**Background and Data**

Hello and welcome to the **Among Us** assignment.

Among Us is a video game that was launched in 2018 and became a popular title during the COVID times. It is possible that a few of you know the game and have played it with friends, and others might not have even heard of it.

Often data scientists are expected to work on domains that are not known to them. In such cases, there are two ways to get the required domain knowledge:

1. Approach the domain experts, and
2. Research on your own.

This assignment is going to be on a domain that might be unfamiliar to many of you. We will include the necessary domain knowledge, but if you feel that you lack certain knowledge we encourage you to look online or talk to people who are familiar with the subject.

**Among Us Gameplay**

Among Us is a murder mystery game that can take place in various isolated locations, aka Maps. A map can be in a spaceship named "The Skeld," the headquarters of the company MIRA, a research base on the planet Polus, or an airship.

You can get a fundamental understanding of how the game is played through [this video](https://youtu.be/jLkFKdL1Q20). It is by “Bawsarnold” on YouTube and explains the basics of the game in a short video.

There are two-player roles in the game: the Crewmates and the Impostors. At the beginning of the game, 1-3 players are randomly assigned as impostors, and the rest of them are crewmates. The number of imposters is based on the total number of players.

**The Impostor Role:**

The objective of the impostor is to blend in and kill the crewmates. Impostors can create distractions on the map by sabotage, which results in either a problem that will cause the crew an inconvenience or a severe problem that gives the crew an important task to complete under a time limit. The impostors' goal is to kill off enough crewmates to leave an equal number of impostors and crewmates. This results in a win for the impostors, as they theoretically could not be voted out.

**The Crewmate Role:**

The crewmates' goal is to complete some tasks in the form of mini-games, such as fixing the wiring, emptying garbage, or swiping an ID card, all the while with the goal of finding out the impostors. The crewmates win the game by voting out all impostors.

**Voting**

Voting is an event in the game where all players come together to discuss and vote out potential impostors. A voting discussion can be triggered when a player finds a dead body. When a voting session begins, the players can vote off another player or skip the vote.

**Pregame**

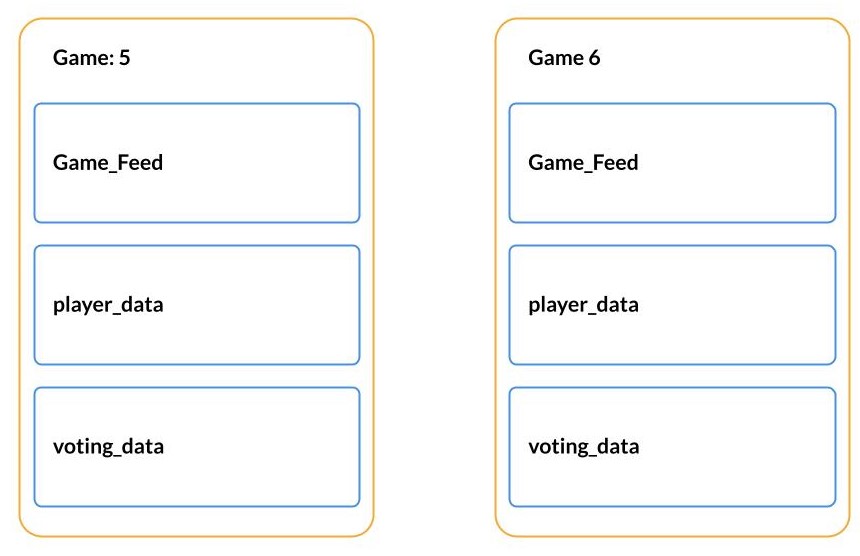
Before the game begins, all players have an option to customize their appearance in the game. Players can select their name, color for their avatar, or even some accessories. But the pregame phase is timed, once the time runs out any player who joins late does not get to customize their avatar the game assigns them an avatar.

Let’s go over the data we will be working with. The data set you will be working with can be found below.

**Among Us Data**

**Download**

The data set has entries 499 Among Us games. Let’s look at one document (one entry in the data) representing one game. The overall structure of the data is shown in the schema below.



