0.1 Environment

name	type	description
DevelopmentVaultUSD8GPercent	real	
tokensInCirculation	real	
TokenInPlatform	real	
PercentLiquidity	real	
CriticalFiatLiquidity	real	
burnSale	real	
stakePercentNft	real	
stakeNftRewardPercent	real	
nftTokenPercent	real	
gameSaleFactor	real	
totalPercentReward	real	
buyFactor	real	
holdFactor	real	
farmingSaleFactor	real	
stakeSaleFactor	real	
INCOME	(real) → real	
LiquidityShare	real	
FullLiquidityPoolToken	real	
percentStakingReward	real	
percentForReinvestment	real	
percentReinvestment	real	
percentShopProfit	real	
coef88day	real	
Reward8F8G	real	
USD8GReward	real	
used8G	real	
percentOutputDevelopmentGame	real	
GAME_TOKENS_8G	real	
GAME_TOKENS_8F	real	
percentStakeBurn	real	
StakingPoolToken	real	
LiquidityPoolFiat	real	
LiquidityPoolToken	real	
totalToken	real	
numerator	(int) → real	
DyExponent	(int) → real	
salesNumber	int	
month	real	
lastMonth	real	
tokenPrice	real	
salesEnd	(int) → int	

salesStart	(int) → int	
fiatIncome	(int) → real	
totalFiatIncome	real	
salesMaximalUSD	(int) → real	
salesMinimalUSD	(int) → real	
DyLinear	(int) → real	
salesFunction	(int) → int	
COEF	(int) → real	
Steepage	(int) → real	
interMonth	(int) → real	
CommissionVault	real	
percentBurnCommissionVault	real	
GameVault	real	
TreasureVault	real	
InvestmentVault	real	
LiquidityVault	real	
percentOutputCommissionTreasure	real	
percentOutputCommissionInvestment	real	
percentOutputCommissionLiquidity	real	
percentOutputCommissionGame	real	
percentFeeCommissionVault	real	
percentOutputTreasureDEX	real	
percentBurnTreasureDEX	real	
RewardVault	real	
percentOutputInvestmentDEX	real	
TeamVault	real	
DevelopmentVault	real	
percentOutputInvestmentTreasure	real	
percentOutputInvestmentTeam	real	
percentOutputInvestmentDevelopment	real	
percentOutputCommissionDevelopment	real	
seed_tokenPrice	real	
preseed_tokenPrice	real	
public_tokenPrice	real	
private_tokenPrice	real	
percentInputTreasurePublic	real	
percentInputDevelopmentPublic	real	
percentInputInvestmentPublic	real	
SlowClaim	real	
criticalPrice	real	
TreasureVaultInit	real	
percentOutputInvestmentGame	real	
RewardVaultShare	real	

percentPartners	real	
BURN	real	
CIRK	real	
traderHoldFactor	real	
traderFarmingFactor	real	
traderStakingFactor	real	
saleFactor	real	
traderSaleFactor	real	
InitialLiquidityPoolToken	real	
LPT_GLOBAL	real	
percentLvToLp	real	
LpDistrPercent	real	
bounty_tokenPrice	real	
RewardVaultLocked	real	
RewardVaultEnd	real	
RewardVaultStart	real	
RewardVaultdUnlock	real	
LiquidityFiatShare	real	
InitialLiquidityPoolFiat	real	
farmPoolRatio	real	
stakePoolRatio	real	
Proposals	real	
Demand	real	
UnlockTokens	real	
burnTokens	real	
RewadVaultInitialUnlock	real	

0.2 Agent types

- USER

name	type	description
token	real	
gameToken	real	

- STAKEHOLDER

name	type	description
token	real	
tokenLocked	real	
tokenShare	real	
startUnlocking	real	
endUnlocking	real	

initialUnlock	real	
dUnlock	real	
sellFactor	real	
fiat	real	
tokenStaked	real	
unstakeFactor	real	
priceFactor	real	
LPT	real	
tokenFarmed	real	
LTL	real	
LPT_pair	real	

0.3 Agent Instances

name	type	description
trader	STAKEHOLDER	
nft	STAKEHOLDER	
marketing	STAKEHOLDER	
stakeUser	STAKEHOLDER	
marketplaceUser	USER	
gameUser	USER	
advisors	STAKEHOLDER	
public	STAKEHOLDER	
preseed	STAKEHOLDER	
seed	STAKEHOLDER	
private	STAKEHOLDER	
team	STAKEHOLDER	
farmer	STAKEHOLDER	
development	STAKEHOLDER	

0.4 Initial Environment

$totalToken == 888888888 \land$
$totalFiatIncome == 0 \land$
$month == 1 \land$
$lastMonth == 60 \land$
$BURN == 0 \land$
$CIRK == totalToken \land$
$LPT\_GLOBAL == 0 \land$

$Demand == 0 \ \land \ Proposals == 0 \land$
$tokensInCirculation == 0 \ \land \ UnlockTokens == 0 \land$
$burnTokens == 0 \land$

$StakingPoolToken == 1 \wedge$
$percentStakeBurn ==\ 0.08 \wedge$

$LiquidityPoolFiat == 200000 \wedge$
$InitialLiquidityPoolToken == 2000000 \wedge$
$InitialLiquidityPoolFiat == 200000 \wedge$
$LiquidityPoolToken == 6666667 \wedge$
$PercentLiquidity ==\ 0.05 \wedge$
$CriticalFiatLiquidity == 5000000 \wedge$

$farmPoolRatio ==\ 0.5 \wedge$
$stakePoolRatio ==\ 0.5 \wedge$

$LiquidityShare ==\ 0.13 \wedge$
$LiquidityFiatShare ==\ 0.13 \wedge$

$percentLvToLp ==\ 0.05 \wedge$
$LpDistrPercent ==\ 0.5 \wedge$

$tokenPrice ==\ 0.03 \wedge$

INCOME (1) == 4000000 $\wedge$
INCOME (2) == 5000000 $\wedge$
INCOME (3) == 1000000 $\wedge$
INCOME (4) == 1100000 $\wedge$
INCOME (5) == 1210000 $\wedge$
INCOME (6) == 1331000 $\wedge$
INCOME (7) == 1464100 $\wedge$
INCOME (8) == 1610510 $\wedge$
INCOME (9) == 1771560 $\wedge$
INCOME (10) == 1948710 $\wedge$
INCOME (11) == 2143580 $\wedge$
INCOME (12) == 2357940 $\wedge$
INCOME (13) == 2593740 $\wedge$
INCOME (14) == 2853110 $\wedge$
INCOME (15) == 3138420 $\wedge$
INCOME (16) == 3452270 $\wedge$
INCOME (17) == 3797490 $\wedge$
INCOME (18) == 4177240 $\wedge$
INCOME (19) == 4594970 $\wedge$
INCOME (20) == 5054470 $\wedge$

INCOME (21) == 5559910 $\wedge$
INCOME (22) == 6115900 $\wedge$
INCOME (23) == 6727500 $\wedge$
INCOME (24) == 6727500 $\wedge$

$CommissionVault == 0 \wedge$
$percentFeeCommissionVault == \ 0.01 \wedge$
$percentBurnCommissionVault == 0 \wedge$
$percentOutputCommissionGame == \ 0.2 \wedge$
$percentOutputCommissionLiquidity == \ 0.1 \wedge$
$percentOutputCommissionInvestment == \ 0.2 \wedge$
$percentOutputCommissionTreasure == \ 0.3 \wedge$
$percentOutputCommissionDevelopment == \ 0.2 \wedge$
$percentPartners == \ 0.01 \wedge$

$TreasureVault == \ (public.fiat \ + \ seed.fiat \ + \ private.fiat \ + \ preseed.fiat) \times$
$percentInputTreasurePublic \wedge$
$percentOutputTreasureDEX == \ 0.01 \wedge$
$percentBurnTreasureDEX == \ 0.5 \wedge$
$percentInputTreasurePublic == 0 \wedge$
$criticalPrice == \ 0.04 \wedge$

$GameVault == 0 \wedge$
$GAME\_TOKENS\_8G == 0 \wedge$
$GAME\_TOKENS\_8F == 0 \wedge$
$used8G == \ 0.5 \wedge$
$USD8GReward == \ 0.5 \wedge$
$Reward8F8G == \ 0.5 \wedge$
$coef88day == \ 0.01 \wedge$
$percentShopProfit == 0.2 \wedge$

$InvestmentVault == 0 \wedge$
$percentOutputInvestmentDEX == 0 \wedge$
$percentOutputInvestmentDevelopment == \ 0.1 \wedge$
$percentOutputInvestmentTeam == \ 0.15 \wedge$
$percentOutputInvestmentTreasure == \ 0.1 \wedge$
$percentInputInvestmentPublic == \ 0.3 \wedge$
$percentOutputInvestmentGame == \ 0.15 \wedge$
$percentForReinvestment == \ 0.5 \wedge$
$percentReinvestment == \ 0.0333 \wedge$

$LiquidityVault == 0 \wedge$

$RewardVaultLocked == \ RewardVaultShare \times totalToken \ \times \ (1 - RewadVaultInitialUnlock) \wedge$

$RewardVault == \ RewardVaultShare \times totalToken \times RewadVaultInitialUnlock \wedge$
$RewardVaultStart == 0 \wedge$
$RewardVaultEnd == 42 \wedge$
$RewadVaultInitialUnlock == \ 0.01 \wedge$
$RewardVaultdUnlock == \ (RewardVaultShare \times totalToken \times (\ 1 - RewadVaultInitialUnlock)) / (RewardVaultEnd - \ RewardVaultStart) \ \wedge$
$RewardVaultShare == \ 0.27 \wedge$
$percentStakingReward == \ 0.01875 \wedge$
$totalPercentReward == \ 0.138 \wedge$

$SlowClaim == \ 0.082191 \wedge$

$DevelopmentVault == \ (public.fiat \ + \ seed.fiat \ + \ private.fiat \ + \ preseed.fiat) \times percentInputDevelopmentPublic \wedge$
$percentInputDevelopmentPublic == \ 0.91 \wedge$
$percentOutputDevelopmentGame == \ 0.05 \wedge$
$DevelopmentVaultUSD8GPercent == \ 0.3 \wedge$

$salesNumber == 10 \wedge$

salesEnd (1) == 30 ∧
salesStart (1) == 1 ∧
salesMaximalUSD (1) == 24000 ∧
salesMinimalUSD (1) == 8000 ∧
salesFunction (1) == 2 ∧
COEF (1) == 0 . 9 ∧
Steepage (1) == 1 ∧
interMonth (1) == 1 ∧
fiatIncome (1) == salesMinimalUSD (1) ∧
numerator (1) == 0 ∧

salesEnd (2) == lastMonth ∧
salesStart (2) == 31 ∧
salesMaximalUSD (2) == 20000 ∧
salesMinimalUSD (2) == 24000 ∧
salesFunction (2) == 1 ∧
DyLinear (2) == (salesMaximalUSD (2) − salesMinimalUSD (2)) / (salesEnd (2) − salesStart (2)) ∧
fiatIncome (2) == salesMinimalUSD (2) ∧

salesEnd (3) == 12 ∧
salesStart (3) == 1 ∧
salesMaximalUSD (3) == 4500000 ∧
salesMinimalUSD (3) == 1000000 ∧
salesFunction (3) == 2 ∧
COEF (3) == 0 . 9 ∧

Steepage (3) == 1 ∧
interMonth (3) == 1 ∧
fiatIncome (3) == salesMinimalUSD (3) ∧
numerator (3) == 0 ∧

salesEnd (4) == lastMonth ∧
salesStart (4) == 13 ∧
salesMaximalUSD (4) == 4000000 ∧
salesMinimalUSD (4) == 4500000 ∧
salesFunction (4) == 1 ∧
DyLinear (4) == (salesMaximalUSD (4) − salesMinimalUSD (4)) / (salesEnd (4) − salesStart (4)) ∧
fiatIncome (4) == salesMinimalUSD (4) ∧

salesEnd (5) == lastMonth ∧
salesStart (5) == 1 ∧
salesMaximalUSD (5) == 2000 ∧
salesMinimalUSD (5) == 50 ∧
salesFunction (5) == 1 ∧
DyLinear (5) == (salesMaximalUSD (5) − salesMinimalUSD (5)) / (salesEnd (5) − salesStart (5)) ∧
fiatIncome (5) == salesMinimalUSD (5) ∧

salesEnd (6) == lastMonth ∧
salesStart (6) == 1 ∧
salesMaximalUSD (6) == 50000 ∧
salesMinimalUSD (6) == 5000 ∧
salesFunction (6) == 1 ∧
DyLinear (6) == (salesMaximalUSD (6) − salesMinimalUSD (6)) / (salesEnd (6) − salesStart (6)) ∧
fiatIncome (6) == salesMinimalUSD (6) ∧

salesEnd (7) == 12 ∧
salesStart (7) == 1 ∧
salesMaximalUSD (7) == 240000 ∧
salesMinimalUSD (7) == 80000 ∧
salesFunction (7) == 2 ∧
COEF (7) == 0 . 9 ∧
Steepage (7) == 1 ∧
interMonth (7) == 1 ∧
fiatIncome (7) == salesMinimalUSD (7) ∧
numerator (7) == 0 ∧

salesEnd (8) == lastMonth ∧

salesStart (8) == 13 ∧
salesMaximalUSD (8) == 200000 ∧
salesMinimalUSD (8) == 240000 ∧
salesFunction (8) == 1 ∧
DyLinear (8) == (salesMaximalUSD (8) − salesMinimalUSD (8)) / (salesEnd (8) − salesStart (8)) ∧
fiatIncome (8) == salesMinimalUSD (8) ∧

salesEnd (9) == 12 ∧
salesStart (9) == 1 ∧
salesMaximalUSD (9) == 15000 ∧
salesMinimalUSD (9) == 2500 ∧
salesFunction (9) == 2 ∧
COEF (9) == 0 . 9 ∧
Steepage (9) == 1 ∧
interMonth (9) == 1 ∧
fiatIncome (9) == salesMinimalUSD (9) ∧
numerator (9) == 0 ∧

