

# CarBorrow

## 实验要求

第二次作业要求（以下内容提交时可以删除）：

简易汽车借用系统，参与方包括：汽车拥有者，有借用汽车需求的用户

- 背景：ERC-4907 基于 ERC-721 做了简单的优化和补充，允许用户对NFT进行租借。
- 创建一个合约，在合约中发行NFT集合，每个NFT代表一辆汽车。给部分用户测试领取部分汽车NFT，用于后面的测试。

在网站中，默认每个用户的汽车都可以被借用。每个用户可以：

- 查看自己拥有的汽车列表。查看当前还没有被借用的汽车列表。
- 查询一辆汽车的主人，以及该汽车当前的借用者（如果有）。
- 选择并借用某辆还没有被借用的汽车一定时间。

上述过程中借用不需要进行付费。

- （Bonus）使用自己发行的积分（ERC20）完成付费租赁汽车的流程

## 如何运行

1. 在本地启动ganache应用。
2. 在 `./contracts` 中安装需要的依赖，运行如下的命令：

```
npm install
```

3. 在 `./contracts` 中编译合约，运行如下的命令：

```
npx hardhat compile
```

4. 配置 `./scripts/deploy.ts` 文件，部署合约并提前发放汽车用于测试

```
await borrowYourCar.CarNFT("0xf2A3818564973Ca9F128f8d892905f4cBf06555C");  
await borrowYourCar.CarNFT("0xf2A3818564973Ca9F128f8d892905f4cBf06555C");  
await borrowYourCar.CarNFT("0x9d379E4164Eca3163BA43E5C6da97eAb4084AC6f");
```

`""` 内替换为ganache中的地址

5. 在 `./contracts` 中运行 `./scripts/deploy.ts` 文件

```
npx hardhat run ./scripts/deploy.ts
```

6. 在 `./contracts` 将合约部署在ganache

```
npx hardhat run scripts/deploy.ts --network ganache
```

7. 在 `./frontend` 中安装需要的依赖，运行如下的命令：

```
npm install
```

8. 在 `./frontend` 中启动前端程序，运行如下的命令：

```
npm run start
```

## 功能实现分析

### 数据结构

Solidity中有 `car` 的结构体，和通过 `tokenId` 查找对应车数据的 `mapping cars` 和一个记录每个用户的车辆的 `mapping owners`，以及记录车数量的 `carCount` 和使用ERC20合约的 `maomaoToken`

```
struct Car {
    uint256 tokenId;
    address owner;
    address borrower;
    uint256 borrowUntil;
}

mapping(uint256 => Car) public cars; // A map from car index to its
information
mapping(address => uint[]) public owners;
uint256 public carCount = 0; // A counter to count how many cars are in the
contract
MaoMaoToken public maomaoToken;
constructor(string memory name_, string memory symbol_) ERC721(name_,
symbol_){
    maomaoToken = new MaoMaoToken("MaoMaoToken", "MMT");
}
```

React中创建了类似于Solidity中 `car` 的结构体

```
interface Car{
    tokenId: Uint256;
    owner: string;
    borrower: string;
    borrowUntil :Uint256;
}
```

使用 `useState` 定义了当前用户 `currentAccount`，用于储存当前用户所有车的 `ownedcars`，用于储存除当前用户所用的的其他车的 `othercars`，用于储存当前用户正在借用的车的 `borrowedcars`，以及一些用于函数传参的数据

```
const [currentAccount, setCurrentAccount] = useState("");
const [ownedcars, setownedcars] = useState([] as Car );
const [othercars, setothercars] = useState([] as Car );
const [borrowedcars, setborrowedcars] = useState([] as Car );
const [carTokenId, setcarTokenId] = useState("");
const [duration, setduration] = useState("");
const [accountBalance, setAccountBalance] = useState(0);
```

