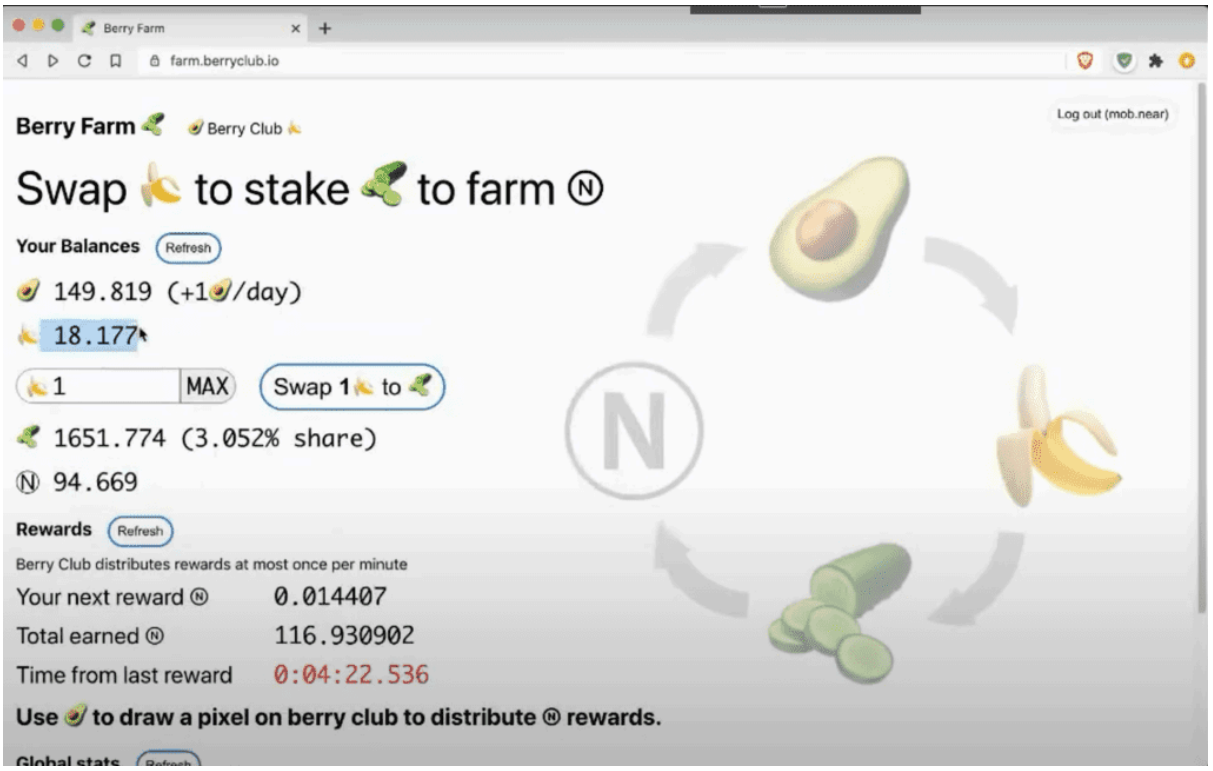
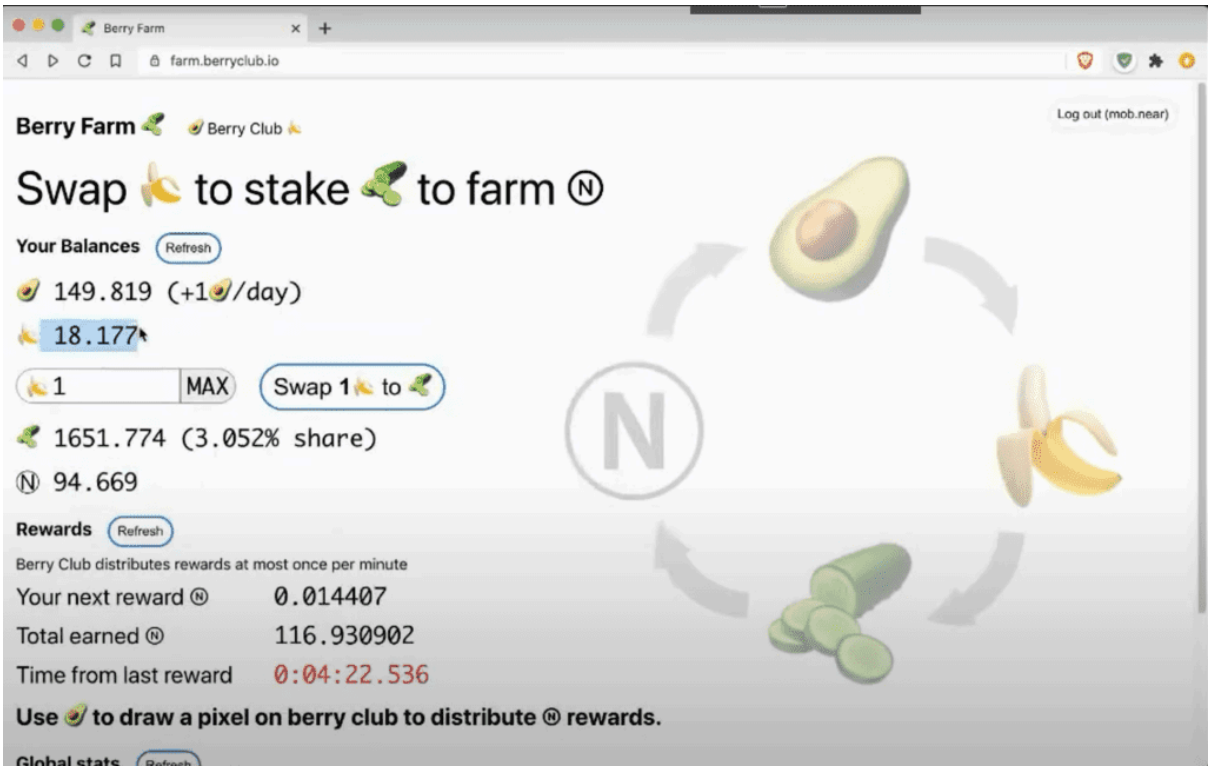
Kiểm tra contract trên Near | Phần 4: Berry Farm