salesEnd (10) == lastMonth ∧
salesStart (10) == 13 ∧
salesMaximalUSD (10) == 10000 ∧
salesMinimalUSD (10) == 15000 ∧
salesFunction (10) == 1 ∧
DyLinear (10) == (salesMaximalUSD (10) − salesMinimalUSD (10)) / (salesEnd (10) − salesStart (10)) ∧
fiatIncome (10) == salesMinimalUSD (10) ∧

team . startUnlocking == 10 ∧
team . endUnlocking == 46 ∧
team . tokenShare == 0 . 15 ∧
team . tokenLocked == team . tokenShare × totalToken ∧
team . token == 0 ∧
team . initialUnlock == 0 ∧
team . dUnlock == (team . tokenShare × totalToken × ($1 - team.initialUnlock$)) / (team . endUnlocking − team . startUnlocking) ∧
team . sellFactor == 0 . 92 ∧
team . LPT == 0 ∧
team . LPT_pair == 0 ∧
team . LTL == 0 ∧
team . tokenStaked == 0 ∧
team . tokenFarmed == 0 ∧
team . unstakeFactor == 0 . 25 ∧
team . priceFactor == tokenPrice ∧

```
seed . startUnlocking  == 3 ∧
seed . endUnlocking  == 18 ∧
seed . tokenShare  ==  0 . 08  ∧
seed . tokenLocked  ==  seed . tokenShare  × totalToken ∧
seed . token  == 0 ∧
seed . initialUnlock  ==  0 . 05  ∧
seed . dUnlock  == ( seed . tokenShare  × totalToken × ( 1 − seed.initialUnlock))
/ ( seed . endUnlocking  −  seed . startUnlocking ) ∧
seed . sellFactor  ==  0 . 92  ∧
seed . fiat  ==  seed . tokenShare  × totalToken × seed_tokenPrice ∧
seed_tokenPrice ==  0. 015∧
seed . unstakeFactor  ==  0 . 25  ∧
seed . tokenStaked  == 0 ∧
seed . LPT  == 0 ∧
seed . LPT_pair  == 0 ∧
seed . LTL  == 0 ∧
seed . priceFactor  == tokenPrice ∧
seed . tokenFarmed  == 0 ∧


preseed . startUnlocking  == 4 ∧
preseed . endUnlocking  == 24 ∧
preseed . tokenShare  ==  0 . 06  ∧
preseed . tokenLocked  ==  preseed . tokenShare  × totalToken ∧
preseed . token  == 0 ∧
preseed . initialUnlock  ==  0 . 02  ∧
preseed . dUnlock  == ( preseed . tokenShare  × totalToken × ( 1 − preseed.initialUnlock))
/ ( preseed . endUnlocking  −  preseed . startUnlocking ) ∧
preseed . sellFactor  ==  0 . 92  ∧
preseed . fiat  ==  preseed . tokenShare  × totalToken × preseed_tokenPrice
∧
preseed . LPT  == 0 ∧
preseed . LPT_pair  == 0 ∧
preseed . LTL  == 0 ∧
preseed . tokenStaked  == 0 ∧
preseed_tokenPrice ==  0. 01∧
preseed . tokenFarmed  == 0 ∧
preseed . unstakeFactor  ==  0 . 25  ∧
preseed . priceFactor  == tokenPrice ∧

development . startUnlocking  == 20 ∧
development . endUnlocking  == 36 ∧
development . tokenShare  ==  0 . 06  ∧
development . tokenLocked  ==  development . tokenShare  × totalToken ∧
development . initialUnlock  == 0 ∧
development . dUnlock  == ( development . tokenShare  × totalToken × (
```

$1 - development.initialUnlock$)) / (development . endUnlocking − development
. startUnlocking) ∧
development . token == 0 ∧
development . sellFactor == 0 . 92 ∧
development . LPT == 0 ∧
development . LPT_pair == 0 ∧

private . startUnlocking == 2 ∧
private . endUnlocking == 12 ∧
private . tokenShare == 0 . 03 ∧
private . tokenLocked == private . tokenShare × totalToken − private .
tokenShare × totalToken × nftTokenPercent ∧
private . token == 0 ∧
private . initialUnlock == 0 . 1 ∧
private . dUnlock == (private . tokenShare × totalToken × ($1 - private.initialUnlock$))
/ (private . endUnlocking − private . startUnlocking) ∧
private . sellFactor == 0 . 92 ∧
private . fiat == private . tokenShare × totalToken × private_tokenPrice ∧
private . LPT == 0 ∧
private . LPT_pair == 0 ∧
private . LTL == 0 ∧
private . tokenStaked == 0 ∧
$private\_tokenPrice == 0.020 \wedge$
private . tokenFarmed == 0 ∧
private . unstakeFactor == 0 . 25 ∧
private . priceFactor == tokenPrice ∧

nft . startUnlocking == 0 ∧
nft . endUnlocking == 24 ∧
nft . tokenShare == 0 . 06 ∧
nft . tokenLocked == private . tokenShare × totalToken × nftTokenPercent
+ private . tokenShare × totalToken × nftTokenPercent × stakePercentNft ×
stakeNftRewardPercent ∧
nft . token == 0 ∧
nft . initialUnlock == 0 ∧
nft . dUnlock == (nft . tokenShare × nft . tokenLocked × ($1 - nft.initialUnlock$))
/ (nft . endUnlocking − nft . startUnlocking) ∧
nft . sellFactor == 0 . 52 ∧
nft . fiat == 0 ∧
nft . LPT == 0 ∧
nft . LPT_pair == 0 ∧
nft . tokenStaked == 0 ∧
nft . tokenFarmed == 0 ∧
nft . unstakeFactor == 0 . 25 ∧
nft . priceFactor == tokenPrice ∧

```
public . startUnlocking  == 0 ∧
public . endUnlocking  == 6 ∧
public . tokenStaked  == 0 ∧
public . tokenShare  ==  0 . 01  ∧
public . tokenLocked  ==  public . tokenShare  × totalToken ∧
public . token  == 0 ∧
public . initialUnlock  ==  0 . 15  ∧
public . dUnlock == ( public . tokenShare  × totalToken × ( 1 − public.initialUnlock))
/ ( public . endUnlocking  −  public . startUnlocking ) ∧
public . sellFactor  ==  0 . 92  ∧
public . fiat  ==  public . tokenShare  × totalToken × public_tokenPrice ∧
public_tokenPrice ==  0.028∧
public . unstakeFactor  ==  0 . 25  ∧
public . priceFactor  == tokenPrice ∧
public . LPT  == 0 ∧
public . LPT_pair  == 0 ∧
public . LTL  == 0 ∧
public . tokenFarmed  == 0 ∧

advisors . startUnlocking  == 10 ∧
advisors . endUnlocking  == 46 ∧
advisors . tokenShare  ==  0 . 03  ∧
advisors . tokenLocked  ==  advisors . tokenShare  × totalToken ∧
advisors . token  == 0 ∧
advisors . initialUnlock  == 0 ∧
advisors . dUnlock == ( advisors . tokenShare  × totalToken × ( 1 − advisors.initialUnlock))
/ ( advisors . endUnlocking  −  advisors . startUnlocking ) ∧
advisors . sellFactor  ==  0 . 92  ∧
advisors . LPT  == 0 ∧
advisors . LPT_pair  == 0 ∧
advisors . LTL  == 0 ∧
advisors . tokenStaked  == 0 ∧
advisors . tokenFarmed  == 0 ∧
advisors . unstakeFactor  ==  0 . 25  ∧
advisors . priceFactor  == tokenPrice ∧


stakeUser . tokenStaked  == 0 ∧
stakeUser . unstakeFactor  ==  0 . 01  ∧
stakeUser . priceFactor  == tokenPrice ∧

farmer . token  == 0 ∧
farmer . tokenStaked  == 0 ∧
farmer . unstakeFactor  ==  0 . 005  ∧
```

farmer . priceFactor == tokenPrice $\wedge$
farmer . LPT == 0 $\wedge$
farmer . LTL == 0 $\wedge$
farmer . LPT_pair == 0 $\wedge$
farmer . tokenFarmed == 0 $\wedge$
trader . unstakeFactor == 0 . 25 $\wedge$

gameUser . token == 0 $\wedge$
marketplaceUser . token == 0 $\wedge$
gameUser . gameToken == 0 $\wedge$
marketplaceUser . gameToken == 0 $\wedge$

marketing . startUnlocking == 0 $\wedge$
marketing . endUnlocking == 40 $\wedge$
marketing . tokenLocked == marketing . tokenShare $\times$ totalToken $\wedge$
marketing . initialUnlock == 0 . 5 $\wedge$
marketing . dUnlock == (marketing . tokenShare $\times$ totalToken $\times$ ($1 - marketing.initialUnlock$))
/ (marketing . endUnlocking $-$ marketing . startUnlocking) $\wedge$
marketing . token == 0 $\wedge$
marketing . tokenShare == 0 . 15 $\wedge$
marketing . sellFactor == 0 . 92 $\wedge$
marketing . LPT == 0 $\wedge$
marketing . LPT_pair == 0 $\wedge$

marketing . LTL == 0 $\wedge$
marketing . tokenStaked == 0 $\wedge$
marketing . tokenFarmed == 0 $\wedge$
marketing . unstakeFactor == 0 . 25 $\wedge$
marketing . priceFactor == tokenPrice $\wedge$

trader . token == 0 $\wedge$
trader . sellFactor == 0 . 92 $\wedge$
trader . LPT == 0 $\wedge$
trader . LPT_pair == 0 $\wedge$
trader . tokenStaked == 0 $\wedge$
trader . tokenFarmed == 0 $\wedge$
trader . unstakeFactor == 0 . 25 $\wedge$
trader . priceFactor == tokenPrice $\wedge$

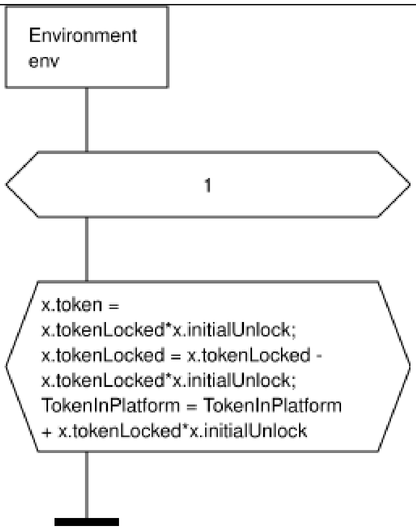
$traderHoldFactor == 0.76 \wedge$
$traderSaleFactor == 0.92 \wedge$

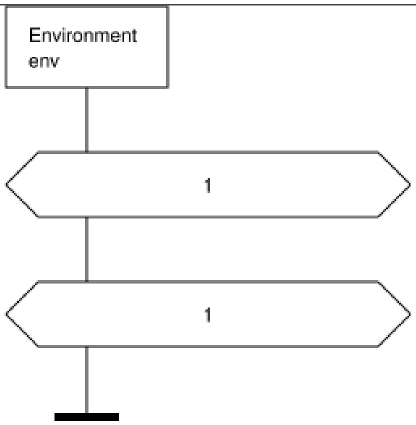
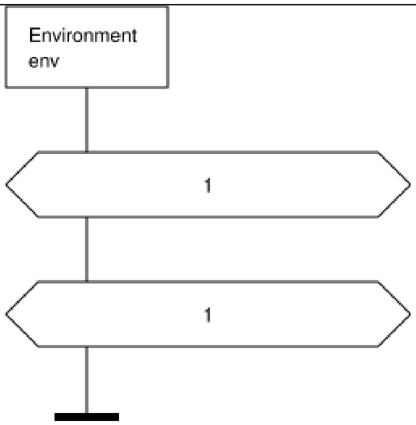
$farmingSaleFactor == 0.22 \wedge$

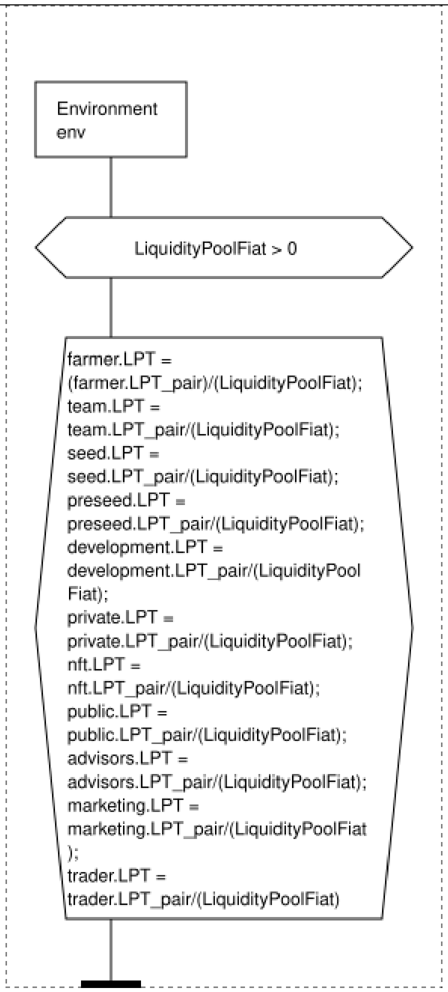
$holdFactor == 0.06\wedge$
$stakeSaleFactor == 0.15\wedge$
$saleFactor == 0.50 \times \ burnSale\wedge$
$burnSale == 0.99\wedge$
$gameSaleFactor == 0.12\wedge$