## NFT(汽车)发行

Solidity中主要通过 `CarNFT` 函数实现，会为参数地址 `Car_to_Owner` 分配一量车，本实验中会直接将 `carCount` 即第几辆车设为 `car.tokenId`，该函数会将 `tokenId` 设置为 `carCount`，将 `owner` 设置为 `Car_to_Owner`，该项目中通常是 `msg.sender`，并未提供在一个用户连接下为另一个用户分配车的功能，同时会将该辆车的 `tokenId` push进 `owners[Car_to_Owner]`，用于记录和管理该地址的车，最后将 `carCount` 加一

在前端布置中主要通过一个展示当前连接的address和一个 `Connect wallet` 的按钮，这些功能都基于助教的 demo 不再介绍，分配车辆主要通过一个 `Come car` 按钮，来为当前连接的用户地址分配一辆车

`NFT` 是一个react的钩子函数，点击按钮即会调用该函数，该函数会调用 `BorrowYourCarContract` 的 `CarNFT` 函数为当前连接用户分配一辆车，同时更新 `ownedcars` 数组，该数组用于储存当前连接用户的车

## 查看已拥有车列表

在 `BorrowYourCar` 中提供了一个函数用于返回 `owner[msg.sender]` 中所对应的所有 `cars[]` 里的车辆的数组，

在前端提供一个 `getOwndedCars` 函数用于获取合约返回的 `car` 数组，并提供按钮以便于刷新

在前端维护一个 `ownedcars` 的数组用于储存该用户的车，并通过 `table` 组件展示出来

## 查看已借用/可借用的车列表

在 `BorrowYourCar` 中提供了一个函数用于返回 `car.owner` 不是 `msg.sender` 的所有的 `cars[]` 的数组，在前端用 `othercars[]` 维护，然后在前端通过 `filter` 方法与当前时间戳对比，判断这辆车是否已经到期或者可以被租截，或者正在被租借

同样用 `table` 组件展示和一个刷新的 `button`

因为在Solidity中时间戳是区块创立的时间戳，所有在前端进行比较

## 借车与还车

输入要借的时间，当前便于测试用秒为单位，点击 `borrow` 按钮，会调用 `BorrowYourCar` 的 `borrow` 函数，检测该车是否存在，且是否发送者自己的车，符合条件后，更新对应 `car` 的 `borrower` 和 `borrowUntil`

点击 `back` 按钮同样调用 `BorrowYourCar` 的 `back` 函数，检测该车是否存在，是否已经被归还即 `borrower==msg.sender`，符合条件后更新对应 `car` 的 `borrower` 和 `borrowUntil`

再刷新已借用/可借用的车列表，可以看到车辆的变化

## 查询一辆车

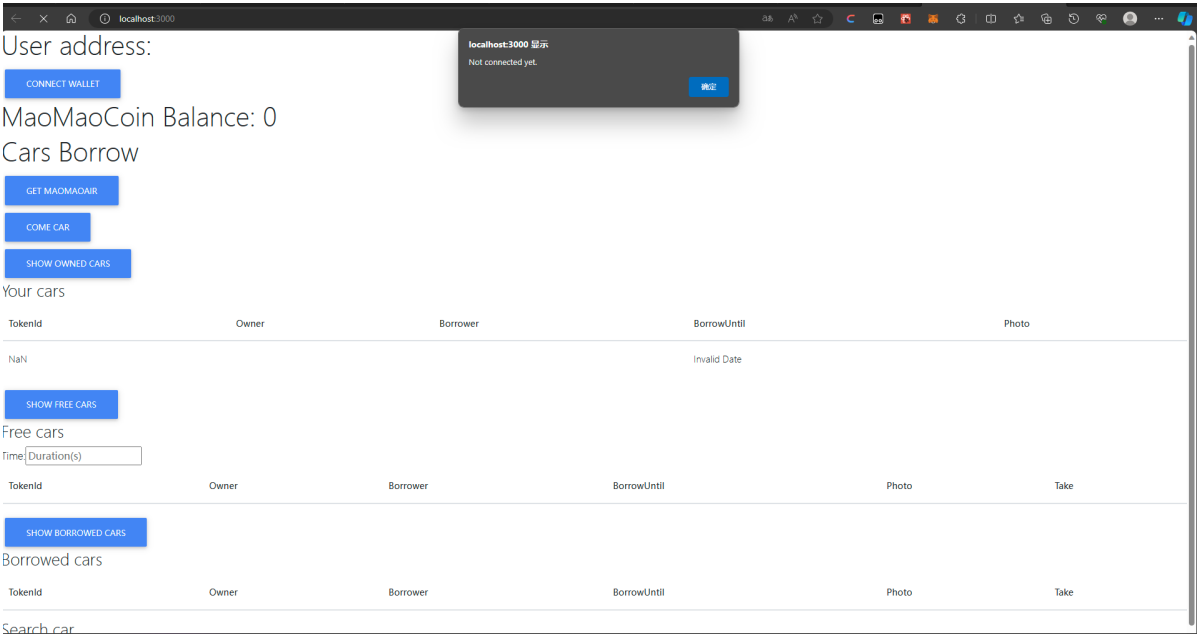
在前端输入想要查找的车的 `tokenId`，如果不合法会弹窗，合法则同样使用 `othercars[]` 在前端筛选出并显示

