Τερζίδου Νικολέττα f3312412

Παναγιώτης Μελάς Φραγκίσκος f3312407

README - CryptoSOS & MultiSOS

Περιγραφή

Το έργο μας περιλαμβάνει δύο εκδόσεις του παιχνιδιού **SOS** στην πλατφόρμα του Ethereum.

1. **CryptoSOS**: Η πρώτη έκδοση του παιχνιδιού, όπου δύο παίκτες μπορούν να παίξουν SOS σε μια κλασική 3x3 πλατφόρμα, τοποθετώντας τα σύμβολα "S" και "O". Το παιχνίδι τελειώνει όταν κάποιος από τους παίκτες δημιουργήσει τη λέξη "SOS" ή όταν το ταμπλό γεμίσει χωρίς νικητή (ισόπαλο αποτέλεσμα).
2. **MultiSOS**: Η δεύτερη έκδοση επεκτείνει το CryptoSOS επιτρέποντας την ταυτόχρονη ύπαρξη πολλών παιχνιδιών. Οι παίκτες μπορούν να συμμετάσχουν σε διαφορετικά παιχνίδια ταυτόχρονα, αλλά με τον περιορισμό ότι κάθε παίκτης μπορεί να συμμετέχει σε ένα μόνο παιχνίδι κάθε φορά.

Σχεδίαση και Λειτουργία

CryptoSOS

- Η λογική του CryptoSOS βασίζεται σε έναν πίνακα 3x3, όπου οι παίκτες τοποθετούν τα σύμβολα τους.
- Οι παίκτες τοποθετούν S ή O στον πίνακα.
- Οι παίκτες εναλλάσσονται και η σειρά είναι αυστηρά καθορισμένη.
- Όταν ο πίνακας γεμίσει ή αν κάποιος σχηματίσει τη λέξη "SOS", το παιχνίδι τελειώνει.
- Αν οι παίκτες δεν καταφέρουν να κερδίσουν, το παιχνίδι καταλήγει σε ισοπαλία.

MultiSOS

- Το MultiSOS είναι μια επέκταση του CryptoSOS. Στην εκδοχή αυτή, πολλοί παίκτες μπορούν να συμμετέχουν ταυτόχρονα σε διαφορετικά παιχνίδια.
- Κάθε παίκτης μπορεί να συμμετάσχει σε μόνο ένα παιχνίδι κάθε φορά.
- Το σύστημα παρακολουθεί ποιος παίζει σε ποιο παιχνίδι μέσω ενός **mapping** για την αποθήκευση των παιχνιδιών κάθε παίκτη.

Οικονομία Gas

Για να εξασφαλίσουμε ότι το σύστημα λειτουργεί αποτελεσματικά και χωρίς υπερβολικό κόστος **gas**, έχουμε λάβει διάφορα μέτρα:

- **Αντικατάσταση τύπων δεδομένων**: Χρησιμοποιούμε uint8[9] αντί για string[9] για την αποθήκευση του πίνακα. Αυτό μειώνει την κατανάλωση μνήμης και κατ' επέκταση το κόστος gas.
- **Αποτελεσματικοί έλεγχοι**: Χρησιμοποιούμε **require** για να επιβεβαιώσουμε έγκυρες θέσεις, έγκυρους παίκτες και χρονικά όρια. Αυτό διασφαλίζει ότι οι συναλλαγές εκτελούνται μόνο όταν πληρούνται οι προϋποθέσεις και αποτρέπει άχρηστες συναλλαγές.
- **Λιγότεροι υπολογισμοί**: Οι έλεγχοι για το αν κάποιος έχει κερδίσει ("SOS") γίνονται με την ελάχιστη δυνατή υπολογιστική πολυπλοκότητα. Για παράδειγμα, οι έλεγχοι της γραμμής (rows), της στήλης (columns) και της διαγωνίου (diagonals) πραγματοποιούνται με λίγες συγκρίσεις για να μειωθεί το κόστος υπολογισμού.

