Jeremias & Alex

1 Introduction

The prisoner's dilemma (https://en.wikipedia.org/wiki/Prisoner's_dilemma) is a famous scenario from game theory. It is about two partners in crime who are being interrogated by the police. They are both offered the following deal:

- If one betrays his partner and cooperates with the police, he goes free, and the accomplice goes to prison for ten years.
- If neither talks, they go to prison for two years each.
- However, if both cooperate with the police and they betray each other, they go to prison for five years.

Note that they are not allowed to communicate with each other.
We adapted this scenario as a betting game on the blockchain. You call contract functions to state your move (stay silent or betray your partner).

- If you win and successfully betray the other player, you receive your initial investment and a bonus 70% on top. The loser loses 90% of their investment.
- If you cooperate with each other, you each get your initial investment and 40% on top.
- If you betray each other, you each receive 80% of your investment back.

2 How to Play

Step 1: Set up your Truffle CLI and obtain a reference to our contract. Use `truffle console --network sepolia` and `let pd = await PrisonersDilemma.deployed()` to do this, like you would with any other contract. Please do it in your own project with your own wallet and API key, because we aren't sharing our `secrets.json` :)

Step 2: Choose how much ether you want to bet. There is a maximum amount of ether you can wager, so that the contract's balance won't fluctuate too much or fall too low. Use `(await pd.getMaxWager()).toString()` to see the maximum amount you can bet at the moment. Important: The amount is given in *wei*. One *eth* is made up of 10^{18} *wei*. You can use the converter at https://eth-converter.com/ to make this step easier.

Step 3: Choose what move you will play (stay silent, or cooperate with the police). Staying silent is a value of 1, while betraying is 2.

Step 4: Play your move. Use `pd.playUnsafe(move, { value: wager })`, with `move` as the value you chose, and `wager` as the amount you want to bet (in *wei*). The transaction can take a while to complete. Example: `pd.playUnsafe(1, { value: 1000000000000000000 })` will mean you cooperate with your partner, and your bet is one `ether`. Note: The function is called `playUnsafe` because it's actually a wrapper around a more secure API that hides your moves from other players, but this one is easier to use. Ask us if you want the full explanation.

Step 5: If you are the first to go in the current game, wait until somebody else has completed the previous step.

Step 6: Call the payout function to distribute the rewards. This needs to be done only once for each 2-player game. Use `pd.payout()`.

Step 7: Wait until the transaction completes and you receive your payout. You can check the contract on the block explorer if you want to look closer on https://sepolia.etherscan.io/. The contract's address is 0x51338F5071d714Ff446B381C62a8c0D1e3a16404.