## 使用自己发行的猫猫币付费借车

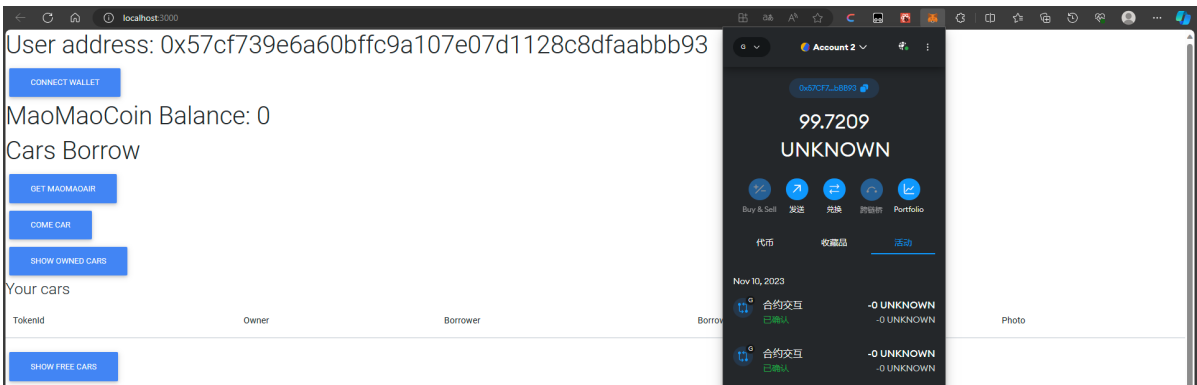
使用继承 `ERC20` 的 `MaoMaoToken` 合约，当用户需要支付费用时，需要先向 `BorrowYourCar` 合约授权，再在合约进行费用扣除

# 项目运行截图

进入系统，未连接状态



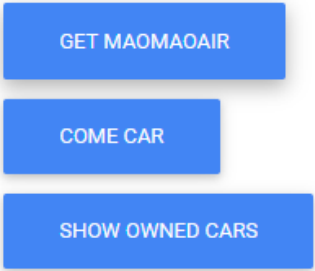
使用小狐狸切换导入的ganache账户，点击 CONNECT WALLET，连接用户



领取猫猫空投

MaoMaoCoin Balance: 1000000



Cars Borrow






点击两次 COME CAR，领取两辆车

Show owned cars

Your cars

Tokenid	Owner	Borrower	BorrowUntil	Photo
3	0x57CF739eA608fBC9a107e07d1128C8dFaabB893	0x00	Not being borrowed yet	
4	0x57CF739eA608fBC9a107e07d1128C8dFaabB893	0x00	Not being borrowed yet	