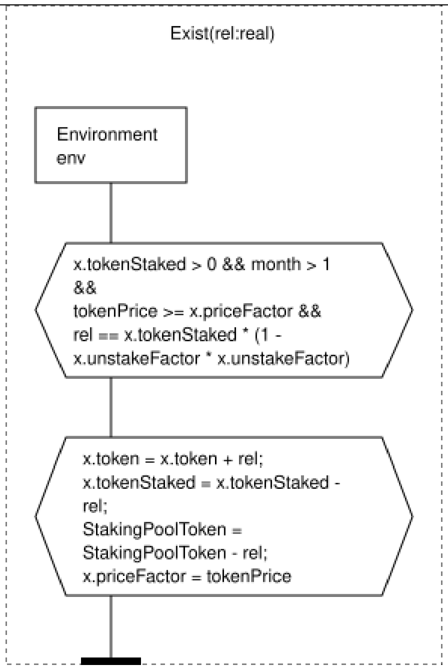
$nftTokenPercent == 0.01\wedge$
$stakeNftRewardPercent == 0.03\wedge$
$stakePercentNft == 0.01$

0.5 Actions

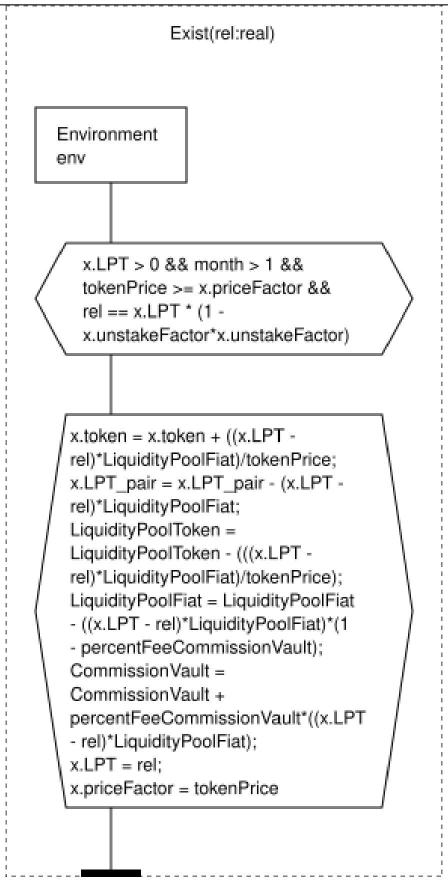
initUnlock (x) :	
	
	x.token =
	x.tokenLocked*x.initialUnlock;
	x.tokenLocked = x.tokenLocked -
	x.tokenLocked*x.initialUnlock;
	TokenInPlatform = TokenInPlatform
	+ x.tokenLocked*x.initialUnlock

print10 :	
print9 :	

$recalcLPT$:	 farmer.LPT = (farmer.LPT_pair)/(LiquidityPoolFiat); team.LPT = team.LPT_pair/(LiquidityPoolFiat); seed.LPT = seed.LPT_pair/(LiquidityPoolFiat); preseed.LPT = preseed.LPT_pair/(LiquidityPoolFiat); development.LPT = development.LPT_pair/(LiquidityPool Fiat); private.LPT = private.LPT_pair/(LiquidityPoolFiat); nft.LPT = nft.LPT_pair/(LiquidityPoolFiat); public.LPT = public.LPT_pair/(LiquidityPoolFiat); advisors.LPT = advisors.LPT_pair/(LiquidityPoolFiat); marketing.LPT = marketing.LPT_pair/(LiquidityPoolFiat); trader.LPT = trader.LPT_pair/(LiquidityPoolFiat)

withdrawStakeNatural (x) :	 Exist(rel:real)
	Environment env
	x.tokenStaked > 0 && month > 1 && tokenPrice >= x.priceFactor && rel == x.tokenStaked * (1 - x.unstakeFactor * x.unstakeFactor)
	x.token = x.token + rel;
	x.tokenStaked = x.tokenStaked - rel;
	StakingPoolToken = StakingPoolToken - rel;
	x.priceFactor = tokenPrice

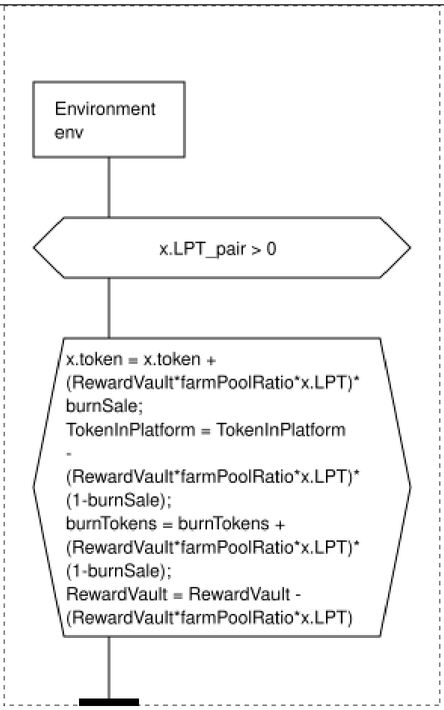
$withdrawFarmingNatural$ (x) :



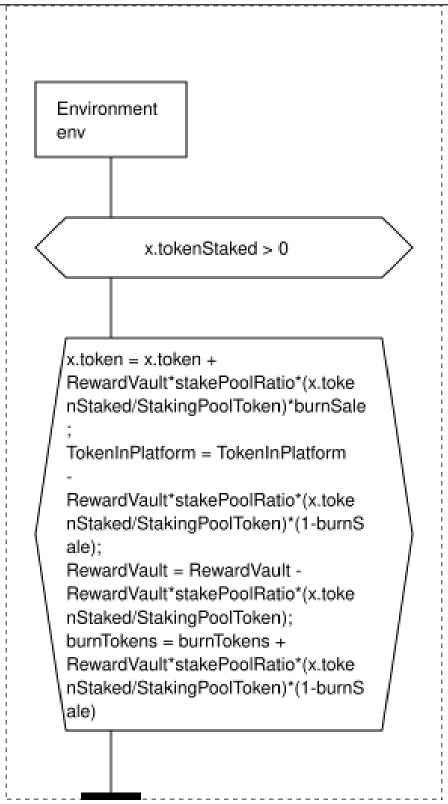
```
x.LPT > 0 && month > 1 &&
tokenPrice >= x.priceFactor &&
rel == x.LPT * (1 -
x.unstakeFactor*x.unstakeFactor)
```

```
x.token = x.token + ((x.LPT -
rel)*LiquidityPoolFiat)/tokenPrice;
x.LPT_pair = x.LPT_pair - (x.LPT -
rel)*LiquidityPoolFiat;
LiquidityPoolToken =
LiquidityPoolToken - (((x.LPT -
rel)*LiquidityPoolFiat)/tokenPrice);
LiquidityPoolFiat = LiquidityPoolFiat
- ((x.LPT - rel)*LiquidityPoolFiat)*(1
- percentFeeCommissionVault);
CommissionVault =
CommissionVault +
percentFeeCommissionVault*((x.LPT
- rel)*LiquidityPoolFiat);
x.LPT = rel;
x.priceFactor = tokenPrice
```

$rewardFarm$ (x) :

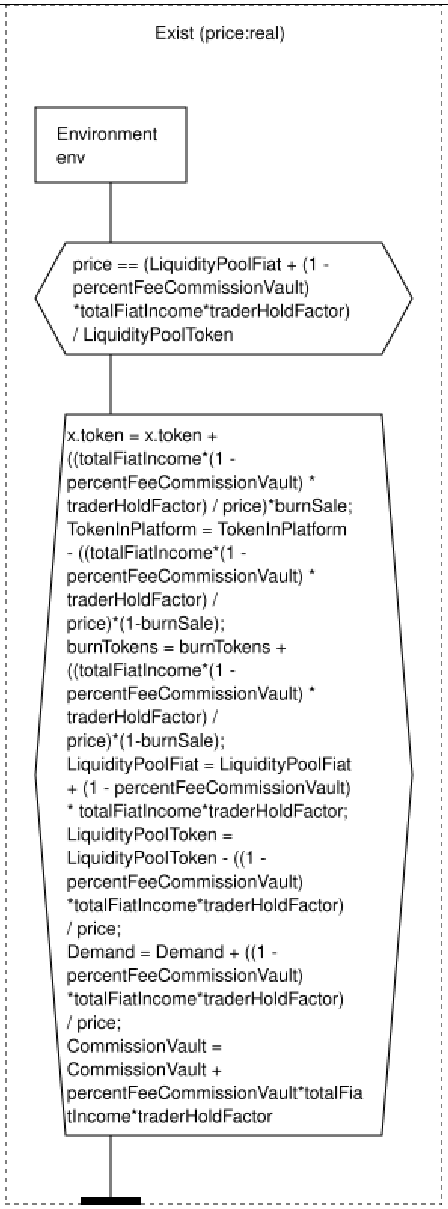


```
x.token = x.token +
(RewardVault*farmPoolRatio*x.LPT)*
burnSale;
TokenInPlatform = TokenInPlatform
-
(RewardVault*farmPoolRatio*x.LPT)*
(1-burnSale);
burnTokens = burnTokens +
(RewardVault*farmPoolRatio*x.LPT)*
(1-burnSale);
RewardVault = RewardVault -
(RewardVault*farmPoolRatio*x.LPT)
```

rewardStake (x) :	

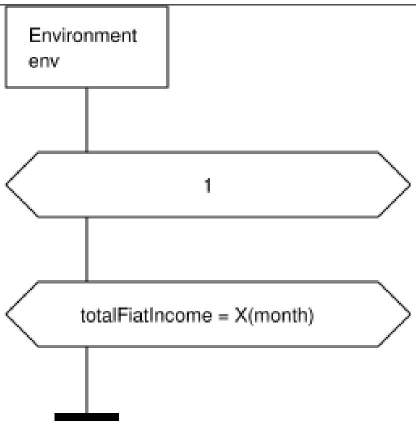
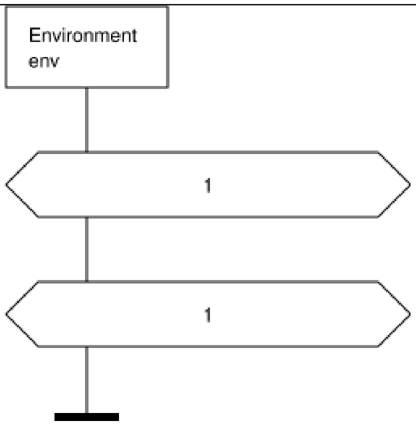
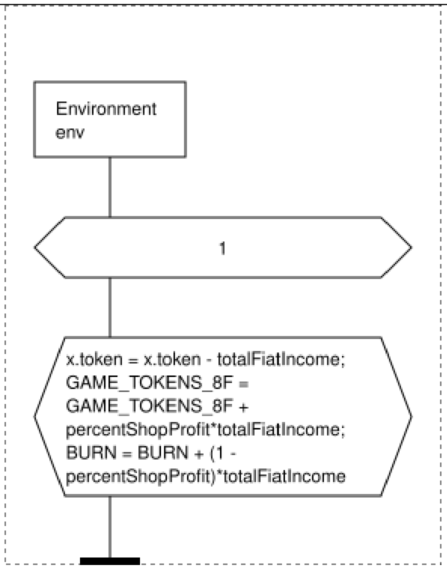
Environment env
x.tokenStaked > 0
x.token = x.token + RewardVault*stakePoolRatio*(x.tokenStaked/StakingPoolToken)*burnSale;
TokenInPlatform = TokenInPlatform - RewardVault*stakePoolRatio*(x.tokenStaked/StakingPoolToken)*(1-burnSale);
RewardVault = RewardVault - RewardVault*stakePoolRatio*(x.tokenStaked/StakingPoolToken);
burnTokens = burnTokens + RewardVault*stakePoolRatio*(x.tokenStaked/StakingPoolToken)*(1-burnSale)

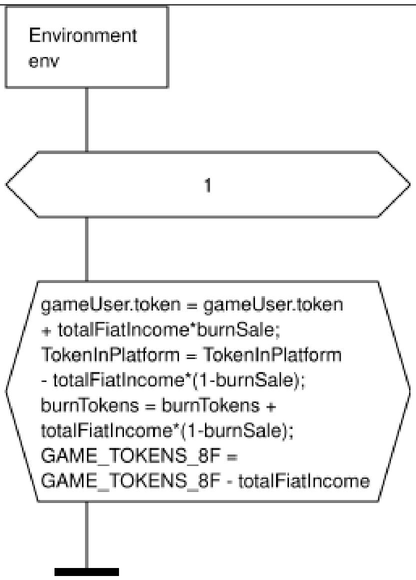
$buyTokenWallet$ (x) :

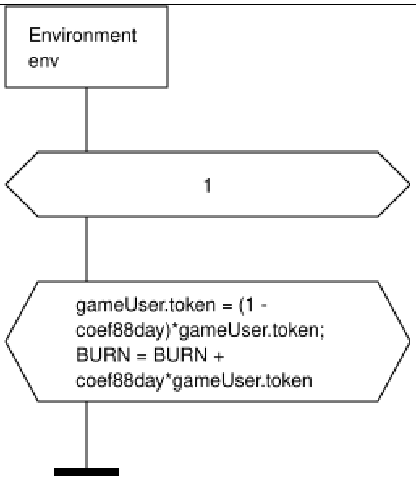


```
price == (LiquidityPoolFiat + (1 -
percentFeeCommissionVault)
*totalFiatIncome*traderHoldFactor)
/ LiquidityPoolToken
```

