

Group 7

Piotr Kram - @peternal_flame

I coded up a number of scripts:

- for casting votes
- for checking the voting power
- for delegating votes
- for deploying the token and ballot
- for minting\giving voting tokens
- for querying the results
- for transferring voting tokens
- for updating target block number

I added a functionality to update the target block number since in my implementation the creation of the ballot and the token were bonded together since I wanted to access the token only through the ballot contract. But that created complications since I didn't know the exact target block number at the time of creation of the contract and calling `getPastVotes` for a target block number in the future generates an error ("ERC20Votes: future lookup"). And having a target block number too low risks your votes not to count even though you think they will ("TokenizedBallot: trying to vote more than allowed" error when casting votes or "ERC20: transfer amount exceeds balance" error when transferring tokens). This issue could be solved in a number of ways:

- decoupling token and ballot creation and creating the ballot only once the token operations are done
- adding a functionality to modify the target block number by an administrator of the ballot
- adding complexity and using `getVotes` instead of `getPastVotes` for target block numbers in the future

I opted out for the second solution as it allows for the ballot to be reusable although if we want it to be reusable we should also start burning tokens while voting, which is not currently implemented.

The tokenized ballot and all of its successful transactions:

<https://sepolia.etherscan.io/address/0xa20a7dbd3510b661d24f865444885ede2a0ba804>

The voting token and all of its successful transactions:

<https://sepolia.etherscan.io/address/0xbbf6e57c721b519183e585635dec3cd22f67992b>