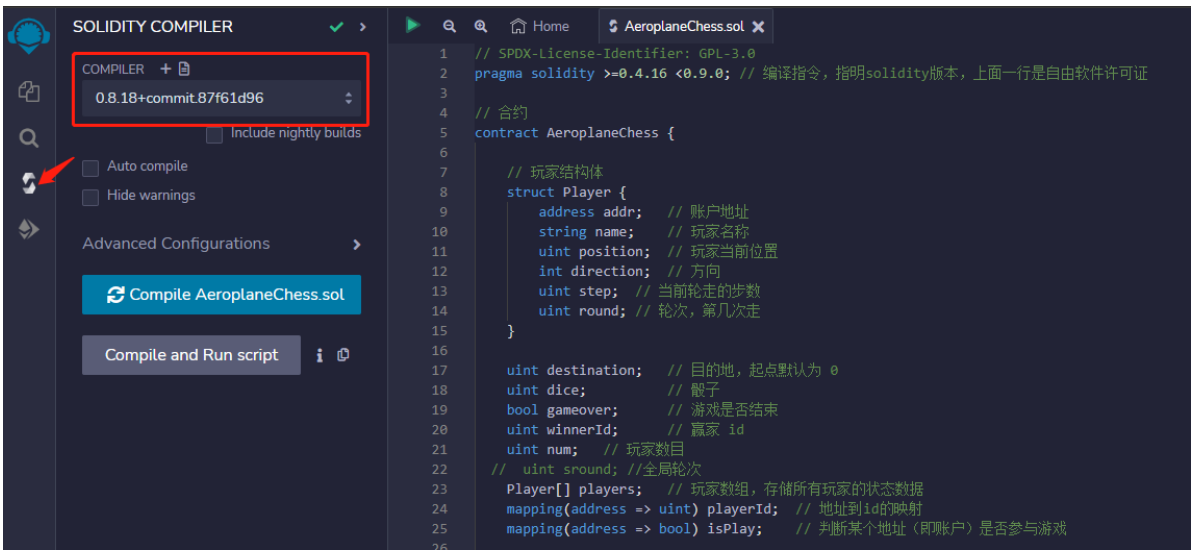
网站

二、如何编译一个智能合约（以飞行棋为例）

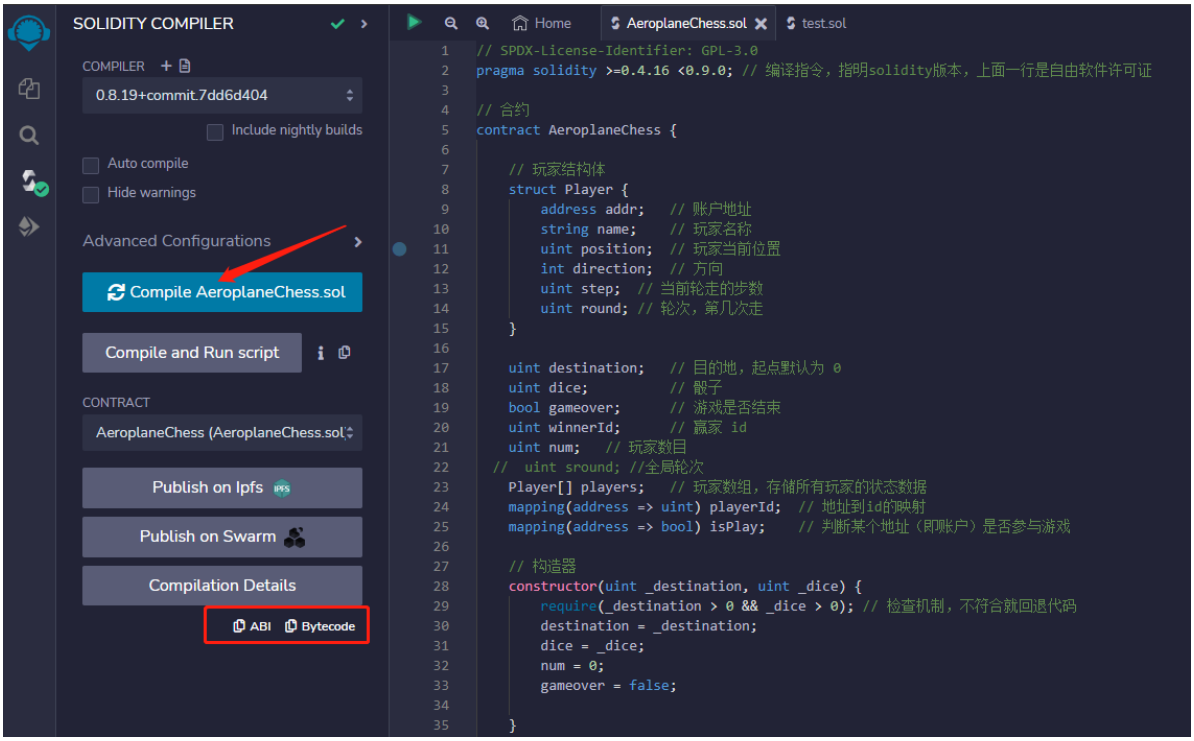
1. 在`contracts`目录下选择新建文件或者上传文件，创建一个`AeroplaneChess.sol`



2. 在`SOLIDITY COMPILER`页面选择合适的编译器版本



3. 点击`Compile AeroplaneChess.sol`编译智能合约之后，点击下方的`ABI`和`Bytecode`可以将相关的数据复制到剪切板


SOLIDITY COMPILER
COMPILER
0.8.19+commit.7dd6d404
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile AeroplaneChess.sol
Compile and Run script
CONTRACT
AeroplaneChess (AeroplaneChess.sol
Publish on Ipfs
Publish on Swarm
Compilation Details
ABI
Bytecode
Home
AeroplaneChess.sol
test.sol
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0; // 编译指令，指明solidity版本，上面一行是自由软件许可证
// 合约
contract AeroplaneChess {
// 玩家结构体
struct Player {
address addr; // 账户地址
string name; // 玩家名称
uint position; // 玩家当前位置
int direction; // 方向
uint step; // 当前轮走的步数
uint round; // 轮次，第几次走
}
uint destination; // 目的地，起点默认为 0
uint dice; // 骰子
bool gameover; // 游戏是否结束
uint winnerId; // 赢家 id
uint num; // 玩家数目
// uint sround; //全局轮次
Player[] players; // 玩家数组，存储所有玩家的状态数据
mapping(address => uint) playerId; // 地址到id的映射
mapping(address => bool) isPlay; // 判断某个地址（即账户）是否参与游戏
// 构造器
constructor(uint _destination, uint _dice) {
require(_destination > 0 && _dice > 0); // 检查机制，不符合就回退代码
destination = _destination;
dice = _dice;
num = 0;
gameover = false;
}