fyp	Cal. Days		Aug 2018	Sep 2018	Oct 2018	Nov 2018	Dec 2018	Jan 2019	Feb 2019	Mar 2019	Apr 2019	May 2019	Jun 2019
	Understand concept of blockchain	32d		Understand concept of blockchain									
1	Understand blockchain concepts	12d		Understand blockchain concepts									
2	Pick framework and language	7d		Pick framework and language									
3	Understand how to code in Solidity	6d		Understand how to code in Solidity									
4	Build application skeleton	5d		Build application skeleton									
	Decide style and design of application	26d		Decide style and design of application									
6	Experiment with different styles and colour schemes	4d		Experiment with different styles and colour schemes									
7	Prototype structure of application	19d		Prototype structure of application									
	Build frontend and visuals of application	79d				Build frontend and visuals of application							
9	Create Connect page	12d		Create Connect page									
10	Create Build page with tabs	15d		Create Build page with tabs									
	Create React components	40d			Create React components								
12	Create Diagram	40d			Create Diagram								
13	Create Diagram Modals	9d			Create Diagram Modals								
14	Create Param List	9d			Create Param List								
15	Create Requires List	9d			Create Requires List								
16	Create Variable List	5d			Create Variabl...								
	Write Interim Report						Write Interim Report						
	Create backend logic						Create backend logic						
	Build code parsing functionality						Build code parsing functionality						
	Build Deployment Functionality							Build Deployment Functionality					
	Write Final Report							Write Final Report					
	Amend Report							Amend Report					
	Prepare for presentation								Prepare for presentation				

Smart Contract Builder

Connect to blockchain

Protocol

HTTP ▾ : // Blockchain Address : Blockchain Port

CONNECT

Smart Contract Builder

GLOBAL STATE   INITIAL STATE   +

Events

Event Name   [ADD +]

Entities

Entity Name   [ADD +]

[BACK] [LOAD] [SAVE] [GENERATE CODE] [DEPLOY]

Smart Contract Builder

Events

deposit

Variable Name	Variable Type
from	Address ▾

Variable Name	Variable Type
id	Text ▾

Variable Name	Variable Type
value	Number ▾

+

Event Name   ADD +

Entities

Entity Name   ADD +

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Smart Contract Builder

Events

Event Name   ADD +

Entities

car

Variable Name
price

Variable Type
Number

Variable Name
brand

Variable Type
Text

Variable Name
seller

Variable Type
Address

+

Entity Name   ADD +

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Function Inputs

Variable Name
Initial Value

Variable Type
Number ▼

Variable Name
Seller Name

Variable Type
Text ▼

+

Smart Contract Builder

GLOBAL STATE	INITIAL STATE	+

Events

Event Name   ADD **+**

Entities

Entity Name   ADD **+**

Constructor Parameters

Value of Initial Value

Value of Seller Name

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Smart Contract Builder

GLOBAL STATE   INITIAL STATE   **PURCHASE**   +

Function Inputs

Variable Name   Variable Type

Number ▾

+

Checking Phase

Variable 1   Comparator   is ▾   Variable 2   Failure Message

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

	New Assignment Node Variable Name $^=$ Assigned Value CANCEL ⊗ DONE ✓

New Event Node

Event to emit

Deposit ▾

Event Parameters

Value of from
_____

Value of id
_____

Value of value
_____

CANCEL ⊗   DONE ✓

New Entity Node

Entity Name

chosen car

Entity Type

Car ▾

Event Parameters

Value of price

1000

Value of brand

"Mercedes"

Value of seller

message sender

CANCEL ⊗   DONE ✓

New Transfer Node

Transfer to

Value

CANCEL ⊗   DONE ✓

	New Return Node Return Variable CANCEL ⊗ DONE ✓
	New Conditional Node Variable 1 Comparator is ▼ Variable 2 CANCEL ⊗ DONE ✓

Action Phase

Nodes

- Assignment Node
- Event Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: result = 10