Giới thiệu

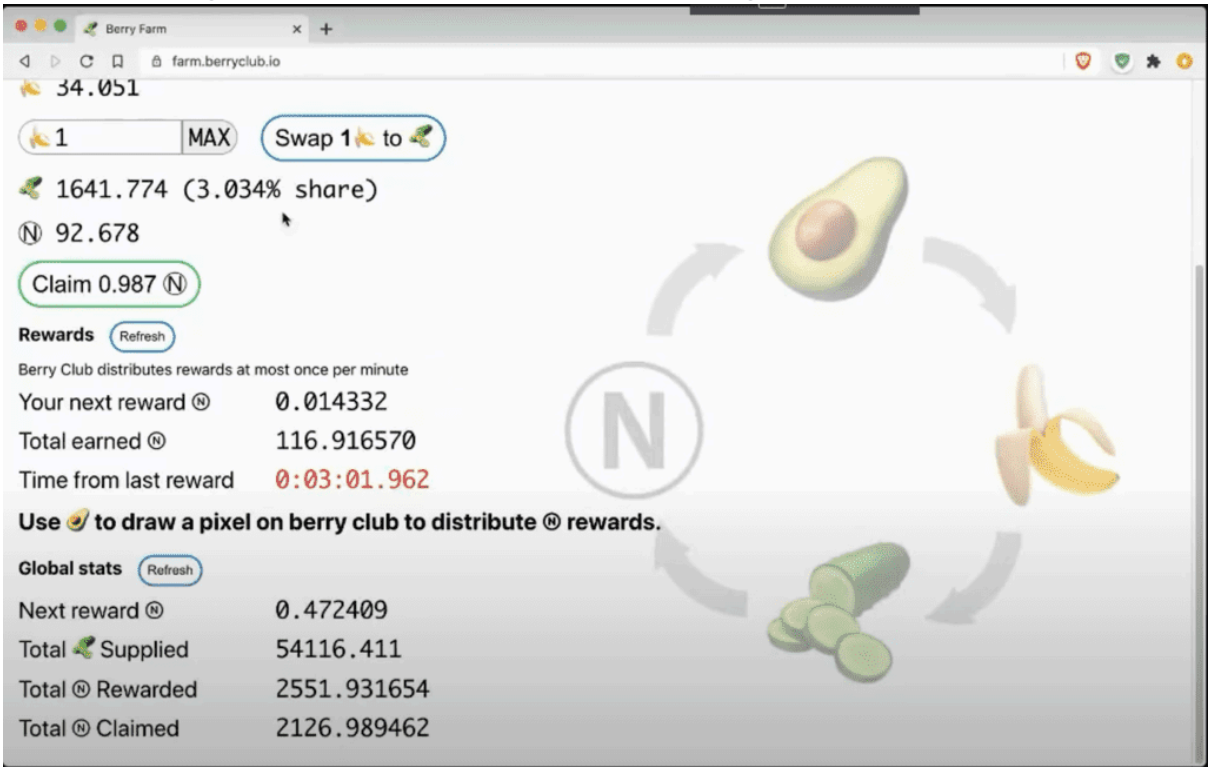
Hôm nay chúng ta sẽ nói về Berry Farm contract, là contract bổ sung của Berry Club cho phép bạn đổi Banana sang Cucumber và Near Farming. Đầu tiên chúng ta hãy đánh giá nhanh về Berry Farm. Contract có sẵn trên kho Github của tôi.

