Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : GameFi Idea Jam – Brainstorming Blockchain Games

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

```
1.Initialize brainstorming environment using group discussions or online collaboration tools.
2.Identify the concept of GameFi (Game + DeFi) — where blockchain integrates gaming and finance.
3.Research existing blockchain-based games (e.g., Axie Infinity, The Sandbox, Gods Unchained).
4.Define the game concept (genre, storyline, and blockchain integration).
5.Design tokenomics – create a model for in-game currency, NFT assets, and reward systems.
6.Draft smart contract logic for NFT minting, player rewards, and asset trading.
7.Sketch the game economy flow showing interactions between players, tokens, and marketplace.
8.Present the final GameFi concept highlighting its features and blockchain benefits.
```

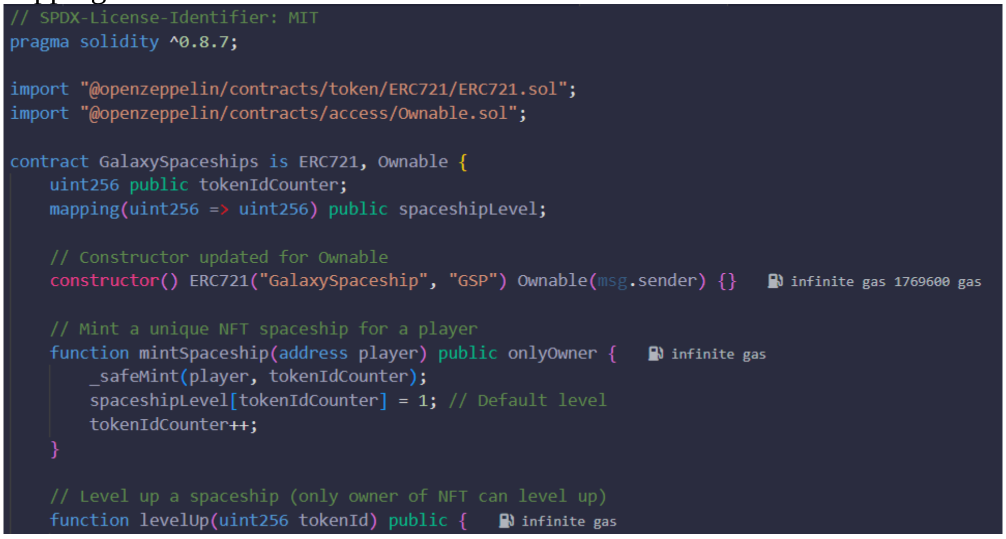
* Software used

1.Remix IDE
2.Metamask Wallet
3.OpenZeppelin Contracts
4.Solidity ^0.8.7

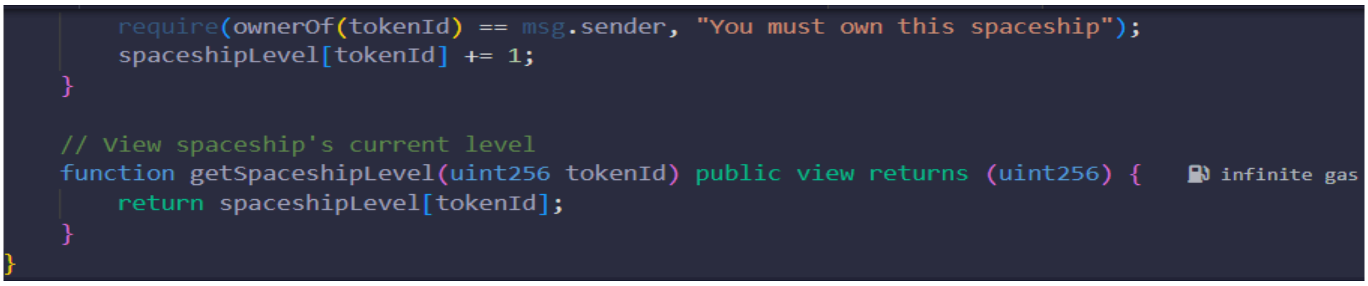
*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Testing Phase: Compilation of Code (error detection)

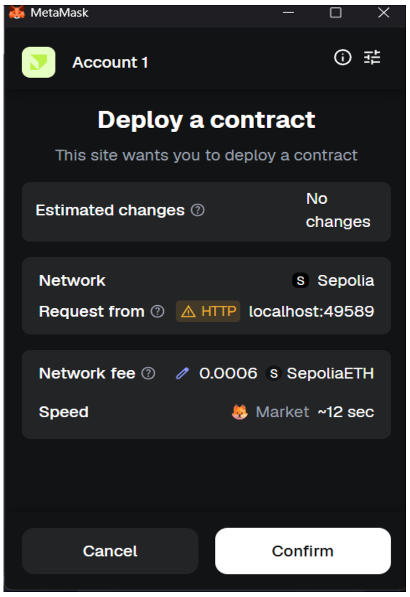
1.The smart contract was compiled successfully and deployed on the Sepolia Test Network using Remix IDE.
2.A new spaceship NFT was created by invoking the mintSpaceship() function from the contract owner's account.
3.The minted NFT was visible under the connected wallet's NFTs section in MetaMask, confirming successful minting.
4.The levelUp() function was executed to upgrade the spaceship's level and verify ownership restrictions.
5.The getSpaceshipLevel() function displayed accurate level data, ensuring that the contract logic and mappings worked as intended.