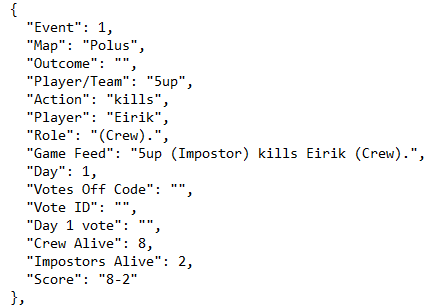
Structure of Data

There are four high-level fields:

1. Game
2. Game feed
3. Player data
4. Voting data

Game is a unique identifier for each game. The other three fields are divided into nested documents with more granular-level data.

A game feed is an array of objects where each object is an event in the game. Each game in Among Us has multiple events. The screenshot below shows the data of one event in one game.



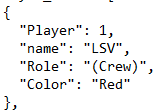
Sample of one event in a particular game

As you can see, each event has 15 fields nested inside it. Given below is the data dictionary for event data fields.

| Data dictionary for one event in a particular game | | |
| --- | --- | --- |
| **Sr. No.** | **Field** | **Details** |
| 1 | Event | Unique identifier for an event in a game. The combination of game and event will create a unique identifier for all events. |
| 2 | Map | The map on which the game is being played. |
| 3 | Outcome | Indicates if the event is the last event of the game. The last event will hold information about who won the game. |
| 4 | Player/team | Name of the player who performed an action. |
| 5 | Action | The action that was performed by the player/team. |
| 6 | Player | Name of the player against whom the action was performed. |
| 7 | Role | Role of the player against whom the action was committed. |
| 8 | Game Feed | The combined form of the above three fields. |
| 9 | Day | Usually, a game is played over multiple days in the game cycle. This is the identifier for each day in the game. |
| 10 | Votes off code | A categorical variable represents the results of a voting discussion.  0    Skipped Vote. 1    Crew voted off. 2    Impostors voted off. |
| 11 | Vote ID | Unique identifier of all voting events in the data set.  For example, vote id 1-1 is the first voting event in the first game. |
| 12 | Day 1 Vote | Results of voting events in words (redundant with Votes off code). |
| 13 | Crew Alive | The number of crew members alive. |
| 14 | Impostors Alive | The number of impostors alive. |
| 15 | Score | Represents the number of crew alive versus number of impostors alive. |

To reiterate, each game will have multiple such events, and all the events in a game will be stored as an array in the Game\_Feed field.

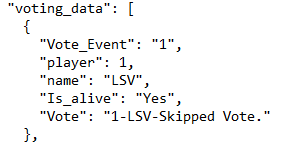
Next, player\_data is a field that holds information about the players who have participated in a specific game. Each game has about 8-10 players, and similar to the Game\_Feed field, player\_data also has an array of player-wise data. The image below shows a typical entry in the player\_data.



Sample of player\_data

| Data dictionary for player\_data | | |
| --- | --- | --- |
| **Sr. no.** | **Field** | **Details** |
| 1 | Player | Unique identifier for each player in each game. |
| 2 | name | Name of the player. |
| 3 | Role | Role of the player in the game: crew or impostor. |
| 4 | Color | The color selected by the player in the game. |

Finally, we have voting data. Similar to the other two, voting data is also a collection of details about each voting event in a single game. The image below shows a typical voting event entry.



Sample for voting\_data

| Data dictionary for voting\_data | | |
| --- | --- | --- |
| **Sr. No.** | **Field** | **Details** |
| 1 | Vote\_Event | Unique identifier for each voting event in a particular game. |
| 2 | player | Unique identifier for players, assigned for every game. This is the same as the field Player in player\_data. For each Vote\_Event, there will be multiple entries, one entry for each player that voted in the event. |
| 3 | name | Name of the player. |
| 4 | Is\_alive | Is the player dead or alive? Dead players cannot vote, so if a player is dead, their vote does not matter. |
| 5 | Vote | Information about the vote cast by the player. |

To reiterate, each game will have multiple voting events, and each voting event will have votes cast by multiple players.

The data descriptions given above describe the details of the nested fields in each high-level field. You can also refer to the sample document attached below to see an example. This is the entire data record for one game. Your data set has 499 such games in total.

**Sample data for 1 document**

**Download**

The data set we are working on is not structured, and loading it into MySQL might not work. Let’s say you want to build a classification model (or any other ML model) on the data that is stored as unstructured or semi-structured data. You can't use unstructured data with classification algorithms. Recall the classification models you have learned so far. Most of them expect data to be structured. One of the jobs of databases like MongoDB is to aggregate the unstructured data and make it structured so that ML algorithms can work on it.

