

Date: 13-01-2022

Student: Rob Honig

Course: CP108 Plutus/Haskell I

Problem statement: Most African governments are trying different measures to eliminate *corruption, bad governance, mismanagement* and *lack of accountability* in their countries but these efforts are mostly unsuccessful. As a tool for change, *blockchain* can help in solving some of these governance issues plaguing Africa.

Task: With this in mind, think of a way you can implement a smart contract that can help eliminate any of these ills, then implement it using Plutus.

Possible solution: Giving access to funds without using the so called “Middleman”. We use “Smart Contracts” for (financial) aid or microcredits, powered by the Cardano Blockchain. Lock Ada funds on the blockchain, redeem funds with the correct redeemer.

In this way no intermediary is used. This keeps human error to a minimum. Based on predetermined rules written in Haskell and Plutus.

- No middlemen involved;
- Low transaction costs;
- 24/7 available;
- Safe, secure and fast

Script: The *Gift.hs* can be used for this exam and can be tested at the Plutus Playground;
<https://playground.plutus.iohkdev.io/>

The screenshot shows the Plutus Playground web interface. At the top, there's a navigation bar with links for 'Getting Started', 'Tutorials', 'API', and 'Privacy'. Below this, a 'Demo files' section lists various scripts like 'Hello.world', 'Starter', 'Game', 'Vesting', 'Crowd Funding', and 'Error Handling'. The 'Vesting' script is selected. The main area is the 'Editor', which displays a Haskell script for a vesting scheme. The script includes imports for Control, Data, Ledger, and Plutus, and defines a vesting scheme as a PLC contract. The script is 25 lines long. At the bottom of the editor, it says 'Compilation successful'. The footer of the page includes links to 'cardano.org' and 'iohk.io', the copyright notice '© 2020 IOHK Ltd.', and links to 'GitHub', 'Twitter', and 'Feedback'.

```
1 -- Vesting scheme as a PLC contract
2 import Control.Lens (view)
3 import Control.Monad (void, when)
4 import Data.Default (Default (def))
5 import Data.Map qualified as Map
6 import Data.Text qualified as T
7
8 import Ledger (Address, POSIXTime, POSIXTimeRange, PaymentPubKeyHash (unPaymentPubKeyHash), Validator)
9 import Ledger.Ada qualified as Ada
10 import Ledger.Constraints (TxConstraints, mustBeSignedBy, mustPayToTheScript, mustValidateIn)
11 import Ledger.Constraints qualified as Constraints
12 import Ledger.Contexts (ScriptContext (..), TxInfo (..))
13 import Ledger.Contexts qualified as Validation
14 import Ledger.Interval qualified as Interval
15 import Ledger.TimeSlot qualified as TimeSlot
16 import Ledger.Tx qualified as Tx
17 import Ledger.Typed.Scripts qualified as Scripts
18 import Ledger.Value (Value)
19 import Ledger.Value qualified as Value
20 import Playground.Contract
21 import Plutus.Contract
22 import Plutus.Contract.Test
23 import Plutus.Contract.Typed.Tx qualified as Typed
24 import PlutusTx qualified
25 import PlutusTx.Prelude hiding (Semigroup (..), fold)
```

Language: Haskell and Plutus