Giao diện người dùng

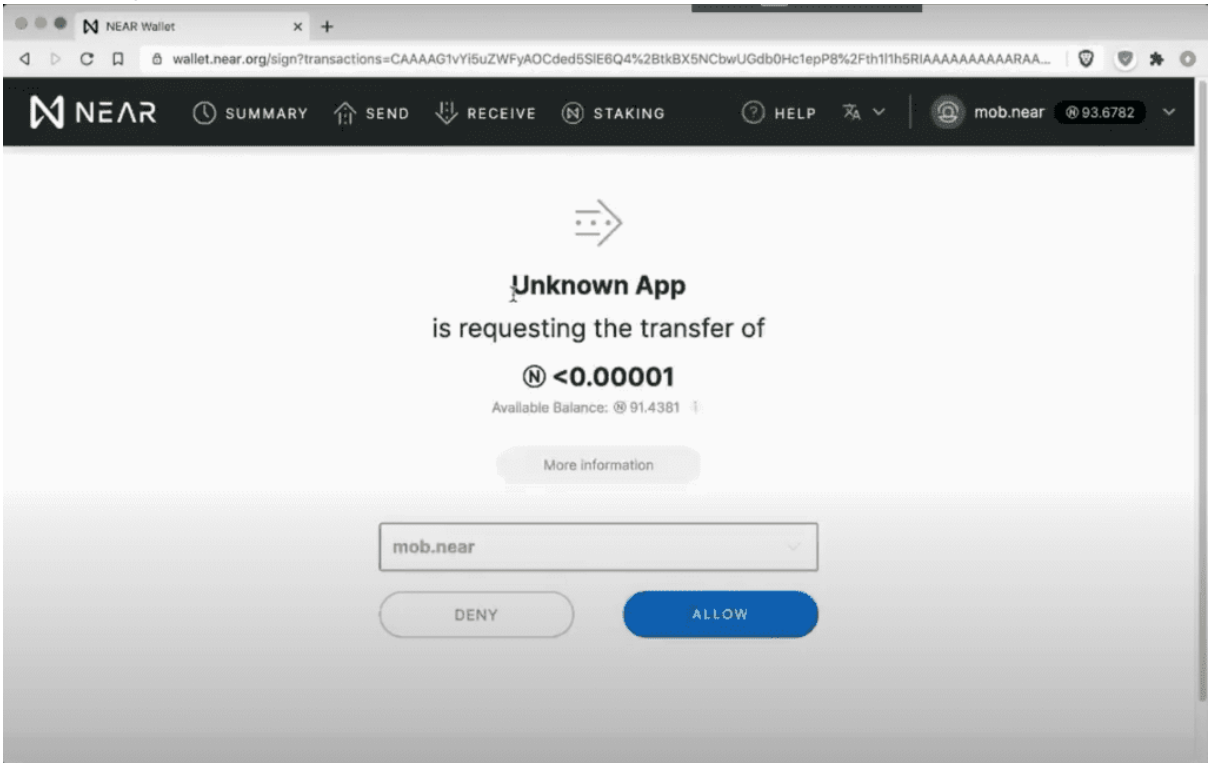


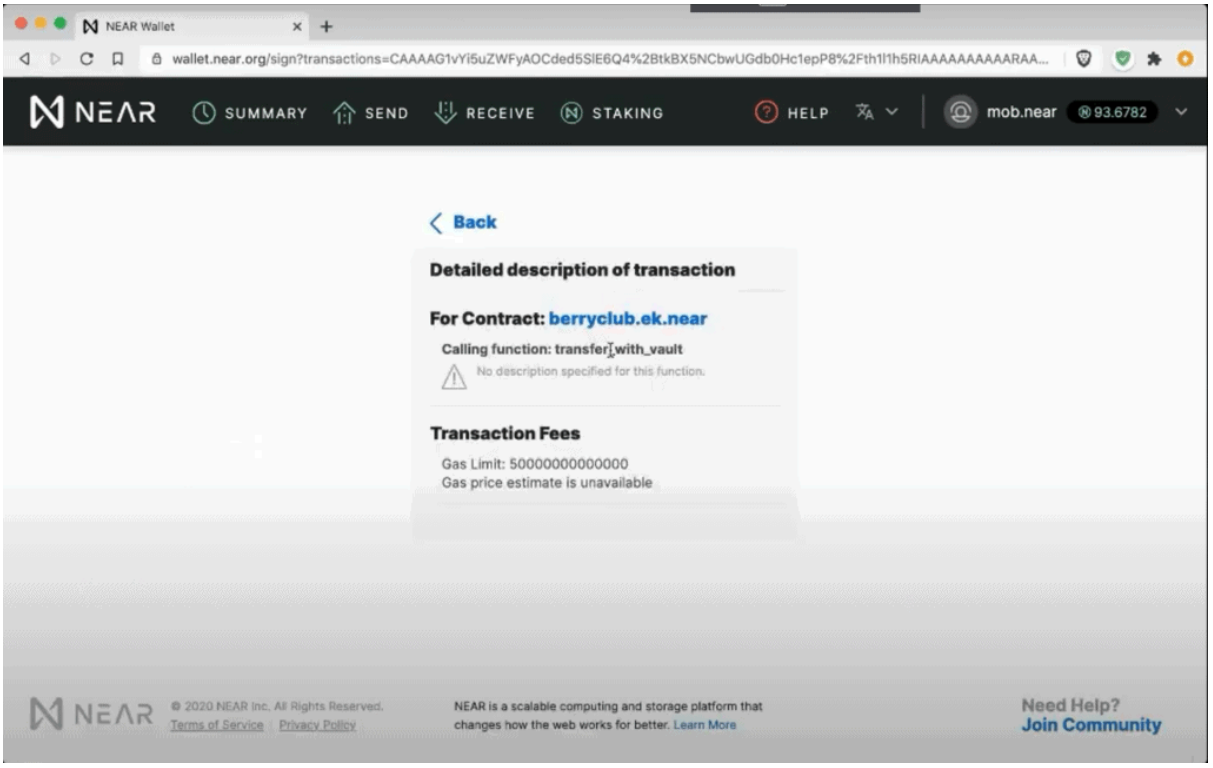
Berry Farm là một ứng dụng được thiết kế với rất nhiều biểu tượng cảm xúc. Nó lấy số dư từ hai hợp đồng khác nhau cũng như tài khoản của

bạn. Avocado và Banana đến từ hợp đồng Berry Club, Cucumber đến từ hợp đồng Berry Farm và Near đến từ số dư tài khoản của bạn. Nếu bạn farm thì bạn sẽ nhận được phần thưởng khi phần thưởng được phân phối. Có vẻ như Near đây ai đó đã mua rất nhiều Avocado và phần thưởng tăng lên. Nó cũng có số liệu thống kê, có nghĩa là 2500 NEAR đã được chuyển từ Berry Club sang Berry Farm. Tôi đã kiếm được 116,9 Near và phần thưởng tiếp theo của tôi sẽ là 0,014332 Near dựa trên số lượng Cucumber mà tôi có so với tổng số Cucumber.

