PROJECT: DARK RALLY BLOCKCHAIN

To integrate Plop Studio's Dark Rally game into Blockchain technology, three Smart Contracts (SC) were developed:

1) **DarkRallyNFT**, for minting and transfer NFT tokens and other related operations.

2) **DarkRallySale**, which handles the management of purchases of the mentioned NFTs by users.

3) **DarkRallyMktPlace**, which manages the Dark Rally's MarketPlace where NFT owners can trade their respective tokens.

This document is a summary of technical information for each of the developed SCs.

Document version: 1.02

SMART CONTRACT #1
SUMMARY

Name of Contract: DarkRallyNFT

Description: This upgradeable Smart Contract (SC) take care of all the required NFTs, working under the ERC-1155 standard. This SC does not handle money. Purchases are made from the RallySale contract.

- NFToys
- NFT Tickets to compete in Dark Rally Multiplayer Sessions
- NFT Trophies for Multiplayer competitions
- NFT Vehicles for Dark Rally player

In addition, in the case of Tickets, it will include a validation of their expiration date before minting

BASIC TECHNICAL INFO	
Standard	ERC-1155
Features	Initializable, Upgradeable (UUPS), Pausable, AccessControl, Burnable
Roles	Bussiness, Admin, Pauser, Upgrader
DEPLOYMENT:	
Testnet	Polygon Mumbai

MAIN FUNCTIONS AVAILABLE

Function	Input Data	Output Data	Notes
registerNewTypeofNft() To register new type of NFT's like vehicles, toys, etc.	uint256 tokenId, string nameOfNft, string category, string metadataHashIpfs, uint256 maxSupply, bool askDateForMint, uint256 validUntil	-	Emits a {RegisterNewTypeOfNFT} event (see details below)

deleteRegisterOfTypeOfNft() To delete registration of a type of NFT	uint256 tokenId		A mandatory condition is not to have minted any token of that type.
getTokensList() To get an array of all registered NFT's	-	uint256[] tokenList	Returns and array like this: [101,102.301,302]
changeMetadataHashOfNft() Permits change of Metadata CID	uint256 tokenID, memory metadataHashIpfs	-	Only by Business role accounts
changeMaxSupplyOfNft() Permits change of Metadata max supply of tokens	uint256 tokenID, uint256 maxSupply	-	Only by Business role accounts
mint() To mint one or more NFT token	address account, uint256 tokenId uint256 amount,	-	Only by Minter role accounts In case of Tickets, validate the expiration date before minting Only when not paused. Emits a {TransferSingle} event.
uri() To get the address of metadata information	uint256 tokenId	string uri	
getAssetsOfAccount() To get all the NFT's owned by an account, based in a list of tokens	address account, uint256[] tokensList	uitn256[] listOfAssets	The input and output are arrays. In example: Input: [101,202,301,401,402] Output [2,0,1,1,3] *Means that account owned 2 tokens 101, no token 202. and so on, respectively
balanceOf() Returns the amount of a type of tokens that owns an account	address account, tokenId (uint256)	uint256 amount	
safeTransferFrom() Transfer an amount of one type of token from one account to another	address from, address to, uint256 tokenId uint256 amount		Emits a {TransferSingle} event.
pause() Pause the transactions of the Smart Contract	–	–	Only by Pauser role accounts
unpause() Unpause the transactions of the Smart Contract	–	–	Only by Pauser role accounts
burn() To Burn an amount of a type of tokens from an account	address account, uint256 tokenId uint256 amount	–	Caller must own `tokenId` or be an approved operator.

Main Functions Details:

Function	Details	Notes
registerNewTypeofNft()	This function allows to register a new type of NFT, like tickets, toys, skins, trophies and vehicles. To mint an NFT it is necessary that token has been previously registered. It needs the input of six arguments: tokenId -> like 101 nameOfNft, like DRchamps-serie24-A category: Tickets metadataHashIpfs: Qmegv2LoRAKssd2ReJfHLo7Gya7mGc6wrT7BMCgVk9gfJX maxSupply: 30,000 askDateForMint: yes (verify date before Mint) validUntil: 1719748799	

SMART CONTRACT #2 **SUMMARY**
Name of Contract: **DarkRallySale**
Description: This upgradeable Smart Contract (SC) handle all NFT purchases and must be used by a front-end, either from a mobile application or from a web application. It contains the prices of each NFT token, as well as the necessary business logics. Purchases will be made in USDC. To do so, the SC will have identified all the tokens of the RallyNFT Smart Contract and their respective prices in USDC.