可以看到 FreeCars 已经展示出配置文件已经分配好的车

Free cars					
Time: <input type="text"/>					
TokenId	Owner	Borrower	BorrowUntil	Photo	Take
0	0xf2a3818564973Ca9f128f8d892905f4cBf065550	0x00	Not being borrowed yet		<a href="#">BORROW</a>
1	0xf2a3818564973Ca9f128f8d892905f4cBf065550	0x00	Not being borrowed yet		<a href="#">BORROW</a>
2	0x9d379E4164ECa3163BA43ESC6da97eAb4084AC6f	0x00	Not being borrowed yet		<a href="#">BORROW</a>

借用Id为0的车60秒，Id为1的车120秒

当前猫猫币余额

User address: 0x57cf739e6a60bffc9a107e07d1128c8dfaabbb93

CONNECT WALLET

MaoMaoCoin Balance: 1000000

## Cars Borrow

GET MAOMAOAIR

COME CAR

[SHOW OWNED CARS](#)

借用后余额

User address: 0x57cf739e6a60bffc9a107e07d1128c8dfaabbb93

CONNECT WALLET

MaoMaoCoin Balance: 999820

Cars Borrow

GET MAOMAOAIR

COME CAR


SHOW OWNED CARS

借用后车辆查看

SHOW FREE CARS



Free cars

Time

TokenId	Owner	Borrower	BorrowUntil	Photo	Take
2	0x9d379e4164Eca3163BA43E5C6da97e4b40844C6f	0x00	Not being borrowed yet		BORROW

SHOW BORROWED CARS

Borrowed cars

TokenId	Owner	Borrower	BorrowUntil	Photo	Take
0	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BfC9a107e07d1128C8dFaabBB93	Nov 10, 2023, 08:24:23 PM		BACK
1	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BfC9a107e07d1128C8dFaabBB93	Nov 10, 2023, 08:25:39 PM		BACK

过60秒后，Id为0的车已经变为可用

SHOW FREE CARS


Free cars

Time

TokenId	Owner	Borrower	BorrowUntil	Photo	Take
0	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BfC9a107e07d1128C8dFaabBB93	Not being borrowed yet		BORROW
2	0x9d379e4164Eca3163BA43E5C6da97e4b40844C6f	0x00	Not being borrowed yet		BORROW



SHOW BORROWED CARS

Borrowed cars

TokenId	Owner	Borrower	BorrowUntil	Photo	Take
1	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BfC9a107e07d1128C8dFaabBB93	Nov 10, 2023, 08:25:39 PM		BACK


再点击归还id为1的车

Time:

TokenId	Owner	Borrower		Photo	Take
0	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BFC9a107e07d1128C8dFaab8B93	Not being borrowed yet		<button>BORROW</button>
2	0x9d379E4164ECa3163BA43E5C6d97eAb4084AC6f	0x00	Not being borrowed yet		<button>BORROW</button>




SHOW BORROWED CARS

Borrowed cars

TokenId	Owner	Borrower	BorrowUntil	Photo	Take
1	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BFC9a107e07d1128C8dFaab8B93	Nov 10, 2023, 08:25:39 PM		<button>BACK</button>

Free cars

Time:


TokenId	Owner	Borrower	BorrowUntil	Photo	Take
0	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x57CF739e6A60BFC9a107e07d1128C8dFaab8B93	Not being borrowed yet		<button>BORROW</button>
1	0xf2A3818564973Ca9F128f8d892905f4cBf06555C	0x00	Not being borrowed yet		<button>BORROW</button>
2	0x9d379E4164ECa3163BA43E5C6d97eAb4084AC6f	0x00	Not being borrowed yet		<button>BORROW</button>

所有车都可租借

查询id为2的车

Search car

Go  SEARCH

TokenId	Owner	Borrower	BorrowUntil	Photo
2	0x9d379E4164ECa3163BA43E5C6d97eAb4084AC6f	0x00	Not being borrowed yet	

## 参考内容

- 课程的参考Demo见：[DEMOS](#)。
- ERC-4907 [参考实现](#)

如果有其它参考的内容，也请在这里陈列。