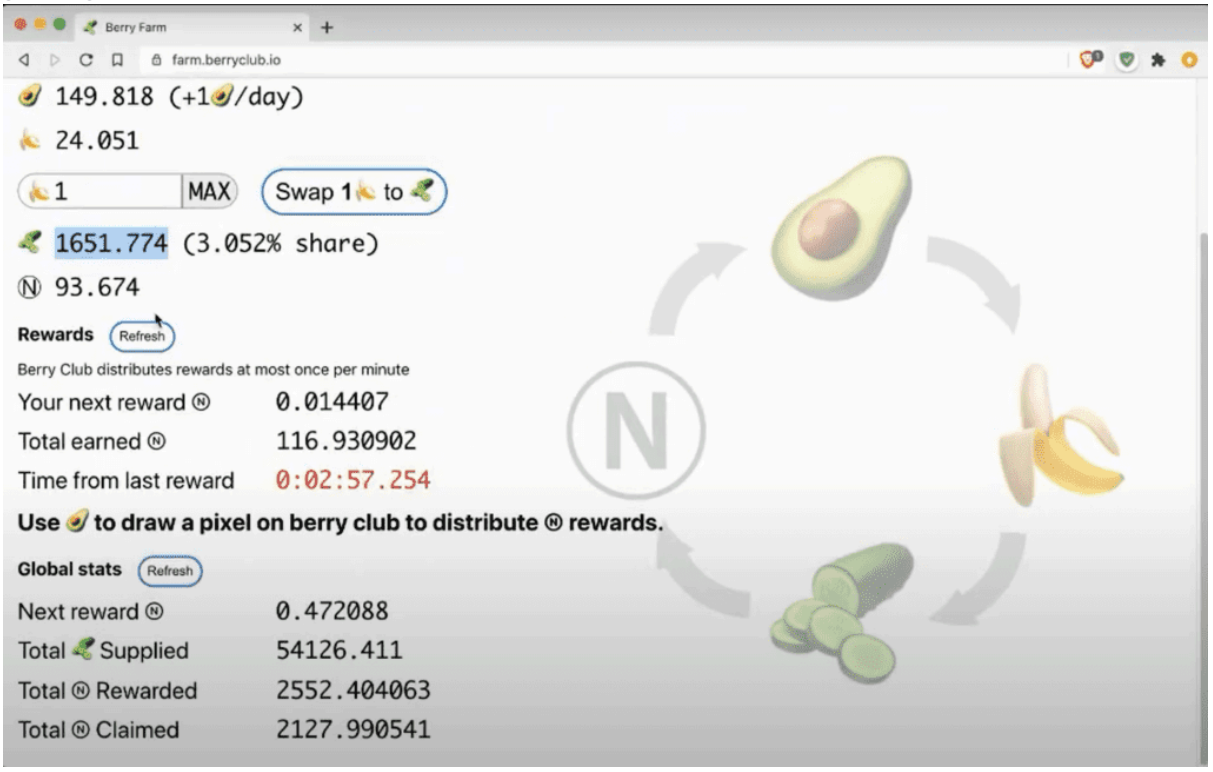


Tổng cung của Cucumber hiện tại là 54116,4 quả. Tôi nhận được 3% chia sẻ phần thưởng mỗi khi có phần thưởng mới. Vì vậy, khi tôi nhấp vào refresh thì số liệu thống kê từ nhiều tài khoản và tôi có thể yêu cầu Near từ hợp đồng sang tài khoản của tôi. Tôi cũng có thể hoán đổi Banana sang Cucumber và tôi đang thực hiện một cuộc gọi ký hợp đồng với Berry Club.





Tôi sẽ gọi transfer_with_vault đang sử dụng tiêu chuẩn mới và nó yêu cầu một lượng Near nhỏ và tôi nghĩ rằng nó giống như 1 yocto Near để xác nhận rằng tôi không thực hiện việc này thông qua khóa truy cập. Tôi không thể thực hiện chuyển khoản thông qua khóa truy cập trong order để tránh việc rút tiền tự động và đây chỉ là biện pháp bảo vệ an toàn. Ví dụ: Nếu bạn ủy quyền Banana từ một số ứng dụng khác đang cố gắng hút hết Banana của bạn, nó sẽ không thành công, vì nó sẽ không thể gọi transfer_raw hoặc transfer_with_vault. Để làm rõ, transfer_raw là tên mới để chuyển mà không cần hợp đồng, vì vậy bạn có thể chỉ cần gửi tiền bằng tài khoản của ai đó tồn tại và trong transfer_with_vault đang gọi người nhận để rút token.



Tôi đã dành 10 Banana để có thêm Cucumber và tôi có thể refresh số dư một lần nữa để xem liệu ai đó đã rút thứ gì đó hay không. Đó là cách giao diện người dùng hoạt động, nhưng có nhiều chức năng hơn là chỉ

thế này. Hợp đồng Cucumber như một token có thể thay thế được, cũng như Banana.

Contract Review

Berry Club

Chúng ta sẽ bắt đầu với Banana Farm contract, nhưng bạn có thể sẽ cần xem xét token contract từ pixelboard, một hợp đồng từ Berry Club, cũng như hợp đồng gửi token.

<script src="https://gist.github.com/FHLC-FUND/9d073517ecb703edb47cfde73b5cb4fb.js"></script>

Berry Club có một hard-coded contract được gọi là id hợp đồng farming.

<script src="https://gist.github.com/FHLC-FUND/75f6e2bd6116dbea39cc8646c00abd45.js"></script>

Khi bạn vẽ, nó xác minh rằng bạn không vẽ các pixel trống. Nếu không, bạn có thể kích hoạt phần thưởng mà không cần tốn Avocado.

<script src="https://gist.github.com/FHLC-FUND/b5a25278add27a34081535f23bf64c6f.js"></script>

Sau đó, nó gọi maybe_send_reward để lấy thời gian hiện tại từ blockchain và kiểm tra xem next_reward_timestamp có nhỏ hơn thời gian hiện tại hay không.