Assignment: counter = 0

counter < 5

False

True

Assignment: counter = counter + 1

Assignment: result = result * 2

Return: result
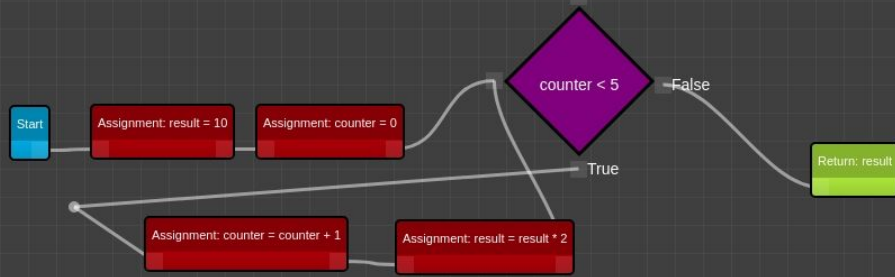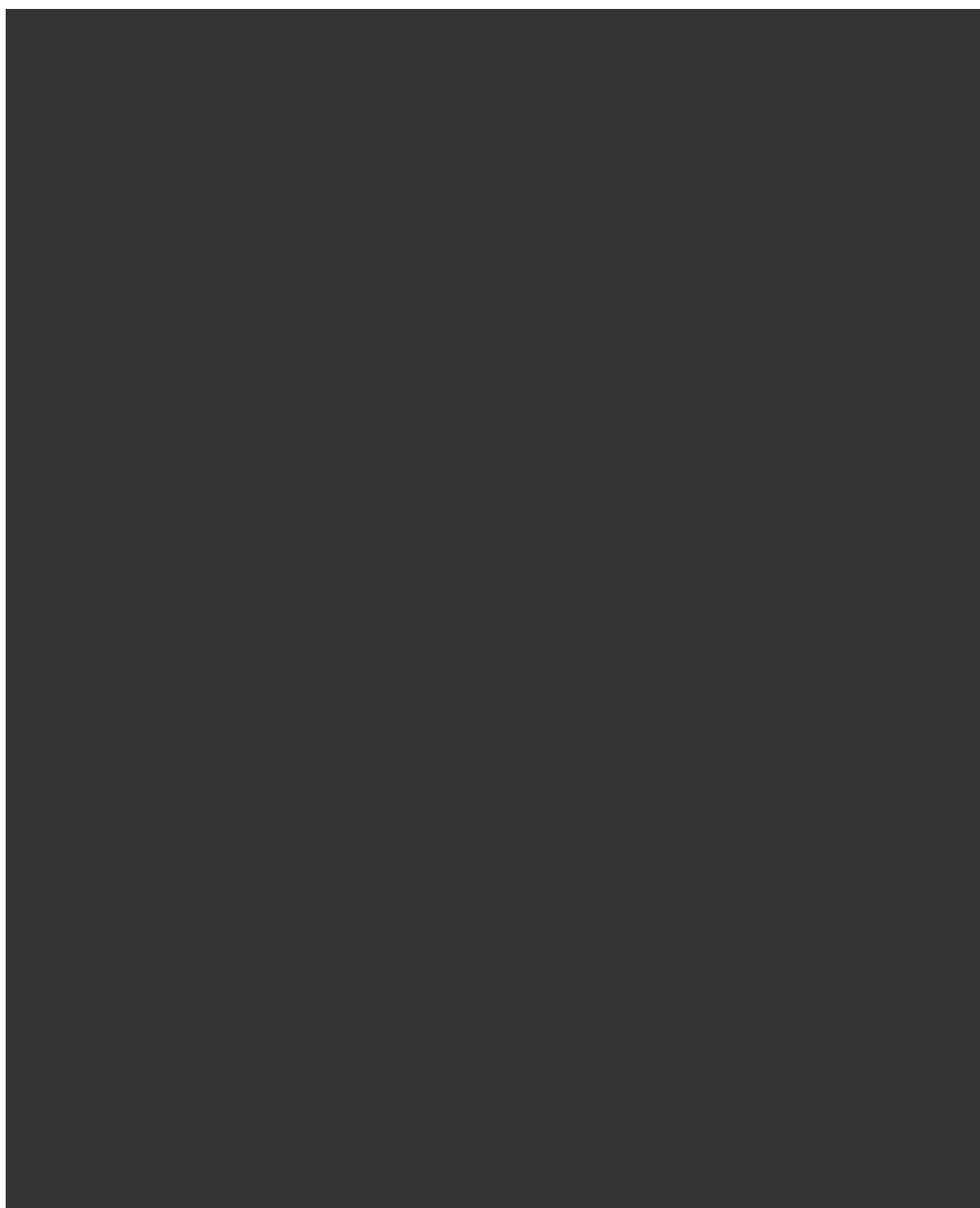
Smart Contract Builder

GLOBAL STATE | INITIAL STATE | ADD PROPOSAL | GIVE RIGHT TO VOTE | VOTE | WINNING PROPOSAL | >

Events

Event Name   ADD +

Entities

Voter

Variable Name	Variable Type
weight	Number

Variable Name	Variable Type
voted	True/False

Variable Name	Variable Type
delegate	Address

Variable Name	Variable Type
vote	Text

+

Entity Name   ADD +

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: chairperson = message sender

New Entity: chairperson voter = Voter(1, false, an address, "")

