MongoDb Project on AmongUs Games

Submitted by - Maya Babu

Instructor - Maryam Kazemi

Introduction

- Among Us is a multiplayer PC and mobile game that has suddenly exploded in popularity, becoming one of the hit video games of 2020. Created by an indie game company in 2018, Among Us remained under the radar until the summer of the pandemic. The game involves 4-10 players dropped onto an alien spaceship, each assigned as a 'crewmate' or 'impostor'. Crewmates must complete tasks and root out impostors to win, while impostors must discreetly eliminate crewmates. Players can be voted off the ship, adding a layer of strategy and survival.

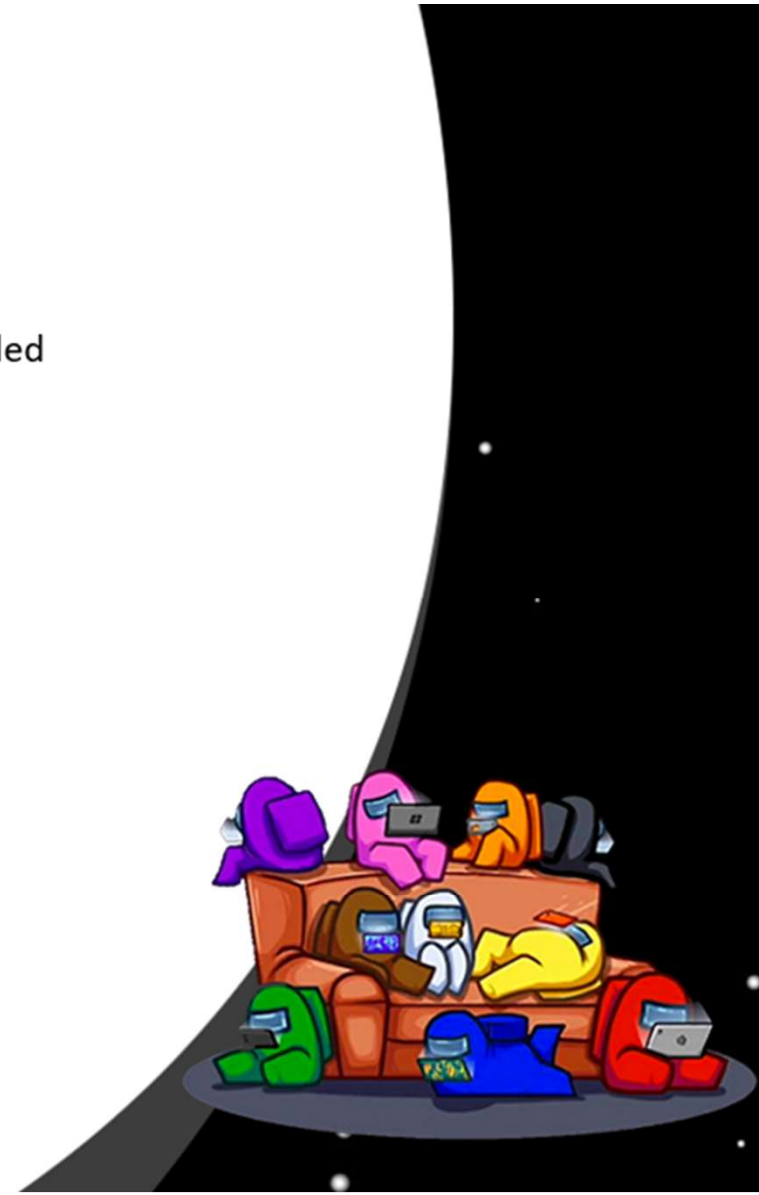
Data Understanding

In this project, we analyze a dataset from 499 Among Us games, seeking detailed insights into gameplay using MongoDB.

The dataset contains four main fields:

- Game
- Game Feed
- Player Data
- Voting Data

each offering unique perspectives on the gameplay.