As you can imagine, converting the data to structured data is a tedious and complicated task, and we will try to explore it in this assignment.

In the sext segment, you will find all the tasks you have to perform in this assignment.

**Background and Data**

Hello and welcome to the **Among Us** assignment.

Among Us is a video game that was launched in 2018 and became a popular title during the COVID times. It is possible that a few of you know the game and have played it with friends, and others might not have even heard of it.

Often data scientists are expected to work on domains that are not known to them. In such cases, there are two ways to get the required domain knowledge:

1. Approach the domain experts, and
2. Research on your own.

This assignment is going to be on a domain that might be unfamiliar to many of you. We will include the necessary domain knowledge, but if you feel that you lack certain knowledge we encourage you to look online or talk to people who are familiar with the subject.

**Among Us Gameplay**

Among Us is a murder mystery game that can take place in various isolated locations, aka Maps. A map can be in a spaceship named "The Skeld," the headquarters of the company MIRA, a research base on the planet Polus, or an airship.

You can get a fundamental understanding of how the game is played through [this video](https://youtu.be/jLkFKdL1Q20). It is by “Bawsarnold” on YouTube and explains the basics of the game in a short video.

There are two-player roles in the game: the Crewmates and the Impostors. At the beginning of the game, 1-3 players are randomly assigned as impostors, and the rest of them are crewmates. The number of imposters is based on the total number of players.

**The Impostor Role:**

The objective of the impostor is to blend in and kill the crewmates. Impostors can create distractions on the map by sabotage, which results in either a problem that will cause the crew an inconvenience or a severe problem that gives the crew an important task to complete under a time limit. The impostors' goal is to kill off enough crewmates to leave an equal number of impostors and crewmates. This results in a win for the impostors, as they theoretically could not be voted out.

**The Crewmate Role:**

The crewmates' goal is to complete some tasks in the form of mini-games, such as fixing the wiring, emptying garbage, or swiping an ID card, all the while with the goal of finding out the impostors. The crewmates win the game by voting out all impostors.

**Voting**

Voting is an event in the game where all players come together to discuss and vote out potential impostors. A voting discussion can be triggered when a player finds a dead body. When a voting session begins, the players can vote off another player or skip the vote.

**Pregame**

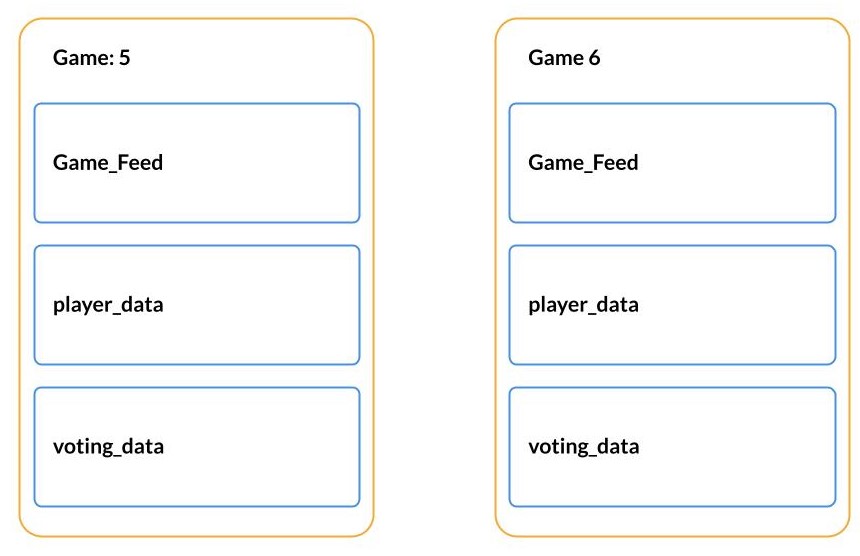
Before the game begins, all players have an option to customize their appearance in the game. Players can select their name, color for their avatar, or even some accessories. But the pregame phase is timed, once the time runs out any player who joins late does not get to customize their avatar the game assigns them an avatar.

Let’s go over the data we will be working with. The data set you will be working with can be found below.

**Among Us Data**

**Download**

The data set has entries 499 Among Us games. Let’s look at one document (one entry in the data) representing one game. The overall structure of the data is shown in the schema below.