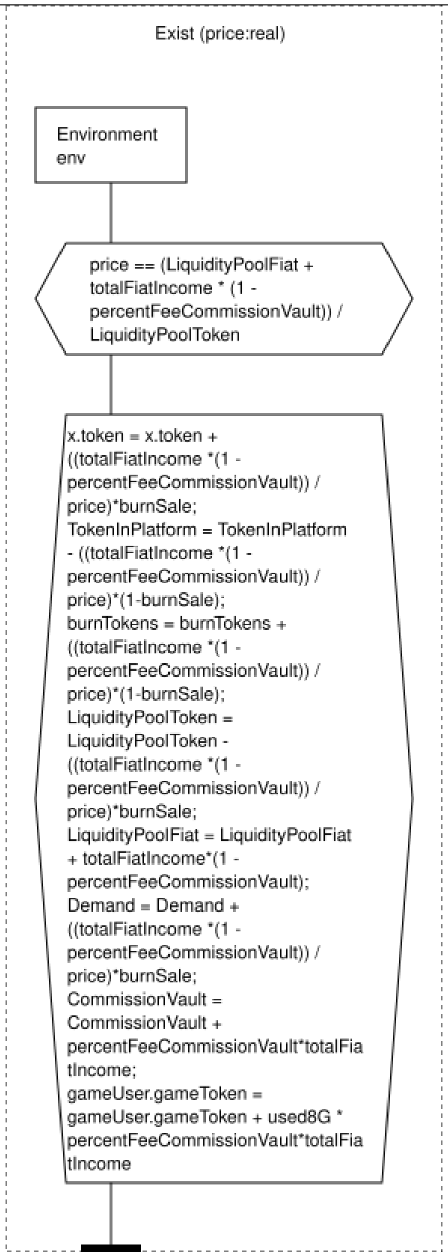
```
x.token = x.token +
((totalFiatIncome*(1 -
percentFeeCommissionVault) *
traderHoldFactor) / price)*burnSale;
TokenInPlatform = TokenInPlatform
- ((totalFiatIncome*(1 -
percentFeeCommissionVault) *
traderHoldFactor) /
price)*(1-burnSale);
burnTokens = burnTokens +
((totalFiatIncome*(1 -
percentFeeCommissionVault) *
traderHoldFactor) /
price)*(1-burnSale);
LiquidityPoolFiat = LiquidityPoolFiat
+ (1 - percentFeeCommissionVault)
* totalFiatIncome*traderHoldFactor;
LiquidityPoolToken =
LiquidityPoolToken - ((1 -
percentFeeCommissionVault)
*totalFiatIncome*traderHoldFactor)
/ price;
Demand = Demand + ((1 -
percentFeeCommissionVault)
*totalFiatIncome*traderHoldFactor)
/ price;
CommissionVault =
CommissionVault +
percentFeeCommissionVault*totalFia
tIncome*traderHoldFactor
```

salesHistorical (X) :	
print7 :	
buyMarketPlace (x) :	 Environment env 1 x.token = x.token - totalFiatIncome; GAME_TOKENS_8F = GAME_TOKENS_8F + percentShopProfit*totalFiatIncome; BURN = BURN + (1 - percentShopProfit)*totalFiatIncome

harvesting :	
	gameUser.token = gameUser.token
	+ totalFiatIncome*burnSale;
	TokenInPlatform = TokenInPlatform
	- totalFiatIncome*(1-burnSale);
	burnTokens = burnTokens +
	totalFiatIncome*(1-burnSale);
	GAME_TOKENS_8F =
	GAME_TOKENS_8F - totalFiatIncome

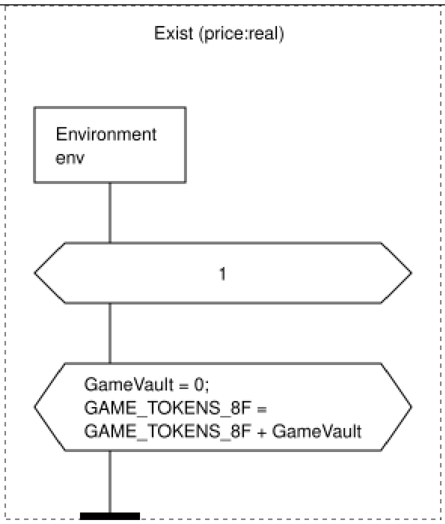
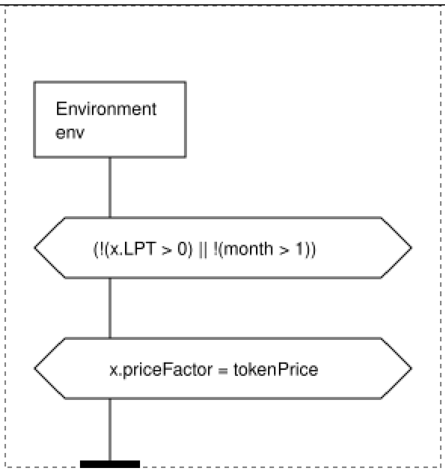
burn88days :	
	gameUser.token = (1 -
	coef88day)*gameUser.token;
	BURN = BURN +
	coef88day*gameUser.token

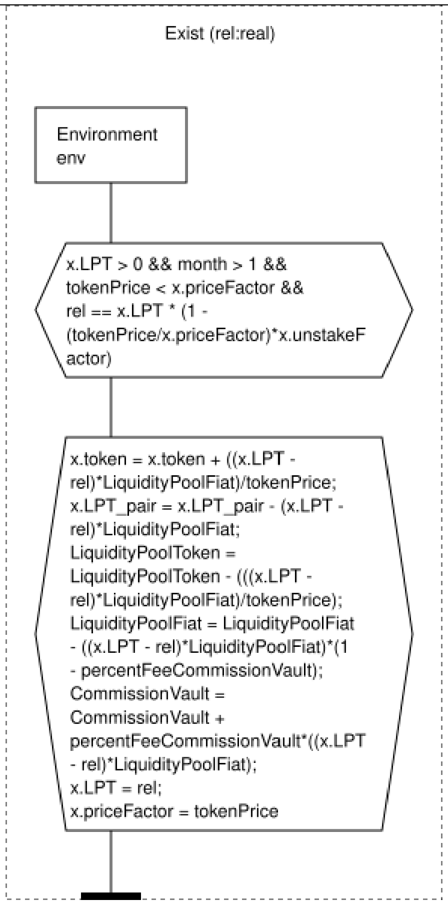
buy (x) :

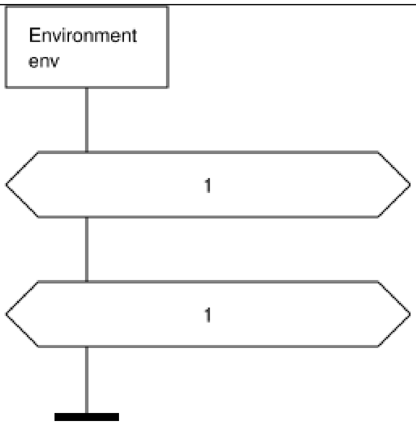


```
price == (LiquidityPoolFiat +
totalFiatIncome * (1 -
percentFeeCommissionVault)) /
LiquidityPoolToken
```

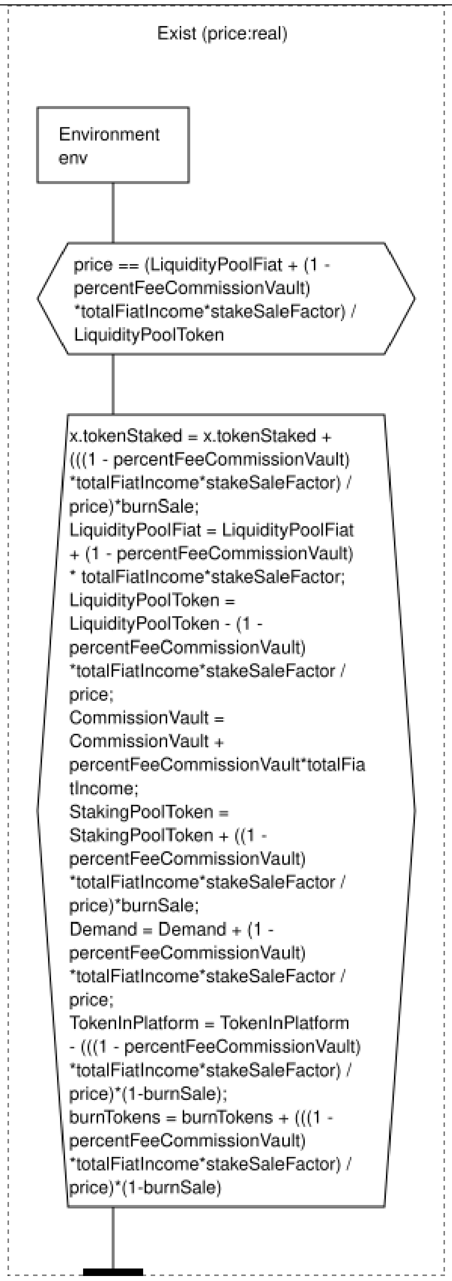
```
x.token = x.token +
((totalFiatIncome *(1 -
percentFeeCommissionVault)) /
price)*burnSale;
TokenInPlatform = TokenInPlatform
- ((totalFiatIncome *(1 -
percentFeeCommissionVault)) /
price)*(1-burnSale);
burnTokens = burnTokens +
((totalFiatIncome *(1 -
percentFeeCommissionVault)) /
price)*(1-burnSale);
LiquidityPoolToken =
LiquidityPoolToken -
((totalFiatIncome *(1 -
percentFeeCommissionVault)) /
price)*burnSale;
LiquidityPoolFiat = LiquidityPoolFiat
+ totalFiatIncome*(1 -
percentFeeCommissionVault);
Demand = Demand +
((totalFiatIncome *(1 -
percentFeeCommissionVault)) /
price)*burnSale;
CommissionVault =
CommissionVault +
percentFeeCommissionVault*totalFia
tIncome;
gameUser.gameToken =
gameUser.gameToken + used8G *
percentFeeCommissionVault*totalFia
tIncome
```

$GameOutput$:	 Exist (price:real) Environment env 1 GameVault = 0; GAME_TOKENS_8F = GAME_TOKENS_8F + GameVault
$noWithdrawFarming$ (x) :	 Environment env (!(x.LPT > 0) || !(month > 1)) x.priceFactor = tokenPrice

withdrawFarming (x) :	 Exist (rel:real)
	Environment env
	x.LPT > 0 && month > 1 && tokenPrice < x.priceFactor && rel == x.LPT * (1 - (tokenPrice/x.priceFactor)*x.unstakeFactor)
	x.token = x.token + ((x.LPT - rel)*LiquidityPoolFiat)/tokenPrice;
	x.LPT_pair = x.LPT_pair - (x.LPT - rel)*LiquidityPoolFiat;
	LiquidityPoolToken = LiquidityPoolToken - (((x.LPT - rel)*LiquidityPoolFiat)/tokenPrice);
	LiquidityPoolFiat = LiquidityPoolFiat - ((x.LPT - rel)*LiquidityPoolFiat)*(1 - percentFeeCommissionVault);
	CommissionVault = CommissionVault + percentFeeCommissionVault*((x.LPT - rel)*LiquidityPoolFiat);
	x.LPT = rel;
	x.priceFactor = tokenPrice

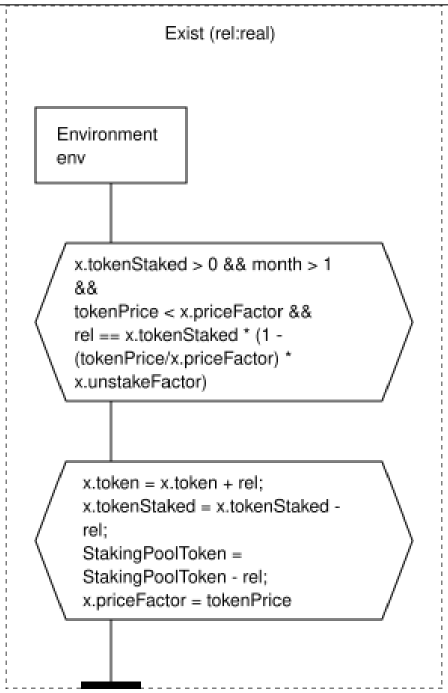
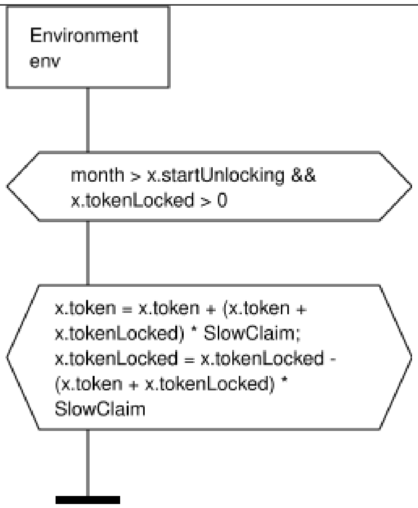
print5 :	

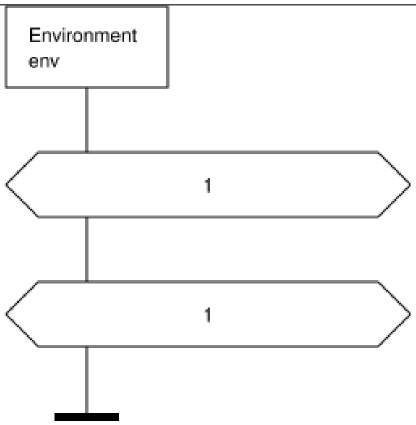
putStake (x) :



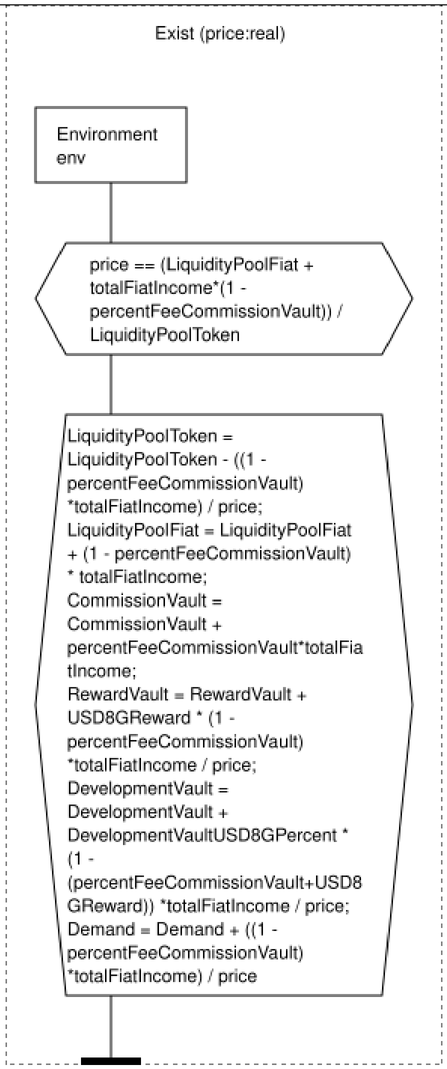
```
price == (LiquidityPoolFiat + (1 -
percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor) /
LiquidityPoolToken
```