➢ 1.1. Read the data and examine the collections.

```
C:\Program Files\MongoDB\Server\7.0\bin>mongoimport -d dbAmongUs -c Among_Us_data --type json --file AmongUs.json --json
Array
2024-05-04T22:59:51.894-0400    connected to: mongodb://localhost/
2024-05-04T22:59:52.101-0400    499 document(s) imported successfully. 0 document(s) failed to import.
```

```
test> use dbAmongUs
switched to db dbAmongUs
dbAmongUs> db.Among_Us_data.find().limit(5)
```


```
[
  {
    _id: ObjectId('6636f628f98aa15b0d11b60a'),
    game: '5',
    Game_Feed: [
      {
        Event: 1,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': '5up',
        Action: 'kills',
        Player: 'Eirik',
        Role: '(Crew).',
        'Game Feed': '5up (Impostor) kills Eirik (Crew).',
        Day: 1,
        'Votes Off Code': '',
        'Vote ID': '',
        'Day 1 vote': '',
        'Crew Alive': 8,
        'Impostors Alive': 2,
        Score: '8-2'
      },
      {
        Event: 2,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'wrapter',
        Action: 'finds body of',
        Player: 'Eirik',
        Role: '(Crew).',
        'Game Feed': 'wrapter (Crew) finds body of Eirik (Crew).',
        Day: 1,
        'Votes Off Code': 0,
        'Vote ID': '5-1',
        'Day 1 vote': 'Skipped Vote.',
        'Crew Alive': 7,
        'Impostors Alive': 2,
        Score: '7-2'
      },
```

- 1.2. Display data for matches where the "game" field equals "3".

```
dbAmongUs> db.Among_Us_data.find({'game':'3'})
[
  {
    _id: ObjectId('6636f628f98aa15b0d11b61f'),
    game: '3',
    Game_Feed: [
      {
        Event: 1,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'kills',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) kills BK (Crew).',
        Day: 1,
        'Votes Off Code': '',
        'Vote ID': '',
        'Day 1 vote': '',
        'Crew Alive': 8,
        'Impostors Alive': 2,
        Score: '8-2'
      },
      {
        Event: 2,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'finds body of',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) finds body of BK (Crew).',
        Day: 1,
        'Votes Off Code': 0,
        'Vote ID': '3-1',
        'Day 1 vote': 'Skipped Vote.',
        'Crew Alive': 7,
        'Impostors Alive': 2,
        Score: '7-2'
      },
```

- 2. Create a new collection with only the document relating to game 3.

```
dbAmongUs>  var game3Document=db.Among_Us_data.findOne({'game':'3'})

dbAmongUs> db.Game3Collection.insertOne(game3Document)
{
  acknowledged: true,
  insertedId: ObjectId('6636f628f98aa15b0d11b61f')
}
dbAmongUs> db.Game3Collection.find().pretty()
[
  {
    _id: ObjectId('6636f628f98aa15b0d11b61f'),
    game: '3',
    Game_Feed: [
      {
        Event: 1,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'kills',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) kills BK (Crew).',
        Day: 1,
        'Votes Off Code': '',
        'Vote ID': '',
        'Day 1 vote': '',
        'Crew Alive': 8,
        'Impostors Alive': 2,
        Score: '8-2'
      },
      {
        Event: 2,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'finds body of',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) finds body of BK (Crew).',
        Day: 1,
        'Votes Off Code': 0,
        'Vote ID': '3-1',
        'Day 1 vote': 'Skipped Vote.',
        'Crew Alive': 7,
        'Impostors Alive': 2,
        Score: '7-2'
      },
```

- 2.1. Show the Game Feed data specifically for game 3 in the newly created collection.

```
dbAmongUs>  db.Game3Collection.find({}, { Game_Feed: 1, _id: 0 }).pretty()
[
  {
    Game_Feed: [
      {
        Event: 1,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'kills',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) kills BK (Crew).',
        Day: 1,
        'Votes Off Code': '',
        'Vote ID': '',
        'Day 1 vote': '',
        'Crew Alive': 8,
        'Impostors Alive': 2,
        Score: '8-2'
      },
      {
        Event: 2,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'finds body of',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) finds body of BK (Crew).',
        Day: 1,
        'Votes Off Code': 0,
        'Vote ID': '3-1',
        'Day 1 vote': 'Skipped Vote.',
        'Crew Alive': 7,
        'Impostors Alive': 2,
        Score: '7-2'
      },
```