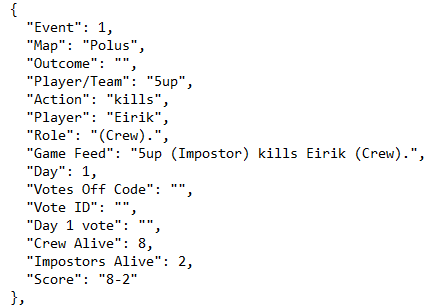
Structure of Data

There are four high-level fields:

1. Game
2. Game feed
3. Player data
4. Voting data

Game is a unique identifier for each game. The other three fields are divided into nested documents with more granular-level data.

A game feed is an array of objects where each object is an event in the game. Each game in Among Us has multiple events. The screenshot below shows the data of one event in one game.



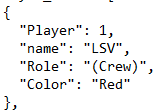
Sample of one event in a particular game

As you can see, each event has 15 fields nested inside it. Given below is the data dictionary for event data fields.

| Data dictionary for one event in a particular game | | |
| --- | --- | --- |
| **Sr. No.** | **Field** | **Details** |
| 1 | Event | Unique identifier for an event in a game. The combination of game and event will create a unique identifier for all events. |
| 2 | Map | The map on which the game is being played. |
| 3 | Outcome | Indicates if the event is the last event of the game. The last event will hold information about who won the game. |
| 4 | Player/team | Name of the player who performed an action. |
| 5 | Action | The action that was performed by the player/team. |
| 6 | Player | Name of the player against whom the action was performed. |
| 7 | Role | Role of the player against whom the action was committed. |
| 8 | Game Feed | The combined form of the above three fields. |
| 9 | Day | Usually, a game is played over multiple days in the game cycle. This is the identifier for each day in the game. |
| 10 | Votes off code | A categorical variable represents the results of a voting discussion.  0    Skipped Vote. 1    Crew voted off. 2    Impostors voted off. |
| 11 | Vote ID | Unique identifier of all voting events in the data set.  For example, vote id 1-1 is the first voting event in the first game. |
| 12 | Day 1 Vote | Results of voting events in words (redundant with Votes off code). |
| 13 | Crew Alive | The number of crew members alive. |
| 14 | Impostors Alive | The number of impostors alive. |
| 15 | Score | Represents the number of crew alive versus number of impostors alive. |

To reiterate, each game will have multiple such events, and all the events in a game will be stored as an array in the Game\_Feed field.

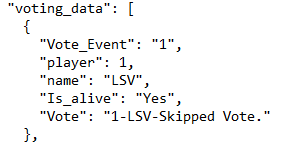
Next, player\_data is a field that holds information about the players who have participated in a specific game. Each game has about 8-10 players, and similar to the Game\_Feed field, player\_data also has an array of player-wise data. The image below shows a typical entry in the player\_data.



Sample of player\_data

| Data dictionary for player\_data | | |
| --- | --- | --- |
| **Sr. no.** | **Field** | **Details** |
| 1 | Player | Unique identifier for each player in each game. |
| 2 | name | Name of the player. |
| 3 | Role | Role of the player in the game: crew or impostor. |
| 4 | Color | The color selected by the player in the game. |

Finally, we have voting data. Similar to the other two, voting data is also a collection of details about each voting event in a single game. The image below shows a typical voting event entry.



Sample for voting\_data

| Data dictionary for voting\_data | | |
| --- | --- | --- |
| **Sr. No.** | **Field** | **Details** |
| 1 | Vote\_Event | Unique identifier for each voting event in a particular game. |
| 2 | player | Unique identifier for players, assigned for every game. This is the same as the field Player in player\_data. For each Vote\_Event, there will be multiple entries, one entry for each player that voted in the event. |
| 3 | name | Name of the player. |
| 4 | Is\_alive | Is the player dead or alive? Dead players cannot vote, so if a player is dead, their vote does not matter. |
| 5 | Vote | Information about the vote cast by the player. |

To reiterate, each game will have multiple voting events, and each voting event will have votes cast by multiple players.

The data descriptions given above describe the details of the nested fields in each high-level field. You can also refer to the sample document attached below to see an example. This is the entire data record for one game. Your data set has 499 such games in total.