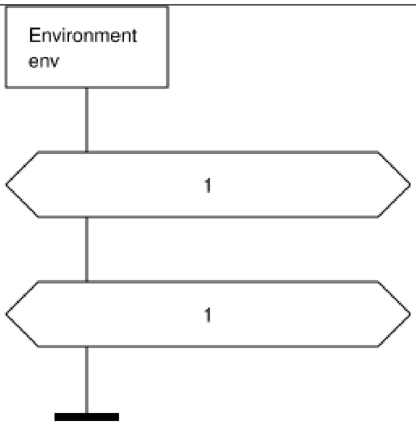
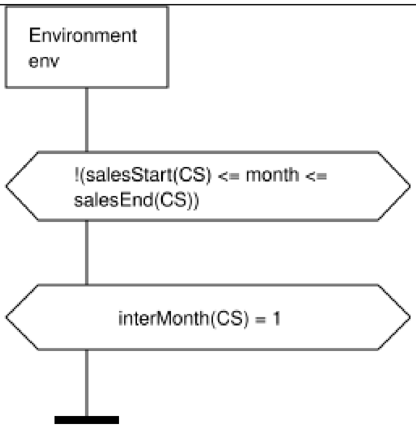
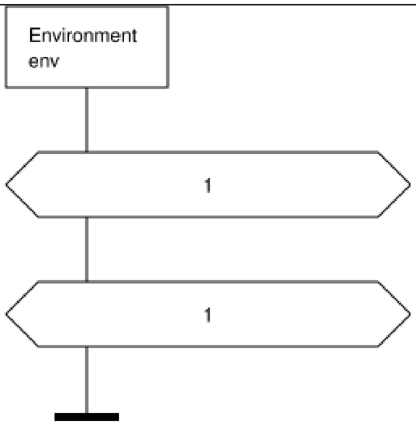
```
x.tokenStaked = x.tokenStaked +
(((1 - percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor) /
price)*burnSale;
LiquidityPoolFiat = LiquidityPoolFiat
+ (1 - percentFeeCommissionVault)
* totalFiatIncome*stakeSaleFactor;
LiquidityPoolToken =
LiquidityPoolToken - (1 -
percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor /
price;
CommissionVault =
CommissionVault +
percentFeeCommissionVault*totalFiatIncome;
StakingPoolToken =
StakingPoolToken + ((1 -
percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor /
price)*burnSale;
Demand = Demand + (1 -
percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor /
price;
TokenInPlatform = TokenInPlatform
- (((1 - percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor) /
price)*(1-burnSale);
burnTokens = burnTokens + (((1 -
percentFeeCommissionVault)
*totalFiatIncome*stakeSaleFactor) /
price)*(1-burnSale)
```

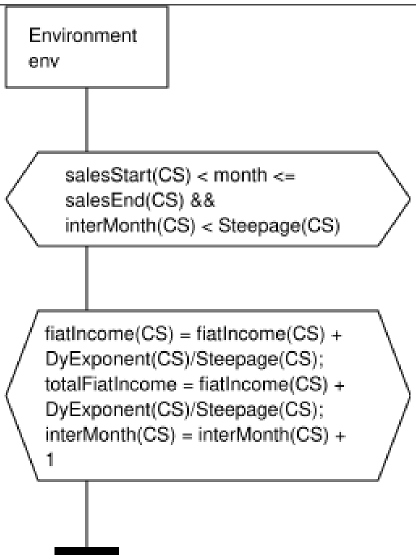
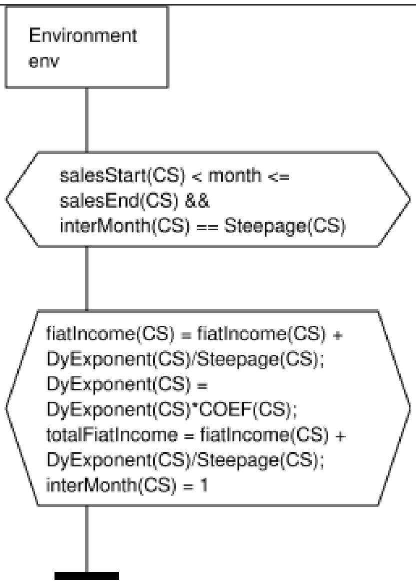
withdrawStake (x) :	 Exist (rel:real) Environment env x.tokenStaked > 0 && month > 1 && tokenPrice < x.priceFactor && rel == x.tokenStaked * (1 - (tokenPrice/x.priceFactor) * x.unstakeFactor) x.token = x.token + rel; x.tokenStaked = x.tokenStaked - rel; StakingPoolToken = StakingPoolToken - rel; x.priceFactor = tokenPrice
unlockSlowClaim (x) :	 Environment env month > x.startUnlocking && x.tokenLocked > 0 x.token = x.token + (x.token + x.tokenLocked) * SlowClaim; x.tokenLocked = x.tokenLocked - (x.token + x.tokenLocked) * SlowClaim

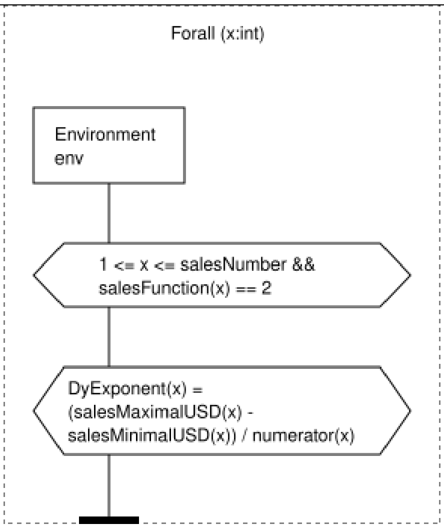
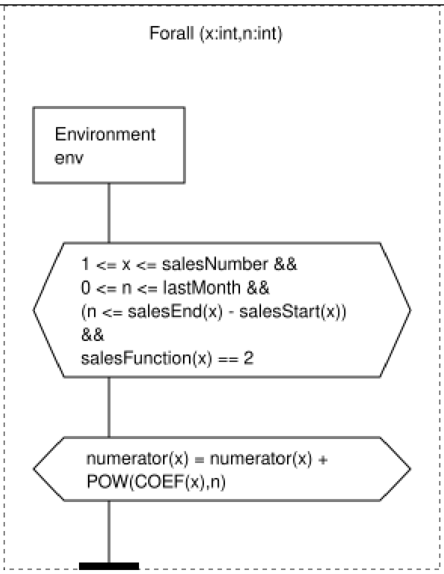
print3 :	

buy8G (x) :

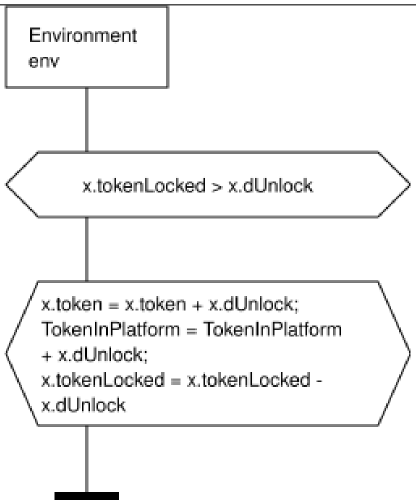

Exist (price:real)
Environment env
price == (LiquidityPoolFiat + totalFiatIncome*(1 - percentFeeCommissionVault)) / LiquidityPoolToken
LiquidityPoolToken = LiquidityPoolToken - ((1 - percentFeeCommissionVault) *totalFiatIncome) / price;
LiquidityPoolFiat = LiquidityPoolFiat + (1 - percentFeeCommissionVault) * totalFiatIncome;
CommissionVault = CommissionVault + percentFeeCommissionVault*totalFiatIncome;
RewardVault = RewardVault + USD8GReward * (1 - percentFeeCommissionVault) *totalFiatIncome / price;
DevelopmentVault = DevelopmentVault + DevelopmentVaultUSD8GPercent * (1 - (percentFeeCommissionVault+USD8GReward)) *totalFiatIncome / price;
Demand = Demand + ((1 - percentFeeCommissionVault) *totalFiatIncome) / price

print2 :	
noSales (CS) :	 Environment env !(salesStart(CS) <= month <= salesEnd(CS)) interMonth(CS) = 1
print1 :	

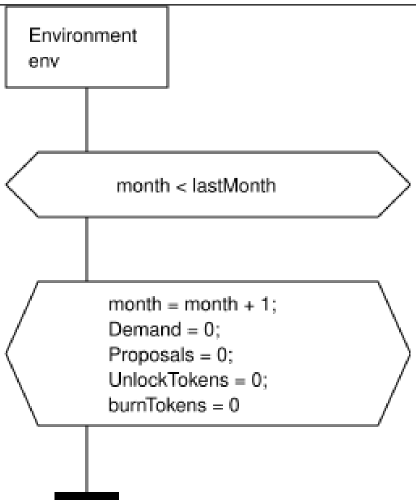
salesExponentInter (CS) :	 Environment env salesStart(CS) < month <= salesEnd(CS) && interMonth(CS) < Steepage(CS) fiatIncome(CS) = fiatIncome(CS) + DyExponent(CS)/Steepage(CS); totalFiatIncome = fiatIncome(CS) + DyExponent(CS)/Steepage(CS); interMonth(CS) = interMonth(CS) + 1
salesExponent (CS) :	 Environment env salesStart(CS) < month <= salesEnd(CS) && interMonth(CS) == Steepage(CS) fiatIncome(CS) = fiatIncome(CS) + DyExponent(CS)/Steepage(CS); DyExponent(CS) = DyExponent(CS)*COEF(CS); totalFiatIncome = fiatIncome(CS) + DyExponent(CS)/Steepage(CS); interMonth(CS) = 1

$initSaleDyComputation$:	 Forall (x:int) Environment env 1 <= x <= salesNumber && salesFunction(x) == 2 DyExponent(x) = (salesMaximalUSD(x) - salesMinimalUSD(x)) / numerator(x)
$initSaleComputation$:	 Forall (x:int,n:int) Environment env 1 <= x <= salesNumber && 0 <= n <= lastMonth && (n <= salesEnd(x) - salesStart(x)) && salesFunction(x) == 2 numerator(x) = numerator(x) + POW(COEF(x),n)

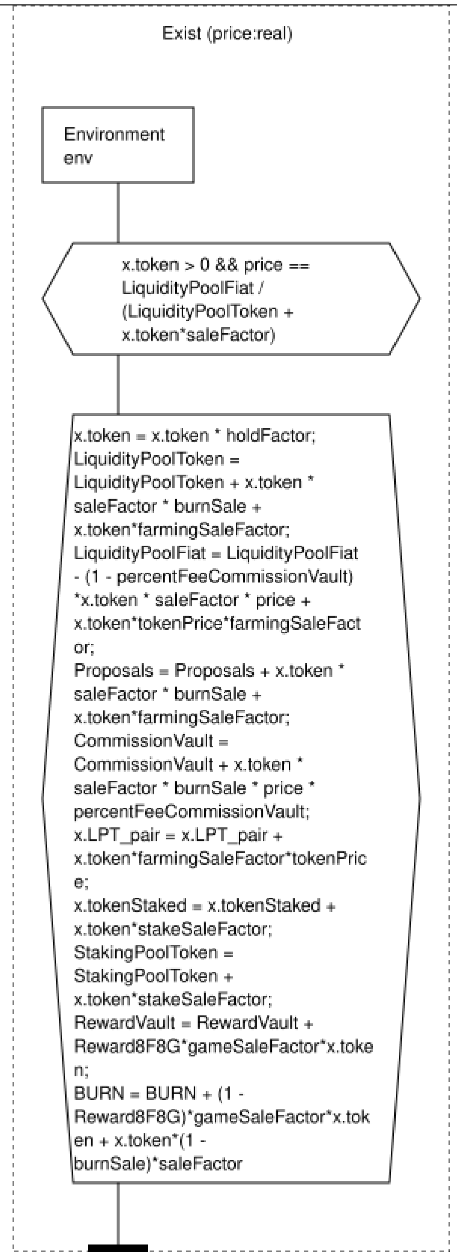
unlock (x) :


Environment
env
x.tokenLocked > x.dUnlock
x.token = x.token + x.dUnlock;
TokenInPlatform = TokenInPlatform + x.dUnlock;
x.tokenLocked = x.tokenLocked - x.dUnlock

nextMonth :

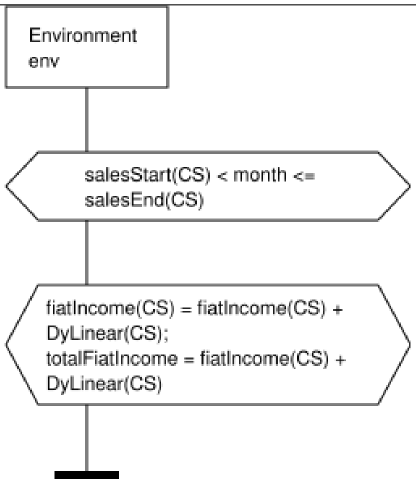


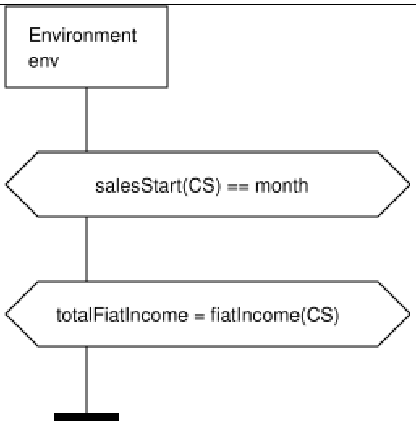
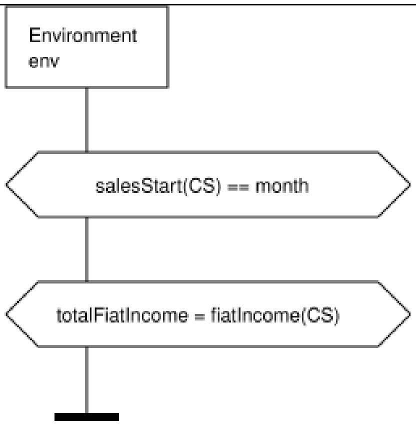
spendToken (x) :