- 2.2.Show the most recent event that occurred in game3.

```
dbAmongUs> db.Game3Collection.find({},{'Game_feed':{$slice:-1},_id:0})
[
  {
    game: '3',
    Game_Feed: [
      {
        Event: 1,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'kills',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) kills BK (Crew).',
        Day: 1,
        'Votes Off Code': '',
        'Vote ID': '',
        'Day 1 vote': '',
        'Crew Alive': 8,
        'Impostors Alive': 2,
        Score: '8-2'
      },
      {
        Event: 2,
        Map: 'Polus',
        Outcome: '',
        'Player/Team': 'Keaton',
        Action: 'finds body of',
        Player: 'BK',
        Role: '(Crew).',
        'Game Feed': 'Keaton (Impostor) finds body of BK (Crew).',
        Day: 1,
        'Votes Off Code': 0,
        'Vote ID': '3-1',
        'Day 1 vote': 'Skipped Vote.',
        'Crew Alive': 7,
        'Impostors Alive': 2,
        Score: '7-2'
```

- 2.3. Determine the winner of game 3 - imposters or crew.

Answer: Crew

```
dbAmongUs> db.Game3Collection.find( {"Game_Feed.Outcome":{$regex:"End"}
},{"_id":0, "Game_Feed.Game Feed.$":1}).pretty()
[
  { Game_Feed: [ { 'Game Feed': 'Crew Win - Voting' } ] }
]
```

- 2.4.Identify the player who chose the black color in game 3 and whether they were a crew member or imposter.

Answer: Player Name: Keaton, Role: Impostor

```
dbAmongUs> db.Game3Collection.findOne({'player_data.Color':'Black'},{'player_d
ata.$':1,_id:0})
{
  player_data: [ { Player: 9, name: 'Keaton', Role: '(Impostor)', Color: 'Blac
k' } ]
}
```

- 2.5 Count the number of voting events that took place in game 3.

Answer: 3

```
dbAmongUs> db.Game3Collection.aggregate([{$match:{'game':'3'}},{$unwind:
'$voting_data'},{$group:{_id:'$voting_data.Vote_Event'}},{$count:'total_
voting_events'}])
[ { total_voting_events: 3 } ]
```

- 2.6.If you were redesigning this database for easier querying, describe the changes you would make and explain your reasoning.

Normalize Data Structures:
Split the data into related collections instead of a single collection with nested arrays. This would reduce data redundancy and make updates easier. For instance, player_data, Game_Feed, and voting_data can be separate collections, linked by a game_id.
Use consistent naming conventions and data types for fields across collections to make queries more intuitive.

Timestamps for Events:
Include timestamp fields for game events and voting events. This allows for more straightforward chronological sorting and querying of events.

Separate Collection for Player Profile:
Instead of repeating player information in each game's player_data, have a separate collection for player profiles. Link these profiles to games through player IDs. This reduces redundancy and makes updating player information easier.

Flatten Document Structures:
Reduce the depth of nested documents. Deeply nested structures can make queries complex and slow. Flattening the structure, while maintaining relationships through references, can simplify queries.

Include Aggregation Data in the Document:
For frequently calculated data (like the total number of voting events), consider storing this aggregated data directly in the game document if it doesn't change often. This reduces the need for aggregation queries.

- 3.1.Calculate the total number of events recorded in this collection across all games.

Answer: 5889

```
dbAmongUs> db.Among_Us_data.aggregate([{$unwind:'$Game_Feed'},
{$group:{_id:null,totalEvents:{$sum:1}}}])
[ { _id: null, totalEvents: 5889 } ]
dbAmongUs>
```

- 3.2.Compare the crew's wins to the impostors' wins and provide the counts.