Δυσκολίες και Αντιμετώπιση

Κατά τη διάρκεια της ανάπτυξης συναντήσαμε μερικές δυσκολίες:

- **Διαχείριση πολλών παιχνιδιών**: Στο MultiSOS, έπρεπε να διασφαλίσουμε ότι οι παίκτες δεν θα μπορούσαν να συμμετέχουν σε περισσότερα από ένα παιχνίδια ταυτόχρονα. Το χρησιμοποιούμενο mapping playerGame εξασφαλίζει ότι κάθε παίκτης είναι μοναδικός σε κάθε παιχνίδι.
- **Επικύρωση κινήσεων**: Χρησιμοποιήσαμε αυστηρούς ελέγχους με την εντολή require για να ελέγξουμε την εγκυρότητα των κινήσεων, τη σειρά των παικτών και τα χρονικά όρια, ώστε να αποτρέψουμε λάθη και κακόβουλες ενέργειες.

Στρατηγικές Αντιμετώπισης Επιθέσεων

Για να διασφαλίσουμε την ασφάλεια του παιχνιδιού και να αποτρέψουμε κακόβουλες επιθέσεις, εφαρμόσαμε τα εξής μέτρα:

- **Περιορισμός συμμετοχής**: Κάθε παίκτης μπορεί να συμμετέχει σε μόνο ένα παιχνίδι κάθε φορά. Αυτό μειώνει τον κίνδυνο spamming με αποτέλεσμα να μην μπορούν να μπουν στο παιχνίδι οι παίκτες (τυπου denial of service).
- **Ασφάλεια συναλλαγών**: Χρησιμοποιούμε την require για την επιβεβαίωση ότι οι πληρωμές και οι κινήσεις είναι έγκυρες πριν εκτελεστούν, προστατεύοντας έτσι τους παίκτες από λάθη και κακόβουλες ενέργειες.
- **Reentrancy Attack**
  Η επίθεση επανεισόδου συμβαίνει όταν ένας εξωτερικός λογαριασμός καλεί ξανά τη συνάρτηση του συμβολαίου, προκαλώντας ανεπιθύμητες αλλαγές στην κατάσταση του. Στο έργο μας, αποτρέπουμε αυτή την επίθεση, καθώς οι μεταφορές χρημάτων γίνονται μόνο σε συγκεκριμένα σημεία του παιχνιδιού, και όχι μέσα σε

επαναλαμβανόμενες διαδικασίες. Επίσης, χρησιμοποιούμε ασφαλείς συναρτήσεις μεταφοράς χρημάτων.

- **Έλεγχος χρονικών περιορισμών**: Ο χρονικός περιορισμός για κάθε κίνηση και για την ολοκλήρωση του παιχνιδιού διασφαλίζει ότι το παιχνίδι δεν θα παραμείνει ανοιχτό για ατελείωτο χρονικό διάστημα και ότι οι παίκτες δεν θα μπορούν να εκμεταλλευτούν καθυστερήσεις για να αποκτήσουν πλεονέκτημα.

- **Access Control (Ελεγχος Πρόσβασης):**
  Για να διασφαλίσουμε ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε κρίσιμες λειτουργίες, εφαρμόζουμε αυστηρούς ελέγχους πρόσβασης. Χρησιμοποιούμε το modifier onlyOwner για να περιορίσουμε την πρόσβαση σε συγκεκριμένες λειτουργίες και αποτρέπουμε τους χρήστες από το να συμμετέχουν σε περισσότερα από ένα παιχνίδια ταυτόχρονα.

- **Front-running :**
  Η επίθεση front-running συμβαίνει όταν κάποιος εκμεταλλεύεται τη σειρά των συναλλαγών προς όφελός του. Στο έργο μας, περιορίσαμε τις κινήσεις των παικτών ώστε να πραγματοποιούνται με καθορισμένη σειρά, αποτρέποντας την εκμετάλλευση της ροής του παιχνιδιού.

Συμπεράσματα

Γραψαμε τα smart contracts για **CryptoSOS** και **MultiSOS** προσπαθώντας να Χρησιμοποιήσουμε βέλτιστες πρακτικές για τη διαχείριση του gas, την ασφάλεια των συναλλαγών και τους ελέγχους εγκυρότητας. Αντιμετωπίσαμε αρκετές δυσκολίες όσον αναφορά την οικονομία και την ασφάλεια, ενώ το multiSOS είχε και μεγαλύτερη πολυπλοκότητα λογικής.