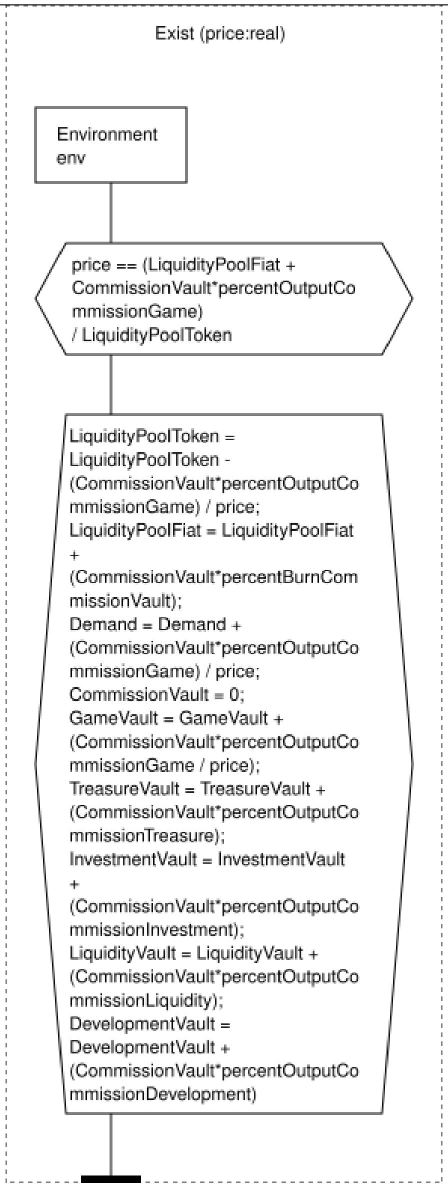


```
x.token > 0 && price == LiquidityPoolFiat / (LiquidityPoolToken + x.token*saleFactor)
```

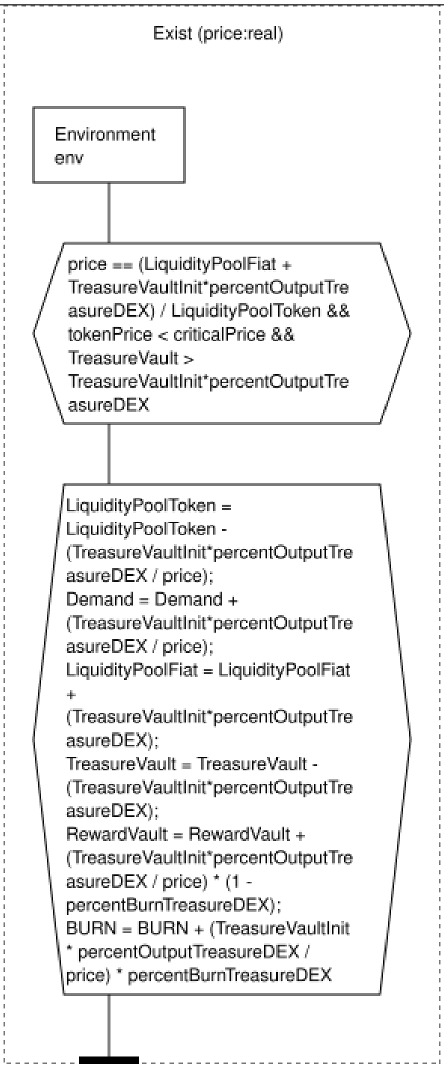
```
x.token = x.token * holdFactor;
LiquidityPoolToken = LiquidityPoolToken + x.token * saleFactor * burnSale + x.token*farmingSaleFactor;
LiquidityPoolFiat = LiquidityPoolFiat - (1 - percentFeeCommissionVault) *x.token * saleFactor * price + x.token*tokenPrice*farmingSaleFactor;
Proposals = Proposals + x.token * saleFactor * burnSale + x.token*farmingSaleFactor;
CommissionVault = CommissionVault + x.token * saleFactor * burnSale * price * percentFeeCommissionVault;
x.LPT_pair = x.LPT_pair + x.token*farmingSaleFactor*tokenPrice;
x.tokenStaked = x.tokenStaked + x.token*stakeSaleFactor;
StakingPoolToken = StakingPoolToken + x.token*stakeSaleFactor;
RewardVault = RewardVault + Reward8F8G*gameSaleFactor*x.token;
BURN = BURN + (1 - Reward8F8G)*gameSaleFactor*x.token + x.token*(1 - burnSale)*saleFactor
```

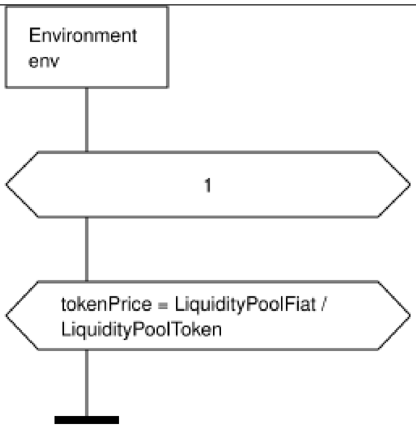
salesLinear (CS) :	 Environment
	env
	salesStart(CS) < month <=
	salesEnd(CS)
	fiatIncome(CS) = fiatIncome(CS) +
	DyLinear(CS);
	totalFiatIncome = fiatIncome(CS) +
	DyLinear(CS)

salesExponentInit (CS) :	 Environment env salesStart(CS) == month totalFiatIncome = fiatIncome(CS)
salesLinearInit (CS) :	 Environment env salesStart(CS) == month totalFiatIncome = fiatIncome(CS)

$CommissionOutput$:	

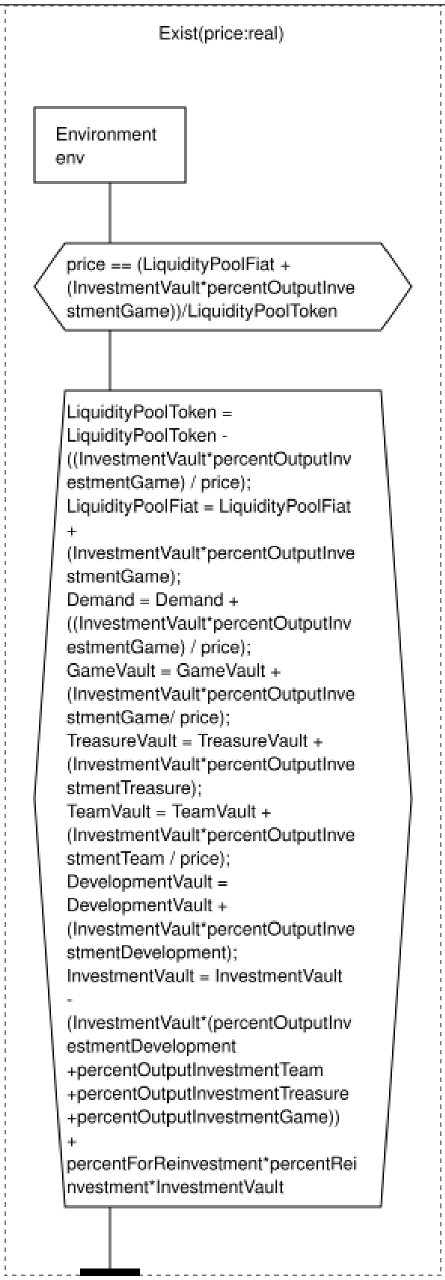
Exist (price:real)
Environment env
price == (LiquidityPoolFiat + CommissionVault*percentOutputCommissionGame) / LiquidityPoolToken
LiquidityPoolToken = LiquidityPoolToken - (CommissionVault*percentOutputCommissionGame) / price;
LiquidityPoolFiat = LiquidityPoolFiat + (CommissionVault*percentBurnCommissionVault);
Demand = Demand + (CommissionVault*percentOutputCommissionGame) / price;
CommissionVault = 0;
GameVault = GameVault + (CommissionVault*percentOutputCommissionGame / price);
TreasureVault = TreasureVault + (CommissionVault*percentOutputCommissionTreasure);
InvestmentVault = InvestmentVault + (CommissionVault*percentOutputCommissionInvestment);
LiquidityVault = LiquidityVault + (CommissionVault*percentOutputCommissionLiquidity);
DevelopmentVault = DevelopmentVault + (CommissionVault*percentOutputCommissionDevelopment)

$TreasureOutput$:	 Exist (price:real)
	Environment env
	price == (LiquidityPoolFiat + TreasureVaultInit*percentOutputTreasureDEX) / LiquidityPoolToken && tokenPrice < criticalPrice && TreasureVault > TreasureVaultInit*percentOutputTreasureDEX
	LiquidityPoolToken = LiquidityPoolToken - (TreasureVaultInit*percentOutputTreasureDEX / price);
	Demand = Demand + (TreasureVaultInit*percentOutputTreasureDEX / price);
	LiquidityPoolFiat = LiquidityPoolFiat + (TreasureVaultInit*percentOutputTreasureDEX);
	TreasureVault = TreasureVault - (TreasureVaultInit*percentOutputTreasureDEX);
	RewardVault = RewardVault + (TreasureVaultInit*percentOutputTreasureDEX / price) * (1 - percentBurnTreasureDEX);
	BURN = BURN + (TreasureVaultInit * percentOutputTreasureDEX / price) * percentBurnTreasureDEX

$newPrice\ \ :$	

Environment
env
1
tokenPrice = LiquidityPoolFiat /
LiquidityPoolToken

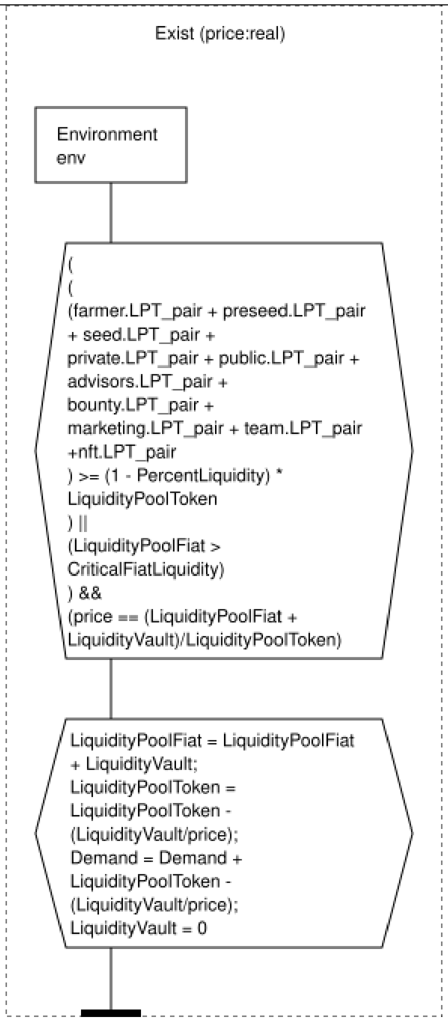
$InvestmentOutput$:



```
price == (LiquidityPoolFiat + (InvestmentVault*percentOutputInvestmentGame))/LiquidityPoolToken
```

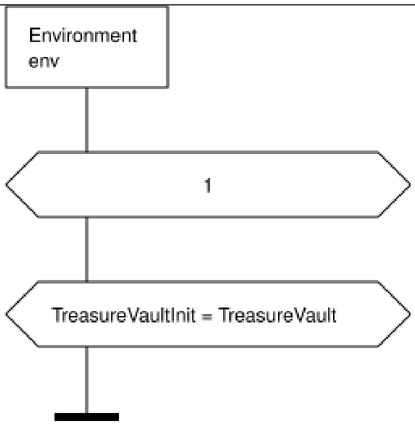
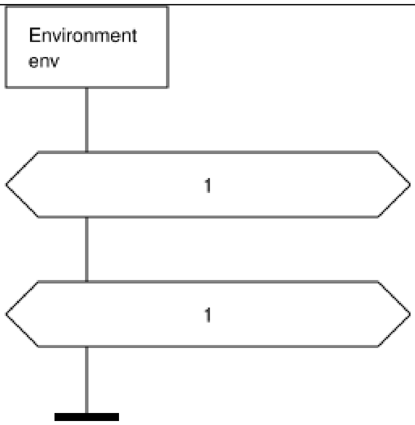
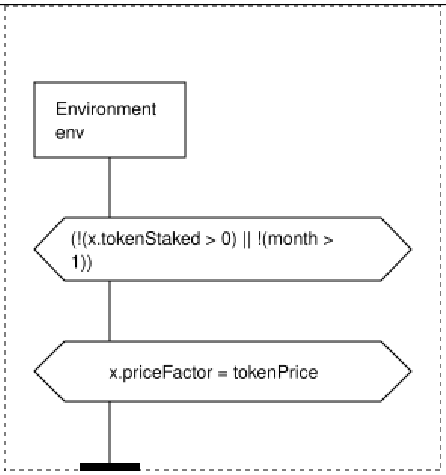
```
LiquidityPoolToken = LiquidityPoolToken - ((InvestmentVault*percentOutputInvestmentGame) / price);
LiquidityPoolFiat = LiquidityPoolFiat + (InvestmentVault*percentOutputInvestmentGame);
Demand = Demand + ((InvestmentVault*percentOutputInvestmentGame) / price);
GameVault = GameVault + (InvestmentVault*percentOutputInvestmentGame/ price);
TreasureVault = TreasureVault + (InvestmentVault*percentOutputInvestmentTreasure);
TeamVault = TeamVault + (InvestmentVault*percentOutputInvestmentTeam / price);
DevelopmentVault = DevelopmentVault + (InvestmentVault*percentOutputInvestmentDevelopment);
InvestmentVault = InvestmentVault - (InvestmentVault*(percentOutputInvestmentDevelopment +percentOutputInvestmentTeam +percentOutputInvestmentTreasure +percentOutputInvestmentGame)) + percentForReinvestment*percentReinvestment*InvestmentVault
```