<script src="https://gist.github.com/FHLC-FUND/dfef7075d34a7a789157315db48057a0.js"></script>

Phần thưởng tiếp theo là thời gian bắt đầu farm, là thời gian ban đầu cho phép đếm ngược 26 giờ hoặc dấu last_reward_timestamp khi chúng được kích hoạt lần cuối vào 60 giây.

<script src="https://gist.github.com/FHLC-FUND/b5a25278add27a34081535f23bf64c6f.js"></script>

Vì vậy, nếu đến lúc gửi phần thưởng tiếp theo, nó sẽ ghi nhớ thời gian hiện tại và tính toán phần thưởng dựa trên logic để lấy số dư hiện tại

của Berry Club, tính toán storage theo chi phí lưu trữ cộng với một ít Near dự trữ.

<script src="https://gist.github.com/FHLC-FUND/b17c676c8a2bd284213d97ce59f6beba.js"></script>

Nó luôn giữ 15 Near cho contract trong tương lai. Sau đó, nếu số dư có sẵn cao hơn số dư chúng tôi cần để lưu trữ cộng với độ an toàn, thì chúng tôi tính toán số dư thừa và chia nó cho phần thưởng. Trước hết, bạn không thể tạo ra một khối được đánh dấu thời gian trước đó, và sau đó bạn không thể tạo khối ngoài vị trí của mình, bạn có một vị trí chuyên dụng khi bạn cần tạo một đoạn. Vì vậy, bạn có thể làm nhiều nhất là tạo ra một đoạn của khối cuối cùng ở phần nghìn giây cuối cùng của vị trí của bạn, tức là trước một giây. Bạn thực sự không thể di chuyển nó về phía trước nếu không tất cả các nút khác cũng sẽ phá vỡ giao thức, vì vậy chúng bị giới hạn bởi sự hiểu biết của chúng về thời gian cộng với thanh mà chúng sử dụng để quyết định có chấp nhận một khối hay không, vì vậy bạn không thể thực sự làm cho thời gian chạy nhanh hơn trừ khi tất cả trình xác thực muốn làm điều này. Điều này có nghĩa là chúng tôi có trình xác thực về cơ bản sẽ ngừng tuân theo giao thức. Họ cần điều phối thời gian của họ.

<script src="https://gist.github.com/FHLC-FUND/b5a25278add27a34081535f23bf64c6f.js"></script>

Vì vậy, cuối cùng nó đã nhận được phần thưởng, nó nói rằng tôi sẽ phân phối phần thưởng này và những gì nó làm là nó gọi là farm contract và gọi take_my_near và nó chỉ không vượt qua bất kỳ đối số nào và chuyển phần thưởng và gas của nó. Chỉ cần một lượng khí tối thiểu cần thiết để thêm số dư. Đó là cách Berry Club được cập nhật để bắt đầu phân phối phần thưởng sau mỗi lần rút thăm. Nó mất nhiều hơn một chút. Nó sử dụng gas đã được sử dụng để vẽ để thực sự phân phối phần thưởng đôi khi cần khoảng 10 tera gas cho việc này.

Berry Farm

Hãy đến Berry Farm và bắt đầu với hợp đồng này. Cách thức hoạt động của trang trại quả mọng mà nó cần, trong thời gian liên tục, có thể phân phối phần thưởng dựa trên tổng số Cucumber, đây không phải là một nhiệm vụ tầm thường, nó hơi giống với nhóm đặt cược và cách nhóm đặt cược phân phối phần thưởng. Nhóm đặt cược tạo ra một loạt cổ phiếu cho mỗi khoản tiền gửi bạn thực hiện và cổ phiếu có giá toàn cầu hiện tại nói rằng, "Này, tôi đã mua hoặc đúc thêm cổ phiếu và bây giờ cổ

phiếu của tôi có thể được đổi thành một số lượng Near nhất định."Giá cổ phiếu bắt đầu từ 1 năm Near trên mỗi cổ phiếu, và sau đó bạn càng tích lũy được nhiều phần thưởng thì giá cổ phiếu càng tăng. Khi bạn sử dụng chức năng hủy bỏ lệnh, bạn bán cổ phiếu ở mức giá hiện tại và bạn nhận lại Near, và cổ phiếu được gỡ bỏ, và giá cổ phiếu giảm xuống. Thách thức ở đó là khi bạn đặt cược Cucumber, giá của một quả Cucumber bằng 0, và khi bạn đặt cược lần đầu tiên, bạn không thể thực sự mua cổ phiếu và cổ phiếu đúc, bởi vì giá của chúng bằng không. Logic của các lượt chia sẻ không hoạt động tốt cho điều này vì vậy đối với điều này, tôi cần tạo ra một cái gì đó khác nhau để thực hiện kế toán. Hãy đi vào ví dụ. Giả sử Mike là người đầu tiên đặt cọc 10 quả Cucumber vào hợp đồng, và sau đó giả sử 100 Near đã được phân phối. Mike sẽ nhận tất cả phần thưởng vì anh ấy là người duy nhất, anh ấy kiểm soát 100% cổ phần ở Berry Farm lúc này. Ví dụ: Mike đang ngoại tuyến và không xác nhận quyền sở hữu của họ, vì vậy chúng tôi không thể thực sự cập nhật tài khoản của anh ấy, vì nó phải được thực hiện liên tục. Chúng ta không thể lặp lại tất cả các tài khoản, nhưng bằng cách nào đó chúng ta cần nhớ rằng Mike đã nhận được tất cả Near cho Cucumber của mình. Bây giờ, giả sử Josh đến và đặt thêm 10 quả Cucumber nữa, bây giờ phần của Mike là 50% và phần của Josh là 50%, nhưng Josh đã đặt cược số Cucumber sau khi 100 Near ban đầu này đã được phân phối. Anh ta không được hưởng bất kỳ điều gì trong số này bây giờ 100 Near giảm nữa Mike bây giờ cần nhận thêm 50 lần nữa, bởi vì anh ta có 50% cổ phần và Josh nhận được 50 Near. Khi Mike trực tuyến trở lại, chúng tôi cần có thể hiển thị số dư của anh ấy ở mức 150 Near mà anh ấy kiếm được trong thời gian không đổi. Vì vậy, đây là thách thức mà chúng tôi cần giải quyết với hợp đồng này và cách nó được giải quyết là đặt tỷ lệ toàn cầu được gọi là Near trên mỗi quả Cucumber, cho biết số Near nhận được bằng số dư Cucumber nhất định khi một tài khoản được tạo, hoặc xúc động theo bất kỳ cách nào. Chúng tôi nhớ giá trị cuối cùng của phân số này. và cách nó được giải quyết là bằng cách có một tỷ lệ toàn cầu được gọi là Near trên mỗi quả Cucumber, cho biết số Near nhận được bằng số dư Cucumber nhất định khi một tài khoản được tạo hoặc chạm vào theo bất kỳ cách nào. Chúng tôi nhớ giá trị cuối cùng của phân số này. và cách nó được giải quyết là bằng cách có một tỷ lệ toàn cầu được gọi là Near trên mỗi quả Cucumber, cho biết số Near nhận được bằng số dư Cucumber nhất định khi một tài khoản được tạo hoặc chạm vào theo bất kỳ cách nào. Chúng tôi nhớ giá trị cuối cùng của phân số này.

