

BIP-X: Sunrise Improvements

Proposer

Beanstalk Farms, Chaikitty

Proposer Wallet: [\[UPDATE\]](#)

Summary

- Implement a Dutch (Morning) auction that adjusts the Temperature that Beanstalk issues Pods at during the first 25 blocks after the `sunrise`;
- Refine the base `sunrise` incentive reward to account for current network and Beanstalk conditions; and
- Make `sunrise` interoperable with the `farm` function.

Links

- [BIP-X GitHub PR \[UPDATE\]](#)
- [GitHub Commit Hash: \[UPDATE\]](#)
- [Safe Transaction \[UPDATE\]](#)

Problem

During times of short-term excess demand for Soil, Beanstalk is paying significantly more in Pods than is necessary to attract creditors. In addition, Pod minting is concentrated towards Farmers who are able to execute transactions on-chain first when the demand for Soil is excessively high.

Beanstalk frequently overpays for the `sunrise`, as most of the reward is used to pay for high gas prices that are due to the competitiveness of calling `sunrise`.

The `sunrise` function is not interoperable with the `farm` function, as the function does not currently return any data.

Proposed Solution

Morning Auction

We propose a Dutch auction for Temperature in the Field during the first $\$b\$$ blocks after the `sunrise` (Morning) of each Season. Each Morning, the Temperature ramps from 1% to 100% of the maximum Temperature.

With this change, the Temperature offered by Beanstalk is dynamic within a Season. In practice, when $\text{deltaB} > 0$, the amount of Soil issued in a Season is dynamic, while the maximum number Pods that can be issued if all Soil is Sown in the Morning is fixed. When $\text{deltaB} < 0$, the amount of Soil issued in a Season is fixed, while the maximum number of Pods that can be issued if all Soil is Sown in the Morning is dynamic.

Before Morning Auction

Parameters within a given Season	$\text{deltaB} > 0$	$\text{deltaB} < 0$
Soil issued	Fixed	Fixed
Maximum number of Pods that can be minted*	Fixed	Fixed
Temperature	Fixed	Fixed

*If all Soil is Sown

Morning Auction

Parameters within a given Season	$\delta B > 0$	$\delta B < 0$
Soil issued	Dynamic	Fixed
Maximum number of Pods that can be minted**	Fixed	Dynamic
Temperature	Dynamic	Dynamic

**If all Soil is Sown *in the Morning*

Specification

1. Redefine all instances of h_t as the maximum Temperature h_t^{max} .
2. Redefine h to:

$$h = \begin{cases} 1 & \text{if } c = 0 \\ \max(h_t^{\text{max}} * \log_{ab+1}(ac+1), 1) & \text{if } 0 < c < b \\ h_t^{\text{max}} & \text{else} \end{cases}$$

Where a is the control variable, b is the number of blocks needed to elapse to reach the maximum Temperature, and c is the number of blocks that have elapsed since the `sunrise` call.

$$a = \lfloor \frac{b}{2} \rfloor \quad b = 25 \quad c = \text{currentBlock} - \text{sunriseBlock}$$

3. Adjust the time in which Beanstalk considers demand for Soil increasing in instances where all but at most one Soil is Sown from 5 minutes to 10 minutes since `sunrise` (i.e., $\Delta E_{t-1}^{u^{\text{first}}} \leq 600$).
4. When $\delta B > 0$, fix the maximum Pods that can be issued during a Season based on the number of Pods that became Harvestable at the start of that Season.

Due to function changes in `LibDibbler.sol`, the Fundraiser Facet must also be updated.

Storage

The following changes are made to Beanstalk's storage:

- Add to the `Season` struct: `uint32 sunriseBlock;`
- Add to the `Season` struct: `bool AbovePeg;`
- Add to the `Field` struct: `uint128 beanSown;`
- Modify `soil` in the `Season` struct from datatype `uint256` to `uint128`; and
- Deprecate `startSoil` from the `Weather` struct in favor of `beanSown`.

Sunrise Incentive Adjustment

We propose changing the base `sunrise` incentive reward to account for current network and Beanstalk conditions.

The following logic is used to calculate the incentive reward:

```
gasUsed = min(deltaGas + GAS_OVERHEAD, MAX_SUNRISE_GAS)
gasFee = block.basefee + PRIORITY_FEE_BUFFER

baseReward = max(MIN_REWARD, min(gasFee * gasUsed * beanEtherPrice + BASE_REWARD, MAX_REWARD))
sunriseReward = baseReward * 1.01^(blocksLate * SECONDS_PER_BLOCK)
```

Where: * `deltaGas` is equal to the delta of `gasleft` at the start and end of transaction; * `GAS_OVERHEAD` is the estimated gas

overhead not included in `deltaGas` (21k plus gas cost to pay the caller); * `MAX_SUNRISE_GAS` is the manipulation protection parameter that defines the maximum possible gas cost to call `sunrise` (likely during a Flood); * `PRIORITY_FEE_BUFFER` is the overestimation of the maximum required `priorityFee` to call `sunrise` (it is possible to access the `baseGasFee` but not the `priorityFee` in Solidity); * `MIN_REWARD` is the minimum Bean reward to payout; * `beanEtherPrice` is the cost of the call in Beans (which is derived from the BEAN:3CRV pool on Curve and the USDC:ETH pool on Uniswap V3); * `BASE_REWARD` is a fixed increase in Bean reward to cover the costs of operating `sunrise` bots; * `MAX_REWARD` is the max base reward to prevent manipulation (still 25 Beans) * `blocksLate` is the number of blocks in which a successful `sunrise()` call was executed past the first block in which it could have been executed; and * `SECONDS_PER_BLOCK` is number of seconds per block in Ethereum (post-Merge this is fixed at 12).

gm function

We propose adding a `gm` function that (1) calls the `sunrise` function, (2) can send the incentive reward to the caller's Farm or Circulating balance and (3) returns the number of Beans paid for the incentive reward.