**Sample data for 1 document**

**Download**

The data set we are working on is not structured, and loading it into MySQL might not work. Let’s say you want to build a classification model (or any other ML model) on the data that is stored as unstructured or semi-structured data. You can't use unstructured data with classification algorithms. Recall the classification models you have learned so far. Most of them expect data to be structured. One of the jobs of databases like MongoDB is to aggregate the unstructured data and make it structured so that ML algorithms can work on it.

As you can imagine, converting the data to structured data is a tedious and complicated task, and we will try to explore it in this assignment.

In the sext segment, you will find all the tasks you have to perform in this assignment.

**Task List**

Given below is a list of tasks and subtasks, some of which are compulsory and some optional. We recommend you complete all of them since the end objective is to create a structured data table that can be used for predictive or descriptive analysis.

General instructions:

1. Command prompt file will not be sharable as a submission, so you can make your submission as a document, a presentation, or a pdf file.
2. In your submission, include
   1. The query you have written to answer each task.
   2. Screenshot of the result you got.

You can use a template like the one linked [here](https://docs.google.com/presentation/d/1Of9QVMi6AIQNcJE6LcWFN3eb2AeiPrkq4c8M_Ll5Saw/edit?usp=sharing)to create your submission file. The file attached is an example and you are free to use any other format.

**Tasks**

1. **Read the data. (10%)**
   1. Read in the data using the mongoimport command.  
        
      Hint: The data is in the form of an array of JSON objects. So the mongoimport command you have used in the module will not work directly. You will have to let mongoimport know that the file you are importing is not just a JSON file but an array of JSON files. The syntax mongoimport for the JSON array is

mongoimport <usual mongo import command> -- jsonArray

* 1. Display data for the match with “game” = “3”  
       
     Suggestion: Use the pretty function to make the data readable.

1. **Explore the data for game 3. (40%)**  
     
   In this subtask, you are expected to create a new collection with only the document relating to game 3. You can create the new document with the help of the code below.   
     
   db.<new\_collection\_name>.save( <query that you used to display data for match 3>.toArray()  )  
   1. Display the Game Feed data for the game in the new collection.
   2. Display the last event in game 3.  
      Hint: Event is a nested field. You can use the dot operator to access the nested fields. For example, to access the Action field inside the Game Feed, you can write the query as  
        
      {"$Game\_Feed.Action": <condition>}
   3. Who won game 3, imposters or crew?  
      How will you know who won the game? check the 'Game Feed' for all the events in game 3.   
      Write a query to display only the Game Feed field for game 3. Note the lack of underscore between Game and Feed. This field is a nested field inside the Game\_Feed field. Refer to serial number 8 in the Game\_Feed data dictionary.
   4. Who picked the black color in game 3? Was that player crew or imposter?
   5. How many voting events happened in game 3?
   6. This task was focused on exploring a single document. As explained earlier the document is made of 4 main fields and three of those are nested and hold a lot of data.  
      If you were to redesign this database to make it easier to query what changes would you make to the structure? Explain your design decisions. Do not modify the structure, simply explain the changes you want to make and justify them.
2. **Overall aggregation. (40%)**  
   For this set of questions, use the earlier data collection with all 499 games.  
   1. How many events in total do you have data for, in this collection (across all games)?   
      An event as explained earlier is a record of development in the game. For instance, game 3 had 10 events.
   2. How many matches did the crew win versus how many matches did the impostors win?  
      Hint: Use regular expressions to analyze the game feed column for all entries documents where the Outcome field is not empty.
   3. Find out how many matches were played on each map. Your output should contain the names of maps and the number of games played on each map.   
      Hint: You will have to ensure only one event from each game is filtered before grouping.
   4. How many times in total across all games did the crew skip a vote?
   5. How many times in total across all matches does the crew vote against imposters?
   6. The questions you answered in this task were all related to high-level aggregations across the entire collection. In your opinion, is the game more or less hard for impostors? Justify your answer with suitable insights from the data.
3. **Player-level aggregation. (10%)**  
   In this set of tasks, we will aggregate the data player-wise. Let’s say that we want to create a dashboard to select players for games. You have to convert the data to a structured format where each row represents one player and some of their statistics.
   1. Find the number of unique players in the data set.  
      Hint: Check the data type of the output of the query you write to get the set of unique players, and then decide how you will calculate the number of unique players.
   2. Who is the best crew member?  
      Find the player who (as a crewmate) voted to remove the imposters the most number of times.
   3. Who is the worst crew member?  
      Find the player who (as a crewmate) voted to remove the other crew members the most number of times.
   4. Find the win percentage for all players. *(Optional)*  
      Win percentage can be calculated by the total number of games the player has won divided by the total matches the player has participated in.
   5. Find the color preference of all players. *(Optional)*  
      The color preference of a player is the color they picked the most number of times.
4. **Create an export from MongoDB in the form given below (Optional)**  
   As discussed earlier the aim of this exercise was to create a table for predictive and descriptive analysis. The table given below is one way to create a structured output from the given data.