<script src="https://gist.github.com/FHLC-FUND/57d7670405de93289946e066be84135c.js"></script>

Khi đó, chúng ta có số dư Near, vì vậy nếu phân số cuối cùng thấp hơn phân số hiện tại, điều đó có nghĩa là chúng ta có thể nhân số Cucumber

của tài khoản đã cho với hiệu số giữa 2 phân số và tính xem tôi được hưởng bao nhiêu Near. Khi tài khoản của mike được tạo, phân số bằng 0 chia cho một và khi 100 Near được gửi vào với tổng số Cucumber là 10, phân số này trở thành bằng 10. Sau đó, một khi Josh tham gia và thả thêm 10 quả Cucumber nữa cho tổng số quả Cucumber và 100 Near khác đã được phân phối, phần này đã tăng lên 5 vì có tổng cộng 20 quả Cucumber và 100 quả Near. Vì vậy 100 Near chia cho 20 thì có nghĩa là 5 Near trên mỗi quả Cucumber nên giá trị này trở thành 15. Bây giờ khi Mike quay lại, anh ấy nhìn vào số dư Cucumber của mình, Near 10 cho mỗi quả Cucumber trên hợp đồng toàn cầu là 15 giá trị được nhớ cuối cùng của anh ấy là 0 vì vậy anh ấy chỉ cần lấy chênh lệch là 15 nhân với 10 và yêu cầu số dư này thành Near, và điều này 1 được đặt lại thành 15. Khi Josh quay lại, anh ta nhận được giá trị này là 10, anh ta có 10 quả Cucumber và giá trị hiện tại là 15. Hiệu số là 5 nhân với số dư Cucumber là 10 nên anh ta nhận được 50 Near. Đó là cách chúng tôi quản lý số dư Near và lấy phần thưởng nhận được từ những người sử dụng thời gian không đổi để tính toán lại. Vì vậy, thay vì giữ toàn bộ một phân số trong bộ nhớ, chúng tôi sử dụng mẫu số toàn cục được cố định và tôi sử dụng nó thành 10 thành lũy thừa của 18, bằng với độ chính xác của Cucumber, Banana và Avocado. Đó là nền tảng của cách tất cả được tính toán.

<script src="https://gist.github.com/FHLC-FUND/f5bdff6f3e3b071ca4119c69bbfcc6a3.js"></script>

Chúng ta hãy nhìn vào cấu trúc chính của hợp đồng, một hợp đồng được gọi là trang trại. Nó có một bản đồ tra cứu các tài khoản và nó sử dụng các hàm băm ngắn tương tự như các mã thông báo có thể thay thế được với kho tiền. Nó có id tài khoản Banana token cũng có thể được mã hóa cứng, nhưng nó đã được sử dụng từ hợp đồng vì chúng tôi thực sự có thể vượt qua nó, vì tài khoản này đã được triển khai sau đó và tôi không muốn nâng cấp trạng thái hợp đồng Berry Club. Tôi không muốn sửa đổi trạng thái một lần nữa thông qua khả năng nâng cấp nên thay vào đó, tôi đã mã hóa cứng giá trị trong hợp đồng trước đó. Near mỗi quả Cucumber, phân số mà tôi đã giải thích trước đó, là tử số và tổng số dư Cucumber cần thiết làm mẫu số. Vault và phiếu bầu tiếp theo được sử dụng để chuyển giao; họ thực sự có thể không bao giờ quen với việc chuyển Cucumber ngoại trừ có thể một lần để kiểm tra.

<script src="https://gist.github.com/FHLC-FUND/3ddabcce95bce6d43f54f76e51326cee.js"></script>

Đây là những gì chúng ta đã thấy trước đây. Đây một lần nữa là mã băm của id tài khoản.