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract GalaxySpaceships is ERC721, Ownable {
    uint256 public tokenIdCounter;
    mapping(uint256 => uint256) public spaceshipLevel;

    // Constructor updated for Ownable
    constructor() ERC721("GalaxySpaceship", "GSP") Ownable(msg.sender) {}    infinite gas 1769600 gas

    // Mint a unique NFT spaceship for a player
    function mintSpaceship(address player) public onlyOwner {    infinite gas
        _safeMint(player, tokenIdCounter);
        spaceshipLevel[tokenIdCounter] = 1; // Default level
        tokenIdCounter++;
    }

    // Level up a spaceship (only owner of NFT can level up)
    function levelUp(uint256 tokenId) public {    infinite gas
```



```
        require(ownerOf(tokenId) == msg.sender, "You must own this spaceship");
        spaceshipLevel[tokenId] += 1;
    }

    // View spaceship's current level
    function getSpaceshipLevel(uint256 tokenId) public view returns (uint256) {    infinite gas
        return spaceshipLevel[tokenId];
    }
}
```



MetaMask

Account 1

Deploy a contract

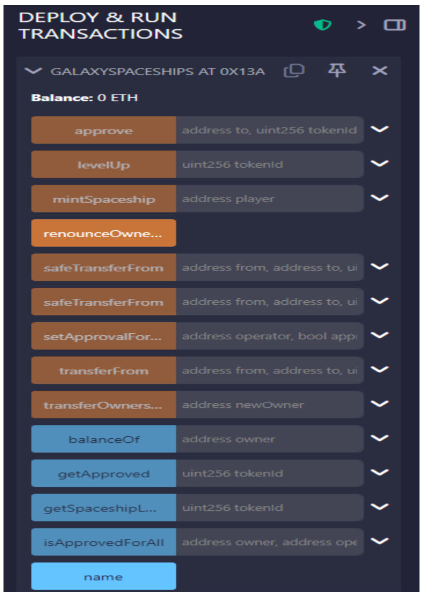
This site wants you to deploy a contract

Estimated changes: No changes

Network: Sepolia
Request from: HTTP localhost:49589

Network fee: 0.0006 SepoliaETH
Speed: Market ~12 sec

Cancel | Confirm

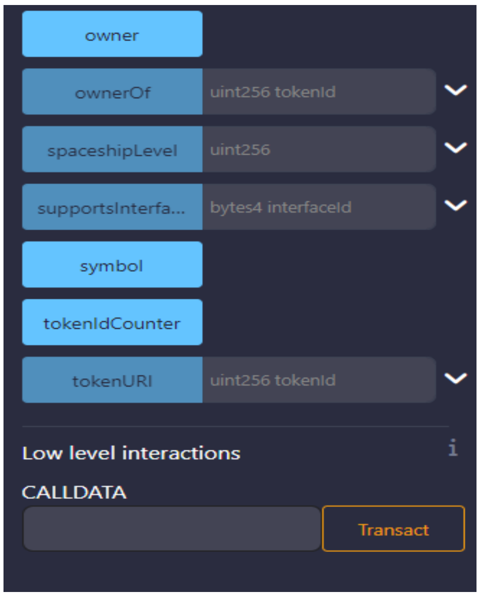


DEPLOY & RUN TRANSACTIONS

GALAXYSPACESHIPS AT 0X13A

Balance: 0 ETH

approve	address to, uint256 tokenId
levelUp	uint256 tokenId
mintSpaceship	address player
renounceOwne...	
safeTransferFrom	address from, address to, ui
safeTransferFrom	address from, address to, ui
setApprovalFor...	address operator, bool appi
transferFrom	address from, address to, ui
transferOwners...	address newOwner
balanceOf	address owner
getApproved	uint256 tokenId
getSpaceshipL...	uint256 tokenId
isApprovedForAll	address owner, address ope
name	



owner	
ownerOf	uint256 tokenId
spaceshipLevel	uint256
supportsInterfa...	bytes4 interfaceId
symbol	
tokenIdCounter	
tokenURI	uint256 tokenId

Low level interactions

CALLDATA

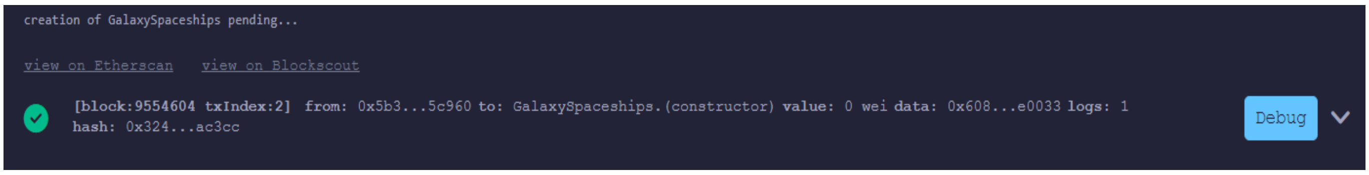
Transact

Page No...........

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

1.The GalaxySpaceships contract was compiled and deployed on Remix using the Sepolia test network.
2.A new NFT spaceship was minted for a specific wallet address through the mintSpaceship() method.
3.The NFT was confirmed to appear under the MetaMask wallet's NFT section after deployment.
4.The levelUp() function was executed successfully to increase the spaceship's level.
5.Output from getSpaceshipLevel() accurately reflected updated levels, validating contract functionality.



```
creation of GalaxySpaceships pending...

view on Etherscan    view on Blockscout

[block:9554604 txIndex:2] from: 0x5b3...5c960 to: GalaxySpaceships.(constructor) value: 0 wei data: 0x608...e0033 logs: 1
hash: 0x324...ac3cc
```

* Observations

1.The smart contract successfully created and managed NFT spaceships with correct ownership and level tracking.

2.All functions executed as expected, validating minting, upgrading, and retrieval of spaceship data on the blockchain.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Regn. No. :

Signature of the Faculty:

****As applicable according to the experiment. Two sheets per experiment (10-20) to be used.***