Assignment: voter details for chairperson = chairperson voter

Function Inputs

Variable Name
proposal

Variable Type
Text

+
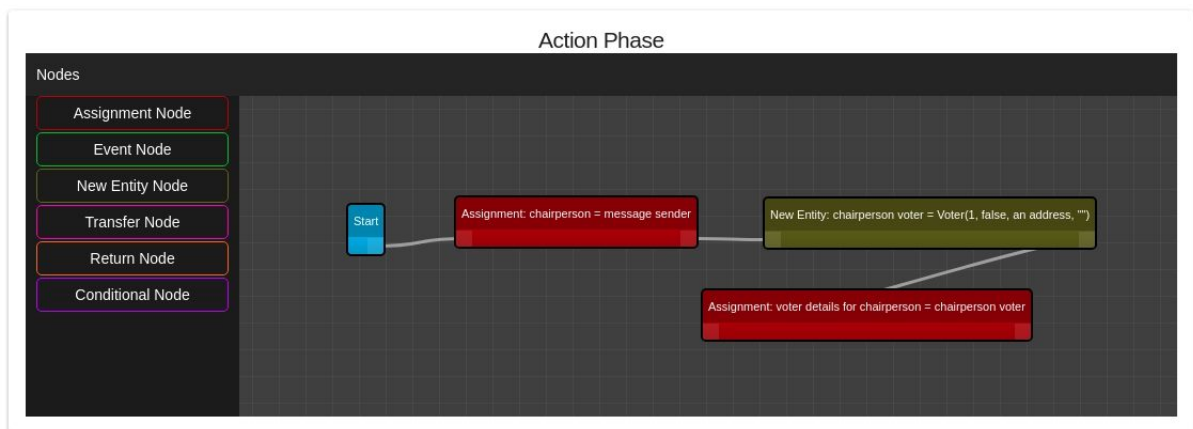
Checking Phase

Variable 1
message sender

Comparator
is

Variable 2
chairperson

Failure Message
Only chairperson can add a

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: vote count for proposal = 0
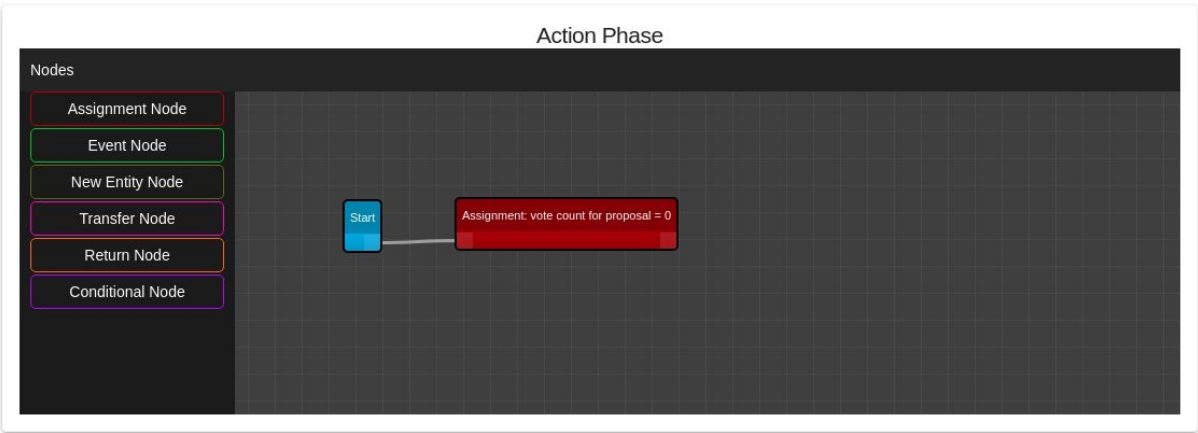
Function Inputs