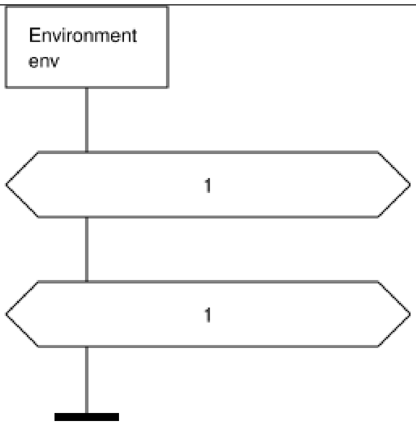
$LiquidityOutputLp$:

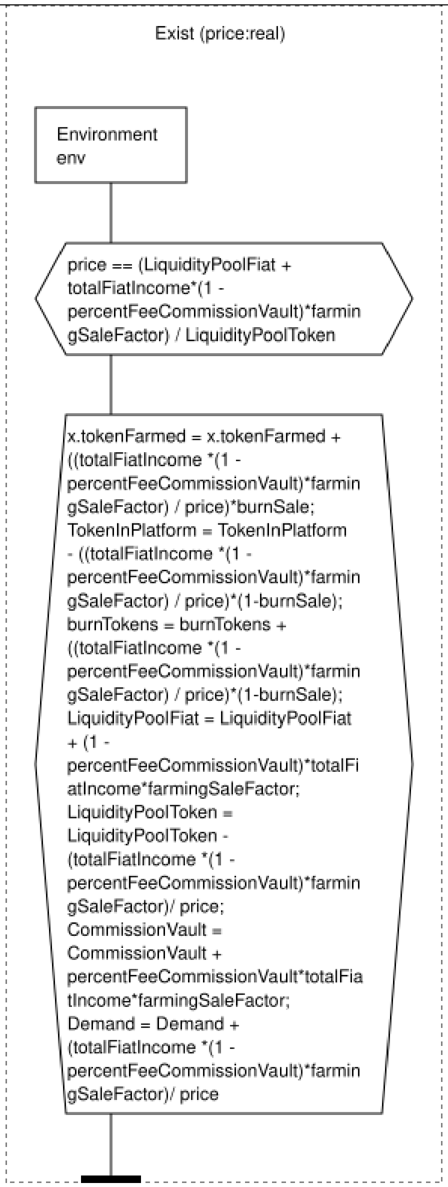


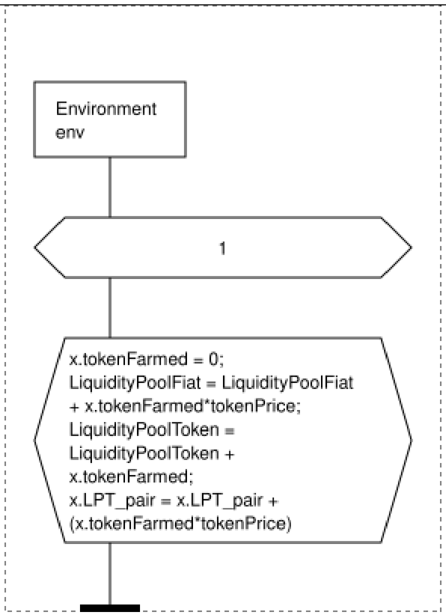
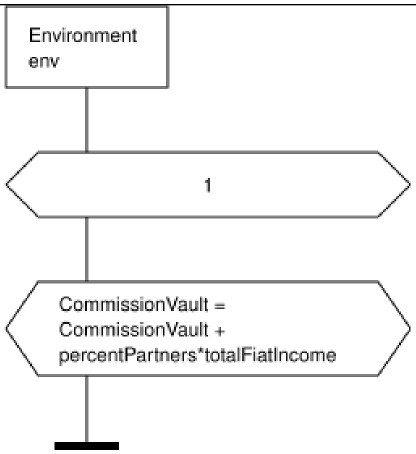
```
(
(
(farmer.LPT_pair + preseed.LPT_pair
+ seed.LPT_pair +
private.LPT_pair + public.LPT_pair +
advisors.LPT_pair +
bounty.LPT_pair +
marketing.LPT_pair + team.LPT_pair
+nft.LPT_pair
) >= (1 - PercentLiquidity) *
LiquidityPoolToken
) ||
(LiquidityPoolFiat >
CriticalFiatLiquidity)
) &&
(price == (LiquidityPoolFiat +
LiquidityVault)/LiquidityPoolToken)
```

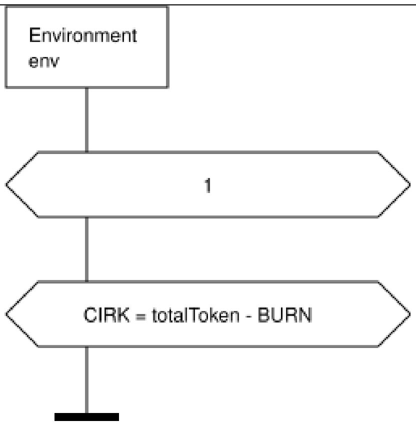
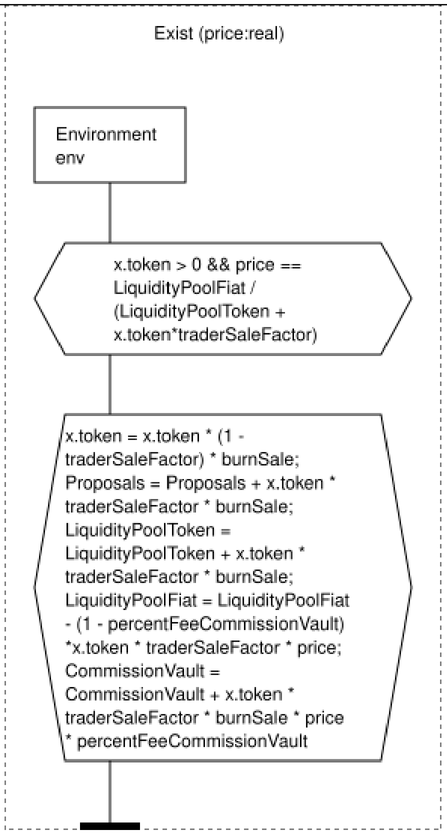
```
LiquidityPoolFiat = LiquidityPoolFiat
+ LiquidityVault;
LiquidityPoolToken =
LiquidityPoolToken -
(LiquidityVault/price);
Demand = Demand +
LiquidityPoolToken -
(LiquidityVault/price);
LiquidityVault = 0
```

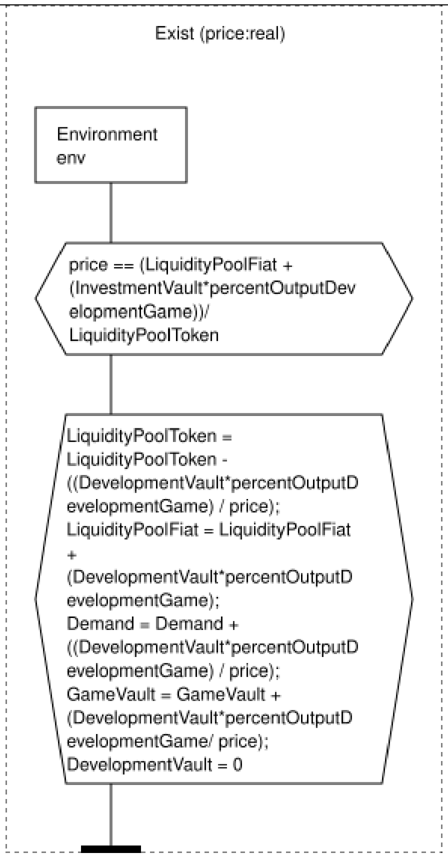
initTreasure :	 Environment env 1 TreasureVaultInit = TreasureVault
print4 :	
noWithdrawStake (x) :	 Environment env (!(x.tokenStaked > 0) || !(month > 1)) x.priceFactor = tokenPrice

$print6$:	

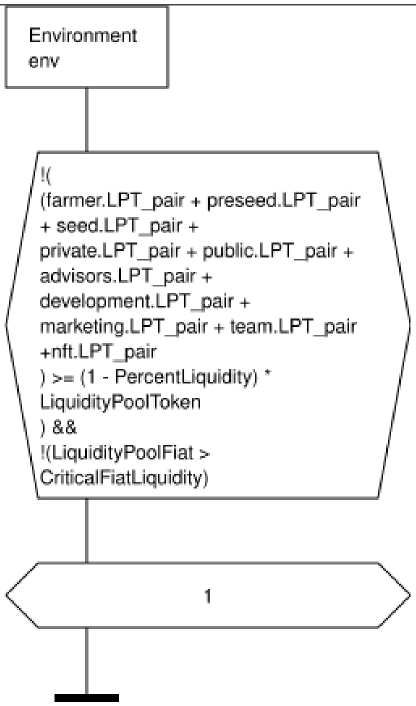
$buyTokenFarming$ (x) :	 Exist (price:real)
	Environment env
	price == (LiquidityPoolFiat + totalFiatIncome*(1 - percentFeeCommissionVault)*farmingSaleFactor) / LiquidityPoolToken
	x.tokenFarmed = x.tokenFarmed + ((totalFiatIncome *(1 - percentFeeCommissionVault)*farmingSaleFactor) / price)*burnSale;
	TokenInPlatform = TokenInPlatform - ((totalFiatIncome *(1 - percentFeeCommissionVault)*farmingSaleFactor) / price)*(1-burnSale);
	burnTokens = burnTokens + ((totalFiatIncome *(1 - percentFeeCommissionVault)*farmingSaleFactor) / price)*(1-burnSale);
	LiquidityPoolFiat = LiquidityPoolFiat + (1 - percentFeeCommissionVault)*totalFiatIncome*farmingSaleFactor;
	LiquidityPoolToken = LiquidityPoolToken - (totalFiatIncome *(1 - percentFeeCommissionVault)*farmingSaleFactor)/ price;
	CommissionVault = CommissionVault + percentFeeCommissionVault*totalFiatIncome*farmingSaleFactor;
	Demand = Demand + (totalFiatIncome *(1 - percentFeeCommissionVault)*farmingSaleFactor)/ price

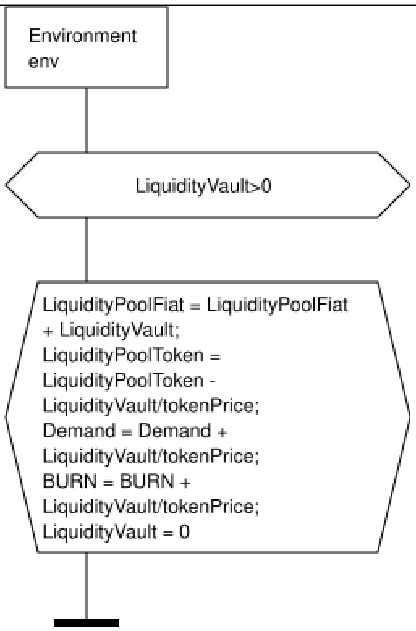
putLiquidity (x) :	 Environment env 1 x.tokenFarmed = 0; LiquidityPoolFiat = LiquidityPoolFiat + x.tokenFarmed*tokenPrice; LiquidityPoolToken = LiquidityPoolToken + x.tokenFarmed; x.LPT_pair = x.LPT_pair + (x.tokenFarmed*tokenPrice)
partners :	 Environment env 1 CommissionVault = CommissionVault + percentPartners*totalFiatIncome

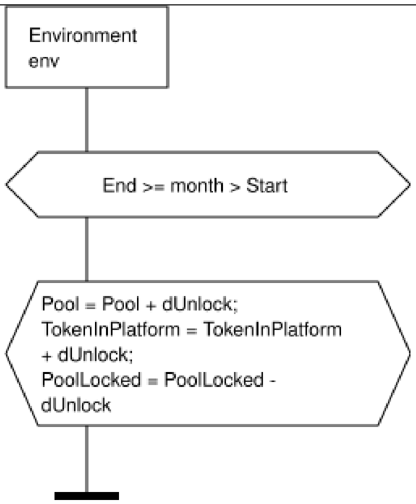
printBURN :	
saleToken (x) :	 Exist (price:real) Environment env x.token > 0 && price == LiquidityPoolFiat / (LiquidityPoolToken + x.token*traderSaleFactor) x.token = x.token * (1 - traderSaleFactor) * burnSale; Proposals = Proposals + x.token * traderSaleFactor * burnSale; LiquidityPoolToken = LiquidityPoolToken + x.token * traderSaleFactor * burnSale; LiquidityPoolFiat = LiquidityPoolFiat - (1 - percentFeeCommissionVault) *x.token * traderSaleFactor * price; CommissionVault = CommissionVault + x.token * traderSaleFactor * burnSale * price * percentFeeCommissionVault

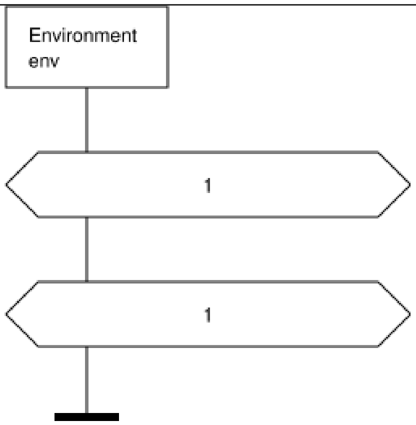
$DevelopmentOutput$:	