BASIC TECHNICAL INFO	
Standard	-
Features	Initializable, Upgradeable (UUPS), Pausable, AccessControl
Roles	Bussiness, Admin, Pauser, Upgrader,
DEPLOYMENT:	
Testnet	Polygon Mumbai

MAIN FUNCTIONS AVAILABLE

Function	Input Data	Output Data	Notes
purchaseNftByTokenId() Purchase an amount of one type of tokens to an account.	uint256 tokenId uint256 amount		Needs the approval of USDC for this SC Emits a DeliverNFT event (see details below)

setNftPrice() To set the Price of NFT Tokens, in batch mode entering arrays of tokenIds and prices.	uint256[] tokenIds uint256[] prices	-	Only by Business role accounts The input is an array. In example: Input tokenIds: [101,202] Input Prices: Output [7000000,8000000] *Means that tokenId'101's price will be 7000000 and tokenId price for 202,wiil be 800000.
setNftScAddress() To change the Address of DarkRallyNFT SC	address newNftSCAddress	-	Only by Business role accounts
setUsdcCoinScAddress() To change the Address of USDC SC	address newAddress	-	Only by Business role accounts
setCompanyWalletAddress() To change the Address of the Company Wallet where coins USDC were deposited.	address newAddress	-	Only by Business role accounts
setFeeWalletAddress() To change the Address of the Wallet where fee coins USDC were deposited.	address newAddress	-	Only by Business role accounts
pause() Pause the transactions of the Smart Contract	–	–	Only pauser_roler
unpause() Unpause the transactions of the Smart Contract	–	–	Only pauser_role

Main Functions Details:

Function	Details	Notes
purchaseNftByTokeId()	This external function allows to buy, on behalf of a user account, a certain amount of a type of token. To do this, prior authorization will be required so that this contract can spend the user's USDC It needs two arguments: tokenId and amount In principle, it will validate the following before purchase the tokens:: 1) That the tokenIDs are still available to be minted 2) That the user has given allowance to this contract in a sufficient amount of USDC 90% of the USDC coins from each purchase is deposited in the company's wallet, and 10% in the fee wallet This SC is connected to DarkRallyNft SC to mint the NFT, which is transferred to the buyer's wallet	

SMART CONTRACT #3 SUMMARY
Name of Contract: DarkRallyMktPlace
Description: This upgradeable Smart Contract (SC) manages the Dark Rally's MarketPlace where NFT owners can trade their respective tokens by entering the respective information, which includes the prices they want for each NFT token, as well as the necessary business logic. Purchases will be made in USDC. The owners set the price they want and if any buyer makes the purchase, 95% of the USDC coins will be deposited in the owner's account and the remaining 5% will be distributed as follows: 4.5% to the Company's wallet 0.5% to the Fee wallet

BASIC TECHNICAL INFO	
Standard	-
Features	Initializable, Upgradeable (UUPS), Pausable, AccessControl
Roles	Bussiness, Admin, Pauser, Upgrader
DEPLOYMENT:	
Testnet	Polygon Mumbai