Variable Name
target voter

Variable Type
Address ▼

+

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
message sender	is ▼	chairperson	Only chairperson can give ri
voter details for target voter':	is ▼	false	The voter already voted.
voter details for target voter':	is ▼	0	The voter already has a righ
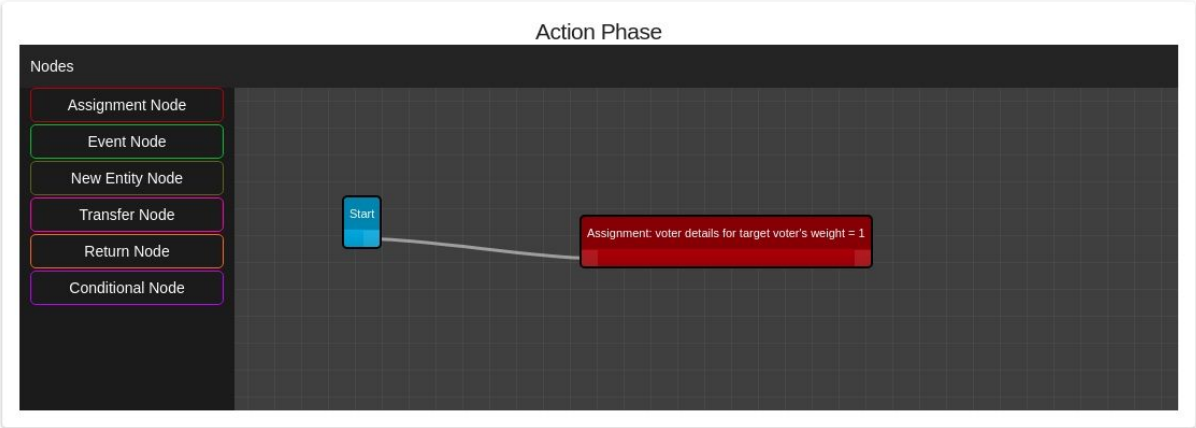
+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: voter details for target voter's weight = 1

Function Inputs

Variable Name
proposal

Variable Type
Text

\+

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
voter details of message ser	is not	0	Has no right to vote.
voter details of message ser	is	false	Already voted.

\+

Action Phase

Action Phase

Smart Contract Builder
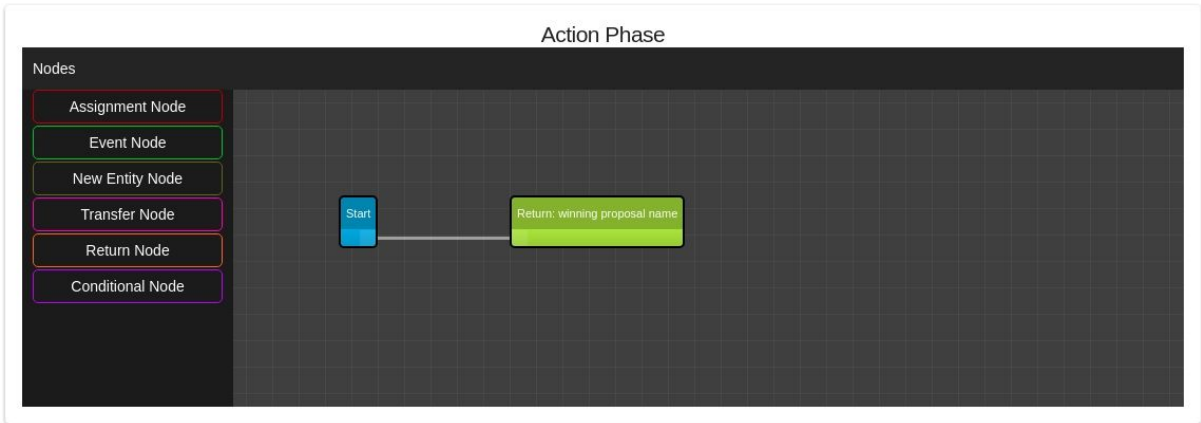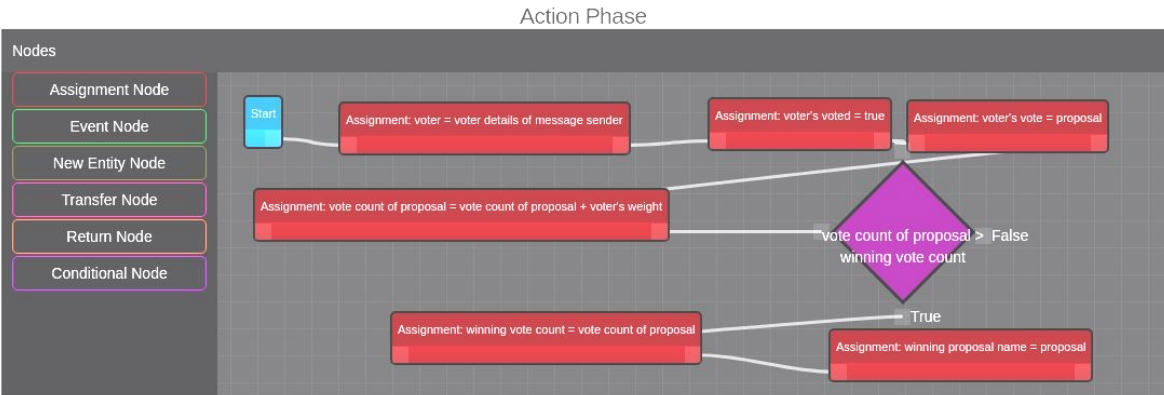
Events