| **Player name** | **Games won as imposter** | **Games won as crew** | **Win percentage (overall)** | **Voted Against Imposter** | **Voted against Crew members** | **Color preference** | **Voting rate** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

1. In tasks 4 and 5 you create some player-wise statistics. Explain statistics that you might want to explore (other than the ones that were calculated earlier) during the exercise of selecting players.

Bối cảnh và Dữ liệu

Xin chào và chào mừng đến với nhiệm vụ Giữa chúng ta.

Giữa chúng ta là một trò chơi điện tử được ra mắt vào năm 2018 và đã trở thành một tựa game nổi tiếng trong thời kỳ COVID. Có thể một số bạn biết trò chơi này và đã chơi với bạn bè, còn những người khác thậm chí có thể chưa từng nghe nói về trò chơi này.

Thông thường, các nhà khoa học dữ liệu phải làm việc trên các lĩnh vực mà họ không biết. Trong những trường hợp như vậy, có hai cách để có được kiến thức miền cần thiết:

1. Tiếp cận các chuyên gia tên miền và

2. Tự nghiên cứu.

Bài tập này sẽ thực hiện trên một miền có thể không quen thuộc với nhiều bạn. Chúng tôi sẽ bao gồm kiến thức miền cần thiết, nhưng nếu bạn cảm thấy mình thiếu một số kiến thức nhất định, chúng tôi khuyến khích bạn tìm kiếm trực tuyến hoặc nói chuyện với những người quen thuộc với chủ đề này.

Trò chơi giữa chúng ta

Giữa chúng ta là một trò chơi giết người bí ẩn có thể diễn ra ở nhiều địa điểm biệt lập khác nhau, hay còn gọi là Bản đồ. Bản đồ có thể nằm trong con tàu vũ trụ có tên "The Skeld", trụ sở của công ty MIRA, cơ sở nghiên cứu trên hành tinh Polus hoặc một chiếc airship.

Bạn có thể hiểu cơ bản về cách chơi trò chơi thông qua video này. Đó là của “Bawsarnold” trên YouTube và giải thích những điều cơ bản của trò chơi trong một video ngắn.

Có hai vai trò người chơi trong trò chơi: Đồng đội và Kẻ mạo danh. Khi bắt đầu trò chơi, 1-3 người chơi được chỉ định ngẫu nhiên làm kẻ mạo danh và những người còn lại là đồng đội. Số lượng kẻ mạo danh dựa trên tổng số người chơi.

Vai trò kẻ mạo danh:

Mục tiêu của kẻ mạo danh là trà trộn vào và giết các thành viên phi hành đoàn. Những kẻ mạo danh có thể tạo ra sự phân tâm trên bản đồ bằng cách phá hoại, điều này dẫn đến một vấn đề gây bất tiện cho nhóm hoặc một vấn đề nghiêm trọng khiến nhóm phải hoàn thành một nhiệm vụ quan trọng trong thời hạn. Mục tiêu của những kẻ mạo danh là giết đủ thành viên phi hành đoàn để số lượng kẻ mạo danh và thành viên phi hành đoàn bằng nhau. Điều này dẫn đến chiến thắng cho những kẻ mạo danh, vì về mặt lý thuyết, chúng không thể bị bỏ phiếu.