<script src="https://gist.github.com/FHLC-FUND/57d7670405de93289946e066be84135c.js"></script>tử số

Cấu trúc tài khoản là giá trị cuối cùng trước khi tài khoản được chạm vào phần này. near_balance là số dư không có người nhận. Bất cứ khi nào bạn chạm vào tài khoản bằng cách gửi thêm Cucumber, hoặc yêu cầu Near, nó sẽ cập nhật giá trị cuối cùng và chuyển tất cả Near không có người nhận sang số dư cục bộ của bạn ra khỏi sự chênh lệch và số dư Cucumber là số dư của bạn trong Cucumber. near_claimed là số liệu thống kê, chỉ cho biết bạn đã nhận được bao nhiêu phần thưởng. Nó không cần thiết cho logic hợp đồng.

<script src="https://gist.github.com/FHLC-FUND/5ca8cde13672c1c725d6ce7699897938.js"></script>

Có hai cấu trúc trợ giúp chỉ dành cho giao diện người dùng. HumanAccount trả về các giá trị theo cách dễ đọc hơn và HumanStats là thống kê toàn cầu về mức tiêu thụ. Hãy xem hàm take_my_near để thực sự xem nó hoạt động như thế nào.

<script src="https://gist.github.com/FHLC-FUND/71554b940d337241fae32b3b8fdd6155.js"></script>

Điều đầu tiên chúng tôi làm là kiểm tra xem có đủ Cucumber ở đó không, bởi vì chúng tôi sử dụng nó như một mẫu số. Chúng tôi chỉ muốn 1 quả Cucumber thực sự kích hoạt điều này, và đây chỉ là biện pháp phòng ngừa. Bạn không thích kích hoạt loại công cụ này. Hãy cùng khám phá giá trị này. Điều gì xảy ra là bạn nhận được số lượng Near, và yocto Near, và bạn nhân nó với mẫu số chung. Vì vậy, bạn đang giải một phân số Near trên mỗi quả Cucumber chia cho mẫu số cộng với số dư kèm theo chia cho tổng số dư Cucumber. Đây là công thức: near_per_cucumber / denom + attachment_deposit / total_cucumber_balance. Vì vậy, new_near_per_cucumber thực sự là có đính kèm và mẫu số sau đó chia cho mẫu số. Đây là công thức: new_near_per_cucumber = (near_per-cucumber_numer +denom + attached deposit) / denom. Đây là công thức mà chúng tôi sử dụng để tính số_tố_tạo_mới. Bất cứ khi nào bạn chạm vào tài khoản, nó luôn biến số dư chưa nhận hiện tại của bạn thành một số cố định và sau đó bạn có thể sửa đổi bất kỳ giá trị nào mà không vi phạm logic. Phương

thức cảm ứng này được gọi theo mọi phương thức getter và mọi thay đổi. Trước khi bạn sửa đổi tài khoản, nó luôn gọi là chạm vào tài khoản.

<script src="https://gist.github.com/FHLC-FUND/7f17646998de5bfc2a58e14cfc520fff.js"></script>

Ví dụ, trong get_near_balance, chúng tôi nhận được tài khoản nội bộ. Nếu tài khoản tồn tại thì chúng tôi chạm vào nó cục bộ. Sau đó, chúng ta sẽ nhận được near_balance

<script src="https://gist.github.com/FHLC-FUND/aefb980b991ffe8b00d67c175384cb06.js"></script>

Nếu bạn nhận được tài khoản mà chúng tôi chạm vào nó trước đó thì bạn sẽ nhận được near_balance, cucumber_balance và yêu cầu số dư của bạn. Đó là những chức năng xem đang thay đổi mọi thứ, nhưng nó không lưu nó nên nó chỉ tạm thời thay đổi giá trị tài khoản và không lưu lại. Nó chỉ tạo ra một liên lạc đó là một loại tính toán lại với giá trị mới nhất.

<script src="https://gist.github.com/FHLC-FUND/4ef64071e23229df2cd38824cb7e5fad.js"></script>

Nó sẽ lưu nó khi bạn thực sự thực hiện bất kỳ lệnh gọi thay đổi nào, vì vậy, ví dụ như lệnh gọi đến register_near. Chúng tôi nhận được tài khoản của bạn và get_mut_account thực sự chạm vào tài khoản. Bạn nhận được tài khoản đã được cập nhật và near_balance là số dư thực tế. Nó đưa near_balance về 0, có nghĩa là bạn đã xác nhận tất cả điều này và nó sẽ lưu tài khoản của bạn. Sau đó, nếu bạn yêu cầu giá trị dương thì chúng tôi sẽ chuyển cho bạn từ tài khoản của bạn. Nó trả về số Near bạn đã yêu cầu. Cái hay của phương pháp cảm ứng này là nó có thể được gọi bất cứ lúc nào. Nó không nhất thiết phải được gọi ở điểm đã cho, vì vậy nó sẽ luôn cho bạn cùng một kết quả bất kể bạn đã gọi nó hai lần ở giữa hay đã gọi nó một lần. Điều này chỉ để trả lại cho người dùng số tiền họ có. Nó chỉ cho bạn biết bạn đã kiếm được bao nhiêu Near bằng cách tính toán từ trạng thái hiện tại. Trong Câu lạc bộ Berry, chúng tôi cũng làm những thứ tương tự mà chúng tôi có thể chạm vào, và điều này là cần thiết, bởi vì giả sử bạn muốn biết bạn đã trồng bao nhiêu Banana. Vì chúng tôi biết lần cuối cùng bạn chạm vào tài khoản của mình, lần cuối cùng bạn vẽ thứ gì đó, chúng tôi biết bạn có bao nhiêu pixel. Chúng tôi có thể tính toán từ một thời điểm nhất định đến thời điểm trước đó nhân với số pixel bạn có bao nhiêu Banana mà

bạn nên có ngay bây giờ và giao diện người dùng cũng làm điều tương tự. Giao diện người dùng mô phỏng nó, nó nhận được cùng một thông tin ngoại trừ thông tin mới nhất và sau đó nó chỉ tạo ra một bộ đếm thời gian về sau để tính toán nó.Về lý thuyết, phương thức get_near_balance không phải trả về giá trị Near của bạn đã cập nhật, chỉ có thể nói đây là tổng số Near trên mỗi quả Cucumber và tổng số Near cuối cùng của bạn trên mỗi quả Cucumber và cũng là số dư Near cuối cùng của bạn, nhưng thay vào đó, nó được thực hiện trên mức hợp đồng. Giả sử chúng tôi không có các phương thức xem mà bạn không phải trả tiền và chỉ trả lại và truy xuất một trạng thái, sau đó bạn sẽ phải thực hiện logic này trên giao diện người dùng thay vì thực hiện nó ở đây trong bộ nhớ.