HighestBidIncreased

Variable Name
bidder

Variable Type
Address

Variable Name
amount

Variable Type
Number

+

AuctionEnded

Variable Name
winner

Variable Type
Address

Variable Name
amount

Variable Type
Number

+

Event Name

ADD +

Entities

Entity Name

ADD +

Constructor Parameters

Value of _beneficiary
0x0000000000000000000000000000000000

Value of bidding time
1000

Function Inputs

Variable Name
_beneficiary

Variable Type
Address ▾

Variable Name
bidding time

Variable Type
Integer ▾

+

Checking Phase

Variable 1
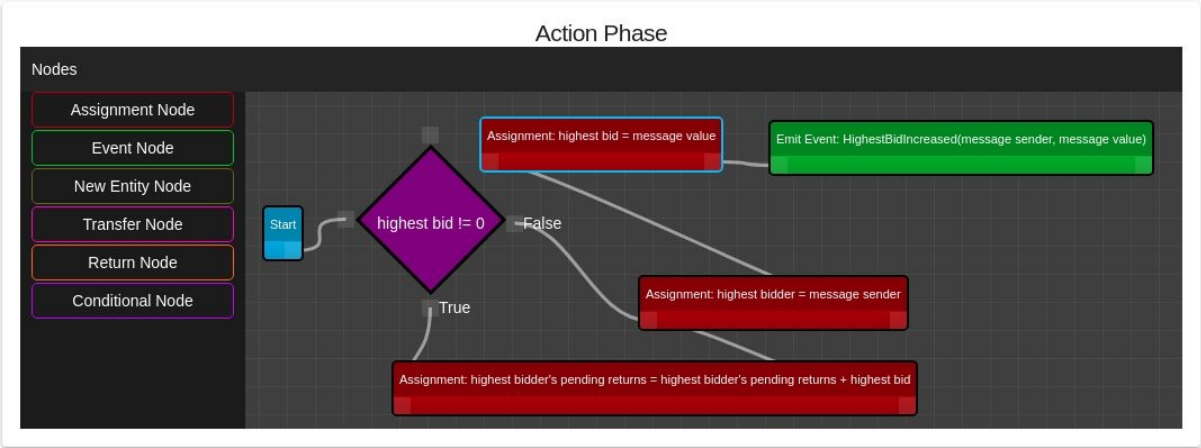
Comparator
is ▾

Variable 2

Failure Message

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: beneficiary = _beneficiary

Assignment: auction end = now + bidding time
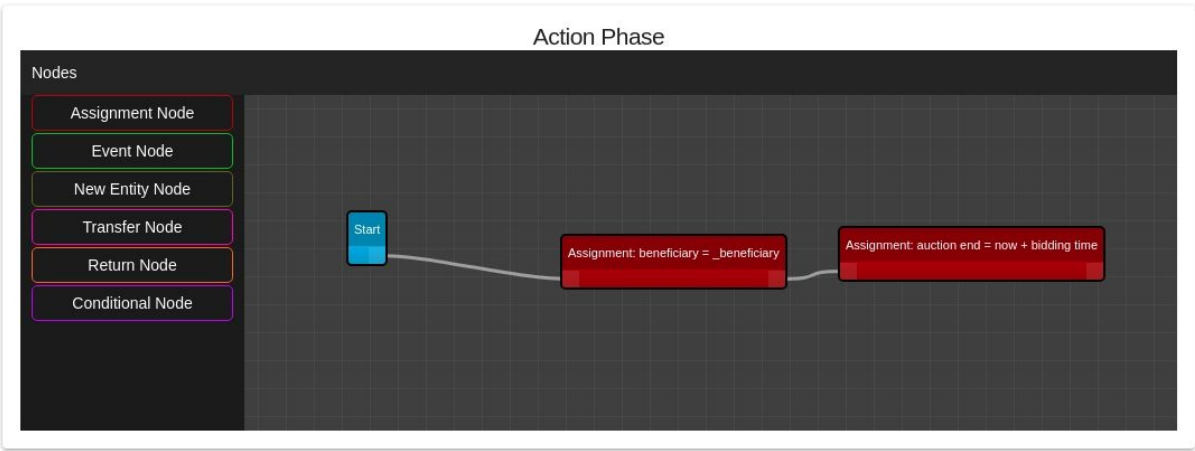
Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
now	less than or e… ▾	auction end	Auction already en
message value	greater than ▾	highest bid	There already is a

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

highest bid != 0

Assignment: highest bid = message value

Emit Event: HighestBidIncreased(message sender, message value)

False

True

Assignment: highest bidder = message sender

Assignment: highest bidder's pending returns = highest bidder's pending returns + highest bid