Vai trò thuyền viên:

Mục tiêu của các thành viên phi hành đoàn là hoàn thành một số nhiệm vụ dưới dạng trò chơi nhỏ, chẳng hạn như sửa hệ thống dây điện, đổ rác hoặc quẹt thẻ ID, đồng thời với mục tiêu tìm ra những kẻ mạo danh. Các thành viên phi hành đoàn giành chiến thắng trong trò chơi bằng cách bỏ phiếu loại bỏ tất cả những kẻ mạo danh.

bỏ phiếu

Bỏ phiếu là một sự kiện trong trò chơi nơi tất cả người chơi cùng nhau thảo luận và bỏ phiếu loại bỏ những kẻ mạo danh tiềm năng. Một cuộc thảo luận bỏ phiếu có thể được kích hoạt khi người chơi tìm thấy xác chết. Khi phiên bỏ phiếu bắt đầu, người chơi có thể bỏ phiếu cho người chơi khác hoặc bỏ phiếu.

chuẩn bị trước

Trước khi trò chơi bắt đầu, tất cả người chơi có một tùy chọn để tùy chỉnh giao diện của họ trong trò chơi. Người chơi có thể chọn tên, màu sắc cho hình đại diện hoặc thậm chí một số phụ kiện. Nhưng giai đoạn trước trò chơi được tính thời gian, khi hết thời gian, bất kỳ người chơi nào tham gia muộn sẽ không thể tùy chỉnh hình đại diện của họ, trò chơi sẽ chỉ định cho họ một hình đại diện.

Hãy xem qua dữ liệu mà chúng ta sẽ làm việc. Tập dữ liệu bạn sẽ làm việc có thể được tìm thấy bên dưới.

Dữ liệu giữa chúng ta

Tải xuống

Tập dữ liệu có các mục 499 trò chơi Giữa chúng ta. Hãy xem một tài liệu (một mục trong dữ liệu) đại diện cho một trò chơi. Cấu trúc tổng thể của dữ liệu được hiển thị trong lược đồ bên dưới.

Cấu trúc dữ liệu

Có bốn trường cấp cao:

1. Trò chơi

2. Nguồn cấp dữ liệu trò chơi

3. Dữ liệu người chơi

4. Dữ liệu bình chọn

Trò chơi là một mã định danh duy nhất cho mỗi trò chơi. Ba trường còn lại được chia thành các tài liệu lồng nhau với dữ liệu cấp độ chi tiết hơn.

Nguồn cấp dữ liệu trò chơi là một mảng các đối tượng trong đó mỗi đối tượng là một sự kiện trong trò chơi. Mỗi trò chơi trong Giữa chúng ta có nhiều sự kiện. Ảnh chụp màn hình bên dưới hiển thị dữ liệu của một sự kiện trong một trò chơi.

Ví dụ về một sự kiện trong một trò chơi cụ thể

Như bạn có thể thấy, mỗi sự kiện có 15 trường được lồng vào bên trong. Dưới đây là từ điển dữ liệu cho các trường dữ liệu sự kiện.

Từ điển dữ liệu cho một sự kiện trong một trò chơi cụ thể

Sr. STT Lĩnh vực Chi tiết

1 Sự kiện Định danh duy nhất cho một sự kiện trong trò chơi. Sự kết hợp giữa trò chơi và sự kiện sẽ tạo ra một mã định danh duy nhất cho tất cả các sự kiện.

2 Bản đồ Bản đồ mà trò chơi đang được chơi.

3 Kết quả Cho biết sự kiện này có phải là sự kiện cuối cùng của trò chơi hay không. Sự kiện cuối cùng sẽ chứa thông tin về người đã thắng trò chơi.

4 Người chơi/đội Tên của người chơi đã thực hiện một hành động.

5 Hành động Hành động được thực hiện bởi người chơi/đội.

6 Người chơi Tên của người chơi mà hành động được thực hiện.

7 Vai trò Vai trò của người chơi chống lại hành động đã được thực hiện.

8 Game Feed Dạng tổng hợp của 3 trường trên.

9 Ngày Thông thường, một trò chơi được chơi trong nhiều ngày trong chu kỳ trò chơi. Đây là định danh cho mỗi ngày trong trò chơi.

10 Mã bỏ phiếu Một biến phân loại đại diện cho kết quả của một cuộc thảo luận bỏ phiếu.

0 Bỏ phiếu.

1 phi hành đoàn đã bỏ phiếu.

2 kẻ mạo danh đã bỏ phiếu.

11 ID bầu cử Mã định danh duy nhất của tất cả các sự kiện bỏ phiếu trong tập dữ liệu.

Ví dụ: id bầu chọn 1-1 là sự kiện bỏ phiếu đầu tiên trong trò chơi đầu tiên.

12 Ngày 1 Bình chọn Kết quả của các sự kiện bình chọn bằng chữ (dư thừa với mã Phiếu bầu).

13 Phi hành đoàn còn sống Số lượng phi hành đoàn của tôi