Token.rs

Logic cuối cùng mà chúng ta cần làm là cách bạn nhận Cucumber từ Berry Club và điều này được thực hiện bằng cách sử dụng transfer_with_vault từ Berry Club. Trước đây chúng ta đã thảo luận về cách map hoạt động.

<script src="https://gist.github.com/FHLC-FUND/ced5c73e5446e8d86dcaa2453aa4a643.js"></script>

Bạn chuyển receiver_id, số tiền và trọng tải. Nó có một kiểm tra khẳng định_ trả tiền mà chỉ kiểm tra rằng khoản tiền gửi là số dương. Nó nói rằng oh bạn cần ít nhất 1 yocto Near để ngăn chặn các lệnh gọi phím truy cập chức năng để tránh giao diện người dùng Berry Club độc hại hút hết Banana của bạn. Nó nhận được khí và sau đó bắt đầu gọi Banana vào một cái hầm. Nó cũng bắt đầu một cuộc gọi đến người nhận. Trong trường hợp này, chúng tôi gọi nó là hợp đồng nông trại và nó chuyển cho id người gửi một lượng Banana, id vault chỉ là u64 không có JSON và trọng tải là một Banana. Trong ví dụ trước, khi chúng ta nói về map 122, payload là một bản nhị phân của vec q8 trong base64, vì nó là một payload được tuần tự hóa borsh. Trong cuộc thảo luận, tôi lưu ý rằng nếu bạn xem lại đối số này trong ví, bạn sẽ không thể thấy khối lượng mà bạn muốn thực hiện, Đó là một enum với một tùy chọn duy nhất được gọi là ký quỹ_and_stake.

<script src="https://gist.github.com/FHLC-FUND/bdf86b9123f01105715a0b71f39a76b5.js"></script>

Bây giờ khi bạn thực hiện một lệnh gọi hàm từ giao diện người dùng trang trại sẽ chuyển đến ví trong các đối số, trọng tải sẽ giống như một Banana liền kề trong một Banana nói rằng tiền gửi và tiền đặt cọc thay

vì được mã hóa bằng 0 làm cơ sở, thay vì là một borsh phiên bản được đăng nhiều kỳ, đó là phiên bản con người có thể đọc được. Hợp đồng này có các loại trọng tải khác với Uniswap, chẳng hạn, nó sẽ có trọng tải cho biết bạn muốn nhận tối đa bao nhiêu Near hoặc nhận ít nhất. Giao diện người dùng Uniswap độc hại thực sự có thể thay thế giá trị này và nếu bạn không xác minh nó ở phía ví thì bạn có thể thực sự bị loại bỏ theo xu hướng phía trước. Chúng tôi đã tạo tài khoản bằng cách sử dụng get_mut_account, đây là tài khoản hiện có hoặc tạo một tài khoản mới trong đó tài khoản này đặt tất cả số dư thành 0, nhưng tử số Near nhất trên mỗi quả Cucumber là số toàn cầu hiện tại. Điều này có nghĩa là bạn không trồng trọt gì và bạn không có số dư Cucumber. Nó vẫn chạm vào dù không cần thiết, chúng tôi có thể tránh nó nếu chúng tôi lập bản đồ, và sau đó nó trả về băm và tài khoản. Tài khoản đã được cập nhật.Những gì chúng tôi làm là đầu tiên chúng tôi tăng tổng số dư Cucumber và sau đó chúng tôi lưu nó trở lại trạng thái của tài khoản và chúng tôi cũng tăng số dư Cucumber toàn cầu bằng số lượng Cucumber đã gửi. Sau đó, chúng tôi không thực sự xác minh rằng hầm chứa Cucumber, vì vậy chúng tôi tin tưởng rằng người gọi trước đó, đó là một hợp đồng Banana, thực sự đã đưa Banana vào hầm và chúng tôi trả lại lời hứa từ phương pháp này. Đó là một kết hợp kỳ lạ với một mục nhập duy nhất bởi vì tải trọng có thể chỉ là một loại ngay bây giờ. Chúng tôi gọi là rút_from_vault, chúng tôi cho biết số tiền chúng tôi muốn rút là toàn bộ số tiền và sau đó chúng tôi gọi nó trở lại id tài khoản mã thông báo Banana là hợp đồng Berry Club. Đã vậy hợp đồng này còn chưa nhận Banana, cũng không quan tâm vì dù sao cũng không dùng Banana. Nó không cần Banana trước khi nó có thể tiếp tục,đã bị khóa. Nó làm tăng tổng số dư Cucumber bằng cách pha loãng cổ phần. Bây giờ khi take_my_near đến, nó sẽ sử dụng tổng số dư Cucumber mới. Bây giờ tôi sẽ chia số lượng cổ phiếu cho tất cả những người tham gia.

Kết luận

Đó là tổng quan về Berry Club, Berry Farm và các hợp đồng token.rs. Cảm ơn bạn đã đọc và chúc bạn may mắn trong những nỗ lực trong tương lai với Near.