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: amount = pending returns for message sender

amount > 0

False

True

Assignment: pending returns for message sender = 0

Transfer: amount to message sender

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
now	greater than or equals to ▾	auction end	Auction not yet ended.
ended	is ▾	false	auctionEnd has already been
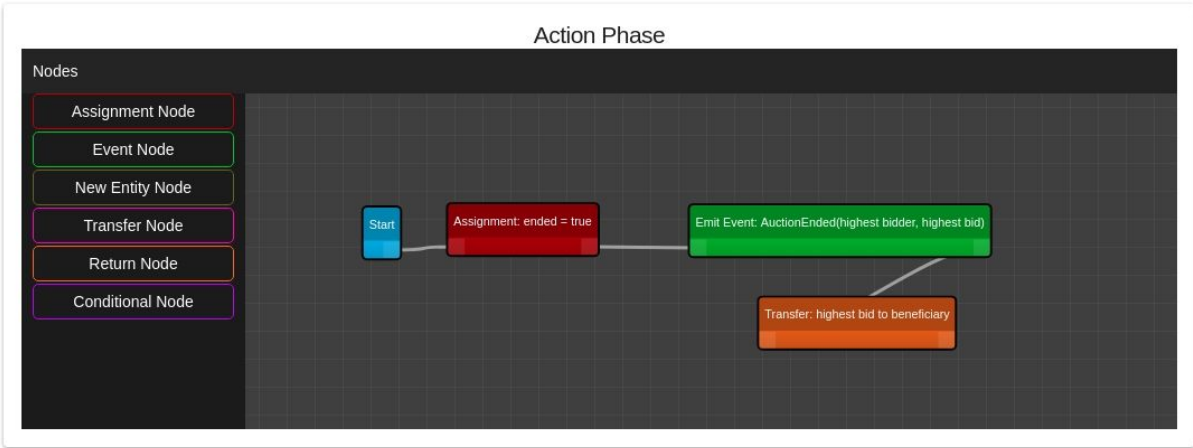
+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start → Assignment: ended = true → Emit Event: AuctionEnded(highest bidder, highest bid) → Transfer: highest bid to beneficiary

Smart Contract Builder

Events

aborted

Variable Name

Variable Type
Number

+

purchase_confirmed

Variable Name

Variable Type
Number

+

item_received

Variable Name

Variable Type
Number

+

Event Name   ADD +

Entities

Entity Name   ADD +

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
2 * message value / 2	is	message value	Value has to be even.

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start → Assignment: seller = message sender → Assignment: sell value = message value / 2
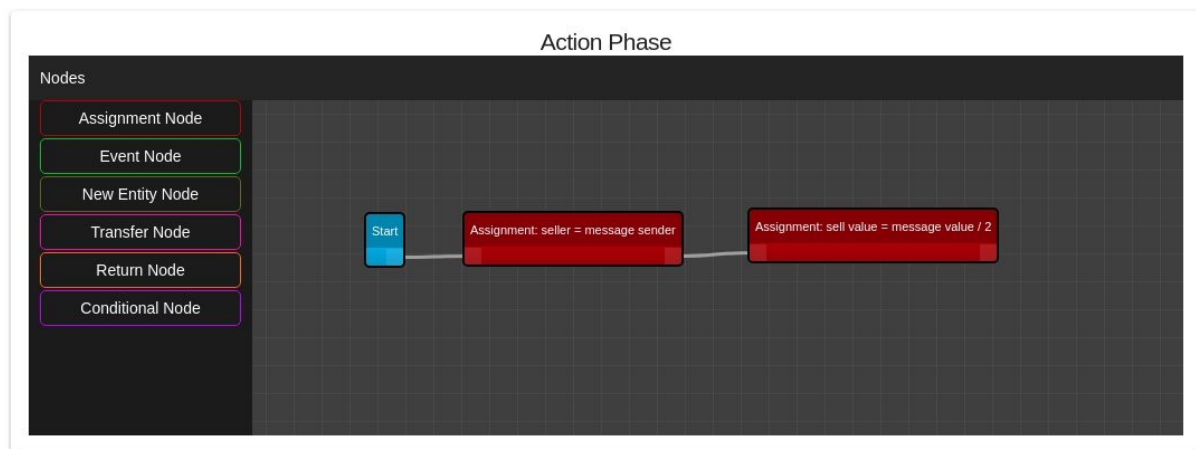
Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
message sender	is	seller	Only seller can call this.
state	is	0	Invalid state.