MAIN FUNCTIONS AVAILABLE

Function	Input Data	Output Data	Notes
createSaleOffer() Where an account can create a sale offer for a type of NFT that owns, by entering the price and quantity desired.	uint256 tokenId, uint256 price, uint256 quantity	-	Needs the approval to transfer NFTs for this SC
purchaseNFT() Where a buyer account can buy one or more offered NFT with their USDC	uint256 tokenId, address owner, uint256 price, uint256 quantity	-	Only when the SC if not Paused Needs the approval of USDC for this SC Emits {PurchaseNft} event
getSalesList() To get all the sales offers of the MarkePlace, returning an array		string[] offerSalesList	Returns an array containing the information of sales offers: tokenId, owner address, price and quantity of tokens
updatePriceAndQuantity() Where an account can update their sale offer information	uint256 tokenId, uint256 newPrice, uint256 newQuantity	-	

removeSaleOffer() Where an account can remove their sale offer	uint256 tokenId	-	
setNftScAddress() To change the Address of DarkRallyNFT SC	address newNftSCAddress	-	Only by Business role accounts
setUsdcCoinScAddress() To change the Address of USDC SC	address newAddress	-	Only by Business role accounts
setCompanyWalletAddress() To change the Address of the Company Wallet where coins USDC were deposited.	address newAddress	-	Only by Business role accounts
setFeeWalletAddress() To change the Address of the Wallet where fee coins USDC were deposited.	address newAddress	-	Only by Business role accounts
pause() Pause the transactions of the Smart Contract	–	–	Only pauser_roler
unpause() Unpause the transactions of the Smart Contract	–	–	Only pauser_role

APPENDICE #1

TOKENS REQUIRED BY PLOP STUDIO:

NFT	Collection Names	Stock (tokens)	Notes
Toys	#01> Serie AA23-A	#1> 10000	
Tickets	#02> DRchamps-serie23-A #03> DRchamps-serie24-A	#2> 10000 #3> 50000	Valids for one use Valids until and Price: #02> Dec31th, 2023. Price: U$DT 7 #03> Jun30th, 2024. Price: U$DT 9
Trophies	#04> DRchamps-trophy-serie23-A-300 #05> DRchamps-trophy-serie23-A-800 #06> DRchamps-trophy-serie23-A-1000	3 each one	Value: #4> 300 U$DT #5> 800 U$DT #6> 1000 U$DT
Vehicles	#07> DRchamps-vehicles-serie23-A-car01 #08> DRchamps-vehicles-serie23-A-car02 #09> DRchamps-vehicles-serie23-A-truck01 #10> DRchamps-vehicles-serie23-A-truck02 #12> DRchamps-vehicles-serie23-A-moto01 #12> DRchamps-vehicles-serie23-A-moto02 #13> DRchamps-vehicles-serie23-A-quad01 #14> DRchamps-vehicles-serie23-A-quad02 #15> DRchamps-vehicles-serie23-A-ssv01 #16> DRchamps-vehicles-serie23-A-ssv02	300 each one	(see section: Vehicles details)

Vehicles details: Max. Value Ingame (MVI) Limited EditionNFT(LEN)

NFT Vehicle	Name	MVI (Dark coins)	LEN (Dark Coins)	Parameters
Car01	Marathox	37,000	74,000	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, **Superpower**: 6/10
Car02	4x4 Herbhi	56,000	156,800	**Drive**:7/10, **Speed**: 7/10, **Acceleration**: 7/10, **Weight**: 7/10, **Superpower**: 7/10
Truck01	Thorment	37,000	55,500	**Drive**:8/10, **Speed**: 8/10, **Acceleration**: 8/10, **Weight**: 8/10, **Superpower**: 8/10
Truck02	Xpeedx	56,000	123,200	**Drive**:9/10, **Speed**: 9/10, **Acceleration**: 9/10, **Weight**: 9/10, **Superpower**: 9/10
Moto01	Geo Sol	37,000	74,000	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, Superpower: 6/10
Moto02	Fortaleza	56,000	156,800	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, **Superpower**: 6/10
Quad01	Ultra 7R7	37,000	74,000	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, **Superpower**: 6/10
Quad02	Go Salma	56,000	156,800	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, **Superpower**: 6/10
Ssv01	Nazkar	41,000	82,000	**Drive**:6/10, **Speed**: 6/10, **Acceleration**: 6/10, **Weight**: 6/10, **Superpower**: 6/10
Ssv02	Kavlov	56,000	162,400	**Drive**:9/10, **Speed**: 7/10, **Acceleration**: 8/10, **Weight**: 9/10, **Superpower**: 9/10