Technical Rationale

The cost to execute `sunrise` in Beans is:

```
gasUsed * (baseGasFee + priorityFee) * beanEthPrice
```

This solution computes an on-chain estimation of the above formula while adding sufficiently tunable parameters to properly account for estimation error and potential manipulation.

Adding the ability to send `sunrise` incentive rewards to the caller's Farm balance increases the composability of Beanstalk. Adding a return value for the new `gm` function makes it much easier for other smart contracts calling Beanstalk to track `sunrise` incentive rewards.

Economic Rationale

Reducing unnecessary Pod issuance should improve Beanstalk's creditworthiness as a borrower. Maximizing Beans borrowed for a given Pod issuance should improve Beanstalk's efficiency.

Reducing unnecessary Bean issuance in the `sunrise` incentive reward can reduce sell pressure on Beans.

The cost of paying the `sunrise` caller differs based on the selected `ToMode` and whether the caller already has Beans in their Circulating and/or Farm balance. `GAS_OVERHEAD` is set such that the incentive amount is high enough for Farmers that don't have Beans in their balances already. For Farmers that already have Beans (outside the Silo), this also has the affect of compensating for one failed transaction per success, such that a caller with 50/50 success rate will retain slight profit.

Improving the composability of Beanstalk should improve the user experience and utility of Beanstalk and Beans.

Contract Changes

Field Facet

The following `FieldFacet` is being removed from Beanstalk: * [0x79801f5cb2592dd2173482198385e62870a0eae2](#)

The following `FieldFacet` is being added to Beanstalk: * [UPDATE](#)

FieldFacet Function Changes

Name	Selector	Action	Type	New Functionality
harvest	0x8fd83ecf	Replace	Call	

sow Name	0x6c8d548e Selector	Replace Action	Call Type	✓ New Functionality
sowWithMin	0x78309c85	Replace	Call	✓
harvestableIndex	0xd6be1816	Replace	View	
plot	0xe1d9d628	Replace	View	
podIndex	0xcb44a6cf	Replace	View	
totalHarvestable	0x067fcd2e	Replace	View	
totalHarvested	0x23dc1142	Replace	View	
totalPods	0xc0aa6a90	Replace	View	
totalSoil	0x300dd6cf	Replace	View	✓
totalUnharvestable	0x4433366d	Replace	View	

FieldFacet Event Changes

None.

Fundraiser Facet

The following FundraiserFacet is being removed from Beanstalk: * 0x538c76976ef45b8ca5c12662a86034434bfc7a8e

The following FundraiserFacet is being added to Beanstalk: * UPDATE

FundraiserFacet Function Changes

Name	Selector	Action	Type	New Functionality
createFundraiser	0x4b4e8d9a	Replace	Call	
fund	0x43c5198e	Replace	Call	✓
fundingToken	0xc869c1eb	Replace	View	
fundraiser	0xce133450	Replace	View	
numberOfFundraisers	0x6299a9af	Replace	View	
remainingFunding	0x0d1a844c	Replace	View	
totalFunding	0x6ee66ddf	Replace	View	

FundraiserFacet Event Changes

None.

Season Facet

The following SeasonFacet is being removed from Beanstalk: * [0x83d6e6b446613c9bfaebc64260962bc4f828a3ac](#)

The following SeasonFacet is being added to Beanstalk: * [UPDATE](#)

SeasonFacet Function Changes

Function Name	Selector	Action	Type	New Functionality
sunrise	0xfc06d2a6	Replace	Call	✓
gm	UPDATE	Add	Call	✓
abovePeg	UPDATE	Add	View	✓
sunriseBlock	UPDATE	Add	View	✓
paused	0x5c975abb	Replace	View	
plentyPerRoot	0xe60d7a83	Replace	View	
poolDeltaB	0x471bcdbe	Replace	View	
rain	0x43def26e	Replace	View	
season	0xc50b0fb0	Replace	View	
seasonTime	0xca7b7d7b	Replace	View	
time	0x16ada547	Replace	View	
totalDeltaB	0x06c499d8	Replace	View	
weather	0x686b6159	Replace	View	
yield	0x28593984	Replace	View	

SeasonFacet Event Changes

None.

Beans Minted

None.

Audit

The commit hash of this BIP is [\[UPDATE\]](#).

Halborn has performed an audit of this commit hash. You can view the Halborn audit report of this commit hash on Arweave here: * [\[UPDATE\]](#)

Effective

Effective immediately upon commit.