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start → Emit Event: aborted() → Assignment: state = 2 → Transfer: balance to seller

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
state	is	0	Invalid state.
message value	is	2 * sell value	Failure Message
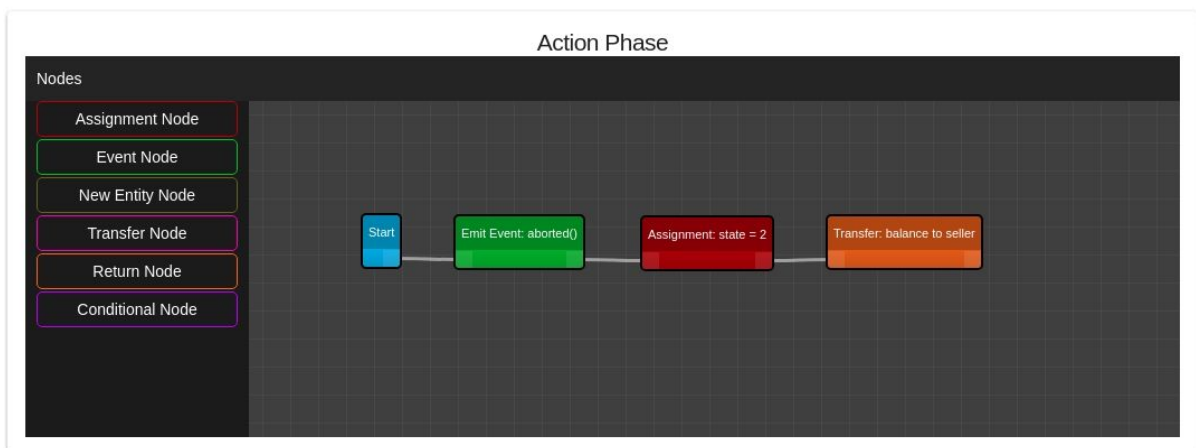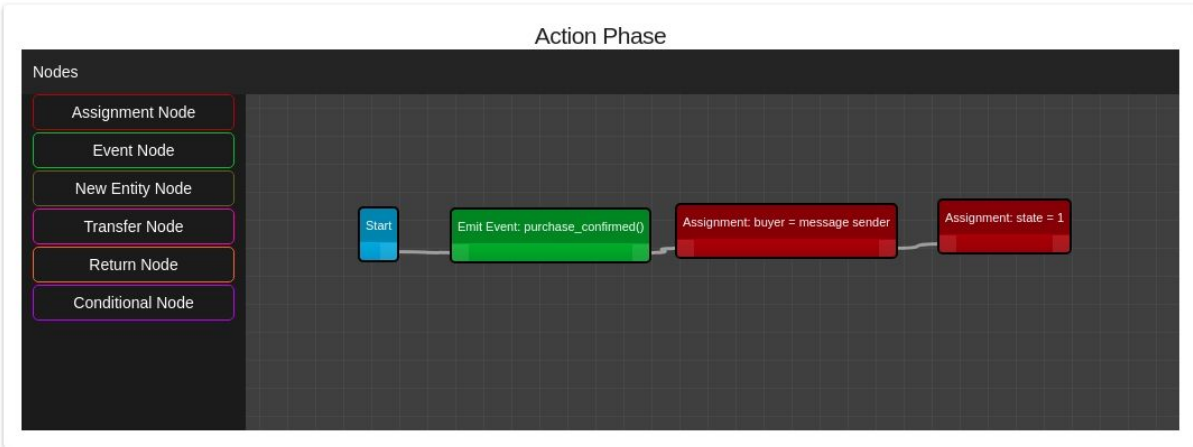
+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start → Emit Event: purchase_confirmed() → Assignment: buyer = message sender → Assignment: state = 1

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
message sender	is	buyer	Only buyer can call this.
state	is	1	Invalid state.

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start → Emit Event: item_received() → Assignment: state = 2 → Transfer: sell value to buyer → Transfer: balance to seller
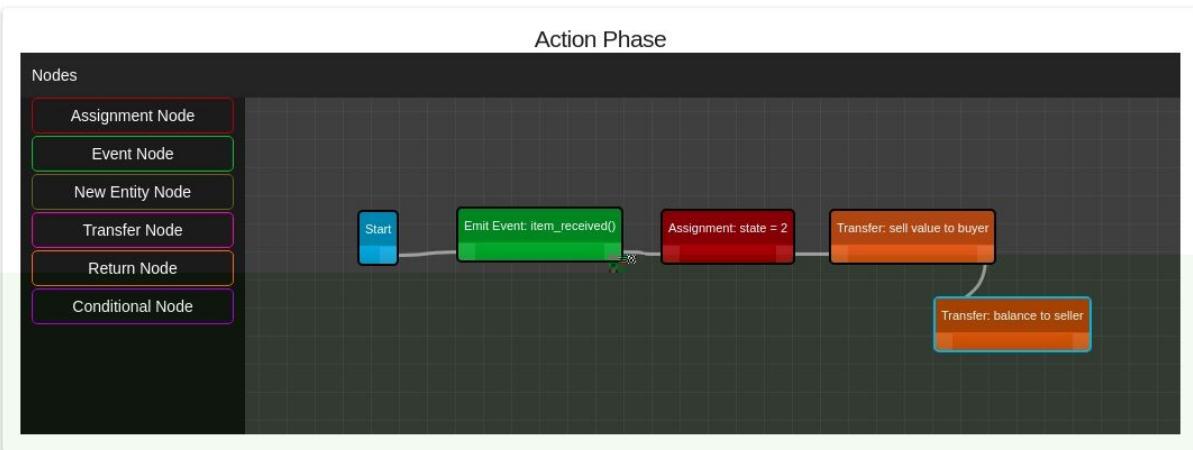
Smart Contract Builder