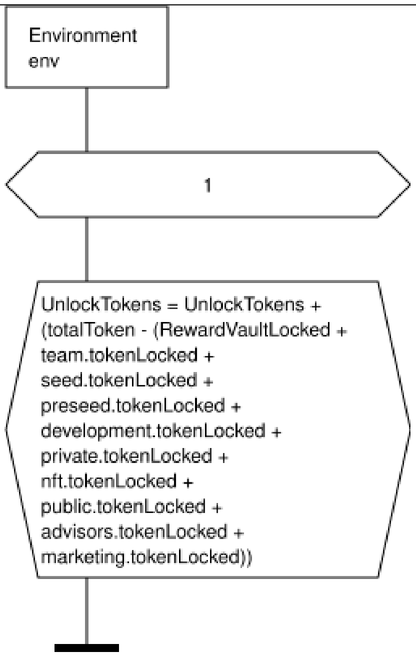
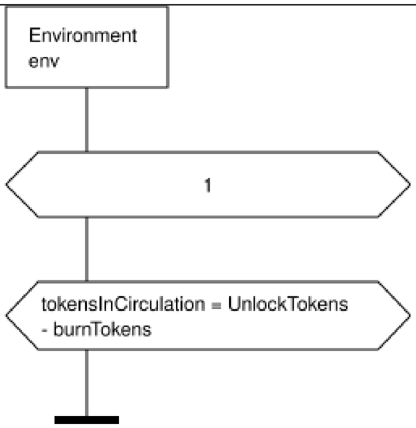
Exist (price:real)
Environment env
price == (LiquidityPoolFiat + (InvestmentVault*percentOutputDevelopmentGame))/ LiquidityPoolToken
LiquidityPoolToken = LiquidityPoolToken - ((DevelopmentVault*percentOutputDevelopmentGame) / price);
LiquidityPoolFiat = LiquidityPoolFiat + (DevelopmentVault*percentOutputDevelopmentGame);
Demand = Demand + ((DevelopmentVault*percentOutputDevelopmentGame) / price);
GameVault = GameVault + (DevelopmentVault*percentOutputDevelopmentGame/ price);
DevelopmentVault = 0

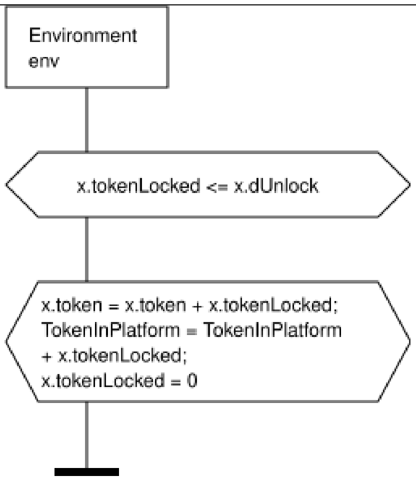
$notLiquidityOutputLp$:	 Environment env !((farmer.LPT_pair + preseed.LPT_pair + seed.LPT_pair + private.LPT_pair + public.LPT_pair + advisors.LPT_pair + development.LPT_pair + marketing.LPT_pair + team.LPT_pair +nft.LPT_pair) >= (1 - PercentLiquidity) * LiquidityPoolToken) && !(LiquidityPoolFiat > CriticalFiatLiquidity) 1

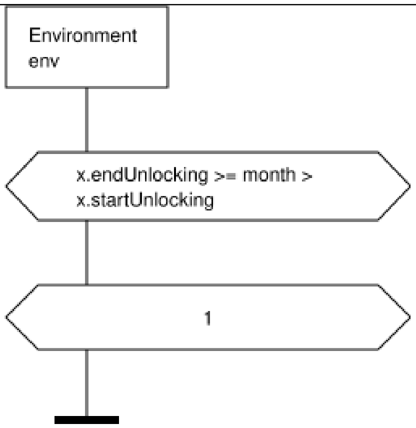
$LiquidityOutputBurn$:	 Environment
	env
	LiquidityVault>0
	LiquidityPoolFiat = LiquidityPoolFiat
	+ LiquidityVault;
	LiquidityPoolToken =
	LiquidityPoolToken -
	LiquidityVault/tokenPrice;
	Demand = Demand +
	LiquidityVault/tokenPrice;
	BURN = BURN +
	LiquidityVault/tokenPrice;
	LiquidityVault = 0

unlockPool (Pool , PoolLocked , End , Start , dUnlock) :	 Environment
	env
	End >= month > Start
	Pool = Pool + dUnlock;
	TokenInPlatform = TokenInPlatform
	+ dUnlock;
	PoolLocked = PoolLocked -
	dUnlock

print8 :	

unlockTokensNum :	 Environment env 1 UnlockTokens = UnlockTokens + (totalToken - (RewardVaultLocked + team.tokenLocked + seed.tokenLocked + preseed.tokenLocked + development.tokenLocked + private.tokenLocked + nft.tokenLocked + public.tokenLocked + advisors.tokenLocked + marketing.tokenLocked))
8FInCirculation :	 Environment env 1 tokensInCirculation = UnlockTokens - burnTokens

unlockend (x) :	 Environment env
	x.tokenLocked <= x.dUnlock
	x.token = x.token + x.tokenLocked;
	TokenInPlatform = TokenInPlatform
	+ x.tokenLocked;
	x.tokenLocked = 0

startUnlock (x) :	 Environment env x.endUnlocking >= month > x.startUnlocking 1

0.6 Behaviours

B0 = initSaleComputation . initSaleDyComputation . B1 ,

B1 =
(UNLOCKING) ;
(unlockTokensNum) ;
(SALES) ;
(TOKEN_CIRCULATION) ;
(VAULTS) ;
(STAKING) ;
(FARMING) ;
(GAMING) ;
(printBURN) ;
(8FInCirculation) ;

(print10) ;
(nextMonth . B1 + ¬ nextMonth . Delta)
) ,

UNLOCKING =

(initUnlock (team) + ¬ initUnlock (team)) ;
(initUnlock (marketing) + ¬ initUnlock (marketing)) ;
(initUnlock (seed) + ¬ initUnlock (seed)) ;
(initUnlock (preseed) + ¬ initUnlock (preseed)) ;
(initUnlock (development) + ¬ initUnlock (development)) ;
(initUnlock (private) + ¬ initUnlock (private)) ;
(initUnlock (public) + ¬ initUnlock (public)) ;
(initUnlock (advisors) + ¬ initUnlock (advisors)) ;
(initUnlock (nft) + ¬ initUnlock (nft)) ;

unlockPool (RewardVault , RewardVaultdUnlock , RewardVaultEnd , RewardVaultStart , RewardVaultdUnlock) +
¬ unlockPool (RewardVault , RewardVaultdUnlock , RewardVaultEnd , RewardVaultStart , RewardVaultdUnlock)
) ;

((startUnlock (team) ; (unlock (team) + unlockend (team))) + ¬ startUnlock (team)) ;
((startUnlock (marketing) ; (unlock (marketing) + unlockend (marketing))) + ¬ startUnlock (marketing)) ;
((startUnlock (seed) ; (unlock (seed) + unlockend (seed))) + ¬ startUnlock (seed)) ;
((startUnlock (preseed) ; (unlock (preseed) + unlockend (preseed))) + ¬ startUnlock (preseed)) ;
((startUnlock (development) ; (unlock (development) + unlockend (development))) + ¬ startUnlock (development)) ;
((startUnlock (private) ; (unlock (private) + unlockend (private))) + ¬ startUnlock (private)) ;
((startUnlock (public) ; (unlock (public) + unlockend (public))) + ¬ startUnlock (public)) ;
((startUnlock (advisors) ; (unlock (advisors) + unlockend (advisors))) + ¬ startUnlock (advisors)) ;
((startUnlock (nft) ; (unlock (nft) + unlockend (nft))) + ¬ startUnlock (nft))

) ,

STAKING =

(rewardStake (stakeUser) + ¬ rewardStake (stakeUser)) ;
(withdrawStake (stakeUser) + withdrawStakeNatural (stakeUser) + noWithdrawStake (stakeUser)) ;
(rewardStake (team) + ¬ rewardStake (team)) ;
(withdrawStake (team) + withdrawStakeNatural (team) + noWithdrawStake (team)) ;
(rewardStake (seed) + ¬ rewardStake (seed)) ;
(withdrawStake (seed) + withdrawStakeNatural (seed) + noWithdrawStake (seed)) ;
(rewardStake (development) + ¬ rewardStake (development)) ;
(withdrawStake (development) + withdrawStakeNatural (development) + noWithdrawStake (development)) ;
(rewardStake (preseed) + ¬ rewardStake (preseed)) ;
(withdrawStake (preseed) + withdrawStakeNatural (preseed) + noWithdrawStake (preseed)) ;
(rewardStake (public) + ¬ rewardStake (public)) ;
(withdrawStake (public) + withdrawStakeNatural (public) + noWithdrawStake (public)) ;
(rewardStake (private) + ¬ rewardStake (private)) ;
(withdrawStake (private) + withdrawStakeNatural (private) + noWithdrawStake (private)) ;
(rewardStake (advisors) + ¬ rewardStake (advisors)) ;
(withdrawStake (advisors) + withdrawStakeNatural (advisors) + noWithdrawStake (advisors)) ;
(rewardStake (marketing) + ¬ rewardStake (marketing)) ;
(withdrawStake (marketing) + withdrawStakeNatural (marketing) + noWithdrawStake (marketing)) ;
(rewardStake (nft) + ¬ rewardStake (nft)) ;
(withdrawStake (nft) + withdrawStakeNatural (nft) + noWithdrawStake (nft))
) ,

FARMING =
(rewardFarm (farmer) + ¬ rewardFarm (farmer)) ;
(withdrawFarming (farmer) + withdrawFarmingNatural (farmer) + noWithdrawFarming (farmer)) ;
(recalcLPT + ¬ recalcLPT) ;
(rewardFarm (team) + ¬ rewardFarm (team)) ;
(withdrawFarming (team) + withdrawFarmingNatural (team) + noWithdrawFarming (team)) ;
(recalcLPT + ¬ recalcLPT) ;
(rewardFarm (seed) + ¬ rewardFarm (seed)) ;
(withdrawFarming (seed) + withdrawFarmingNatural (seed) + noWithdrawFarming (seed)) ;
(recalcLPT + ¬ recalcLPT) ;
(rewardFarm (preseed) + ¬ rewardFarm (preseed)) ;
(withdrawFarming (preseed) + withdrawFarmingNatural (preseed) + noWi-

```
thdrawFarming (preseed)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (development) + ¬ rewardFarm (development)) ;
(withdrawFarming (development) + withdrawFarmingNatural (development) + noWithdrawFarming (development)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (public) + ¬ rewardFarm (public)) ;
(withdrawFarming (public) + withdrawFarmingNatural (public) + noWithdrawFarming (public)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (private) + ¬ rewardFarm (private)) ;
(withdrawFarming (private) + withdrawFarmingNatural (private) + noWithdrawFarming (private)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (advisors) + ¬ rewardFarm (advisors)) ;
(withdrawFarming (advisors) + withdrawFarmingNatural (advisors) + noWithdrawFarming (advisors)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (marketing) + ¬ rewardFarm (marketing)) ;
(withdrawFarming (marketing) + withdrawFarmingNatural (marketing) + noWithdrawFarming (marketing)) ;
( recalcLPT + ¬ recalcLPT ) ;
(rewardFarm (nft) + ¬ rewardFarm (nft)) ;
(withdrawFarming (nft) + withdrawFarmingNatural (nft) + noWithdrawFarming (nft)) ;
( recalcLPT + ¬ recalcLPT )
) ,

TOKEN_CIRCULATION =
(spendToken (team) + ¬ spendToken (team)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (seed) + ¬ spendToken (seed)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (preseed) + ¬ spendToken (preseed)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (development) + ¬ spendToken (development)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (public) + ¬ spendToken (public)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (private) + ¬ spendToken (private)) ;
(newPrice) ;
```

```
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (advisors) + ¬ spendToken (advisors)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (marketing) + ¬ spendToken (marketing)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(spendToken (nft) + ¬ spendToken (nft)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(saleToken (stakeUser) + ¬ saleToken (stakeUser)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(saleToken (farmer) + ¬ saleToken (farmer)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(saleToken (trader) + ¬ saleToken (trader)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(print7)
) ,

SALES =
(salesExponentInit (1) + salesExponent (1) + salesExponentInter (1) + noSales
(1)) ;
(salesLinearInit (2) + salesLinear (2) + noSales (2)) ;
(buy8G (gameUser)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(partners) ;
(print1) ;
(salesExponentInit (3) + salesExponent (3) + salesExponentInter (3) + noSales
(3)) ;
(salesLinearInit (4) + salesLinear (4) + noSales (4)) ;
(putStake (stakeUser)) ;
(newPrice) ;
(print5) ;
(buyTokenFarming (farmer)) ;
(newPrice) ;
(putLiquidity (farmer)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(buyTokenWallet (trader)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(print6)
```

```
) ,

VAULTS =
(CommissionOutput) ;
(newPrice) ;
(initTreasure) ;
(TREASURE_RANSOM) ;
(newPrice) ;
(InvestmentOutput) ;
(newPrice) ;
(print9) ;
( LiquidityOutputLp +   notLiquidityOutputLp ) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(print3) ;
(GameOutput)

) ,

TREASURE_RANSOM =
TreasureOutput .    newPrice . print4  . TREASURE_RANSOM + ¬ TreasureOutput
) ,

GAMING =
(salesExponentInit (7) + salesExponent (7) + salesExponentInter (7) + noSales
(7)) ;
(salesLinearInit (8) + salesLinear (8) + noSales (8)) ;
(harvesting) ;
(print8) ;
(salesExponentInit (9) + salesExponent (9) + salesExponentInter (9) + noSales
(9)) ;
(salesLinearInit (10) + salesLinear (10) + noSales (10)) ;
(buyMarketPlace (gameUser)) ;
(newPrice) ;
( recalcLPT + ¬ recalcLPT ) ;
(print2) ;
(burn88days)
)
```