Answer: Crew wins 323 times, impostors win 176 times

```
dbAmongUs>  db.Among_Us_data.aggregate([
... {$project:{lastEvent:{$arrayElemAt:['$Game_Feed',-1]}}},
... {$project:{outcome:'$lastEvent.Game Feed'}},
... {$group:{_id:{$cond:{if:{$regexMatch:{input:'$outcome',regex:'Crew Win'}},
... then:'Crew',else:'Impostor'}},count:{$sum:1}}])
[ { _id: 'Impostor', count: 176 }, { _id: 'Crew', count: 323 } ]
```

- 3.3.List the maps played and the total number of games on each map.

Answer: “Polus”: 404, “MIRA HQ”: 7, “The Skeld”: 88

```
dbAmongUs> db.Among_Us_data.aggregate([
... {$unwind:'$Game_Feed'},
... {$match:{'Game_Feed.Event':1}},
... {$group:{_id:'$Game_Feed.Map',count:{$sum:1}}}])
[
  { _id: 'Polus', count: 404 },
  { _id: 'MIRA HQ', count: 7 },
  { _id: 'The Skeld', count: 88 }
]
```

- 3.4.Determine the total instances of crew members skipping a vote across all games.

Answer: 693

```
dbAmongUs> db.Among_Us_data.aggregate([
... {$unwind:'$Game_Feed'},
... {$match:{'Game_Feed.Votes Off Code':0,}},
... {$count:'skippedVotes'}])
[ { skippedVotes: 693 } ]
```

- 3.5.Calculate the total occurrences of crew members voting against imposters across all matches.

Answer: 639

```
dbAmongUs> db.Among_Us_data.aggregate([
... {$unwind:'$Game_Feed'},
... {$match:{'Game_Feed.Votes Off Code': 2,}},
... {$count:'skippedVotes'}])
[ { skippedVotes: 639 } ]
```

➢ 3.6.Share your opinion on whether the game is more or less challenging for impostors, supported by insights from the data.

Win Rates:
Crew members have won 323 times, while impostors have won 176 times. This significant difference in win rates suggests that the game might be more challenging for impostors. A higher win rate for the crew indicates that they are more effective, on average, at completing their tasks and identifying impostors.

Skipping Votes:
There are 693 instances of crew members skipping votes. Skipped votes can imply uncertainty or a lack of consensus among crew members. This situation could be advantageous for impostors as it might indicate missed opportunities to identify and vote off impostors. However, the fact that the crew still wins more often suggests they overcome this challenge effectively.

Voting Against Impostors:
There are 639 instances of crew members voting against impostors. This high number, combined with the crew's higher win rate, might indicate that crew members are generally successful in identifying and voting off impostors, adding to the challenge for impostors to remain undetected.

In conclusion, the data suggests that the game poses a greater challenge for impostors. The higher win rate for crew members, combined with a substantial number of events where the crew votes against impostors, indicates that impostors might find it more difficult to deceive the crew and achieve their objectives.

- 4.1.Find the count of unique players in the dataset.

Answer: 108

```
dbAmongUs> db.Among_Us_data.distinct('player_data.name').length
108
```

- 4.2.Identify the player considered the best crew member.

Answer: 'BK'

```
dbAmongUs> db.Among_Us_data.aggregate([
... { $unwind: "$voting_data"},
... { $match: {"voting_data.Vote": { $regex:"Impostor voted off"}} },
... { $group: { _id:"$voting_data.name", Count: { $sum:1}}},
... { $sort: { Count: -1 } },
... { $limit: 1 }])
[ { _id: 'BK', Count: 178 } ]
```

- 4.3.Identify the player regarded as the least effective crew member.

Answer: 'Sam'

```
dbAmongUs> db.Among_Us_data.aggregate([
... {$unwind:"$voting_data"},
... {$match:{"voting_data.Vote":{$regex:"Crew voted off"}}},
... {$group:{_id:"$voting_data.name",Count:{$sum:1}}},
... {$sort:{Count:-1}},
... {$limit:1}])
[ { _id: 'Sam', Count: 60 } ]
```