GLOBAL STATE INITIAL STATE TRANSFER APPROVE TRANSFER FROM INCREASE ALLOWANCE DECREASE ALLOWANCE +

Events

transferred

Variable Name	Variable Type
from	Address
to	Address
amount	Number

+

approval

Variable Name	Variable Type
owner	Address
spender	Address
amount	Number

+

Event Name   ADD +

Entities

Entity Name   ADD +

BACK   LOAD   SAVE   GENERATE CODE   DEPLOY

Function Inputs

Variable Name	Variable Type
to	Address
amount	Number

+

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
to	is not	an address	

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: _balances of message sender = _balances of message sender - amount

Assignment: _balances of to = _balances of to + amount

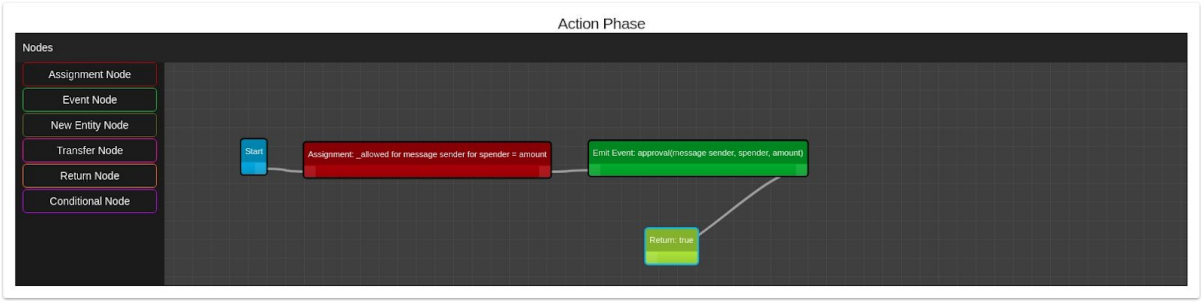Emit Event: transferred(message sender, to, amount)

Return: true

Function Inputs

Variable Name
spender

Variable Type
Address

Variable Name
amount

Variable Type
Number

+

Checking Phase

Variable 1	Comparator	Variable 2	
spender	is not	an address	Failure Message
message sender	is not	an address	Failure Message

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: _allowed for message sender for spender = amount

Emit Event: approval(message sender, spender, amount)

Return: true

Function Inputs

Variable Name	Variable Type
from	Address
to	Address
amount	Number

[+]

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
from	is not	an address	
to	is not	an address	

[+]

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: _balances for from = _balances for from - amount

Assignment: _balances for to = _balances for to + amount

Emit Event: transferred(from, to, amount)

Assignment: _allowed for from for message sender = _allowed for from for message sender - amount

Emit Event: approval(from, message sender, amount)

Return: true

Function Inputs

Variable Name	Variable Type
spender	Address
added value	Number

[+]

Checking Phase

Variable 1	Comparator	Variable 2	Failure Message
spender	is not	an address	
message sender	is not	an address	

[+]

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: _allowed for message sender for spender = _allowed for message sender for spender + added value

Emit Event: approval(message sender, spender, added value)

Return: true

Function Inputs

Variable Name: spender
Variable Type: Address

Variable Name: subtracted value
Variable Type: Number

+

Checking Phase

Variable 1: spender
Comparator: is not
Variable 2: an address
Failure Message:

Variable 1: message sender
Comparator: is not
Variable 2: an address
Failure Message:

+

Action Phase

Nodes

- Assignment Node
- Event Node
- New Entity Node
- Transfer Node
- Return Node
- Conditional Node

Start

Assignment: _allowed for message sender for spender = _allowed for message sender for spender - subtracted value

Emit Event: approval(message sender, spender, subtracted value)

Return: true