Project Title:

Graph Analysis of Upcoming Video Game Releases by Platform

Objective:

To explore the relationships and patterns among video game releases from 2022 to 2024, focusing on connectivity and influence among gaming platforms.

Data Set Description:

The dataset contains information about the number of major video games scheduled for release between 2022 and 2024, categorized by platform. This dataset is interesting as it provides insights into the gaming industry's trends, the popularity of platforms, and the potential saturation of the market.

Program Analyses and Explanation:

Graph Construction:

Nodes: Each node represents a video game platform (e.g., PC, Xbox, PlayStation).

Edges: Connect platforms if games are released on multiple platforms within the same timeframe.

Explanation:

Struct GameGraph: This structure maintains the graph and a map of platform names to their respective indices in the graph.

Method add\_platform: Adds a platform to the graph if it isn't already there.

Method add\_release\_edge: Adds an edge between two platforms, assuming shared releases connect platforms.

Method build\_graph: This method accepts a vector of GameRelease (defined in data\_preparation.rs) and constructs the graph by linking platforms based on shared game releases.

Six Degrees of Separation:

Goal: Determine the average path length between any two platforms, exploring the 'six degrees of separation' concept in the context of video game platform connectivity.

Method: Use Breadth-First Search (BFS) and Shortest Paths algorithms to calculate distances.

Explanation:

Function average\_path\_length: This function computes the average shortest path length between all pairs of nodes in the graph. It iterates over each node, performing BFS to determine the shortest paths from that node to all others, and accumulates these distances.

Function bfs\_shortest\_paths: This is a helper function that performs BFS starting from a given node and records the shortest path distances to all other nodes. It returns a map of node indices to their respective distances from the start node.

Integration: To use these functions, ensure that the GameGraph is being correctly constructed and passed from the graph\_construction.rs module. The example\_usage function demonstrates how to call the average\_path\_length function.

Degree Distributions:

Goal: Analyze the degree distribution of the graph to understand how connected each platform is in terms of shared game releases.

Method: Calculate the degree of each vertex (platform) and examine the distribution, comparing it to theoretical distributions like power-law.

Explanation：

Function compute\_degrees: This function calculates the degree (number of outgoing edges) for each platform in the graph and stores the results in a HashMap linking platform names to their respective degrees.

Function analyze\_degree\_distribution: After computing the degrees, this function calculates how many platforms have each degree, effectively producing the degree distribution. It then prints out this distribution.

Integration: These functions should be used after the graph has been constructed using the graph\_construction.rs module. The example\_usage function demonstrates how to invoke these operations.

Centrality Measures:

Goal: Identify the most influential gaming platforms based on various centrality measures.

Method: Compute centrality metrics such as closeness and betweenness to determine the influence of each platform.

Explanation：

Function closeness\_centrality: This function calculates closeness centrality for each platform. Closeness centrality is calculated as the inverse of the sum of the shortest paths to all other nodes. This function uses the all\_pairs\_dijkstra method from petgraph to compute shortest path distances from each node to all other nodes and then calculates the centrality.

Densest Subgraph:

Goal: Identify periods when the gaming market is most competitive across platforms.

Method: Implement the 2-approximation algorithm for finding a dense subgraph to discover times with the highest game release density.

Explanation：

Function find\_densest\_subgraph: This function attempts to find the densest subgraph by simplifying the problem to removing one node at a time and calculating the density of the resulting subgraph. It iterates over each node, removes it, reconstructs the subgraph, and calculates its density.

Implementation Steps:

Data Preparation: Convert the Excel data into a suitable format for graph analysis, mapping game releases to platforms and times.

Graph Construction: Develop the graph structure in Rust, ensuring all nodes and edges are correctly represented.

Algorithm Implementation: Code the BFS, Shortest Paths, and centrality measure algorithms in Rust.

Explanation：

Functions breadth\_first\_search and depth\_first\_search: These functions are general implementations of BFS and DFS, respectively. Each function takes the graph, a starting node index, and a function to apply to each visited node. This design allows you to perform different actions during traversal, such as updating metrics or collecting data.

Analysis Execution: Run all analyses on the constructed graph, collecting and interpreting results.

Documentation and Final Submission: Compile the findings into a comprehensive report, detailing methodologies, results, and insights.

#Not confirm the time setting, might plan differently.

Testing Strategy:

Unit Testing: Implement unit tests for each graph algorithm to ensure correctness.

Integration Testing: Test the integration of data import, graph construction, and analysis modules.

Explanation:

Modular Tests: Each test is wrapped in a mod tests with the #[cfg(test)] attribute. This ensures they're only compiled and run when you execute cargo test.

Using super(Deleted)::\*: This imports all public items from the parent module into the scope of the tests, making it easy to test any public function or struct.

Test Functions: Each test function is annotated with #[test]. This tells Rust's testing framework that this is a test to run. Inside each function, typically:

1. Set up any necessary data.
2. Call the function(s) you are testing.
3. Assert conditions about the output.

Assertions: Use assert! or assert\_eq! to ensure that the function behaves as expected. These will panic if the conditions are not met, indicating a test failure.

Validation(Delete): Compare the graph's degree distribution and centrality results with expected theoretical outcomes.

Conclusion of plan:

This project will provide valuable insights into the interconnectivity of video game platforms and market trends. By applying graph theory, we can uncover underlying patterns and influences in the video game industry, offering a unique perspective on its competitive landscape.

Analysis of Results and reflection:

1. Data and results:

"PC Games": 156,

"PlayStation 4 Games": 78,

"PlayStation 5 Games": 115,

"PlayStation Not Xbox Games": 22,

"Switch Games": 73,

"Switch Only Games": 33,

"Xbox Not PlayStation Games": 26,

"Xbox One Games": 76,

"Xbox Series X/S Games": 113

path\_lengths, average\_path\_length

({'PC Games': {'PC Games': 0,

'PlayStation 4 Games': 1,

'PlayStation 5 Games': 2,

'PlayStation Not Xbox Games': 3,

'Switch Games': 4,

'Switch Only Games': 5,

'Xbox Not PlayStation Games': 6,

'Xbox One Games': 7,

'Xbox Series X/S Games': 8},

'PlayStation 4 Games': {'PlayStation 4 Games': 0,

'PC Games': 1,

'PlayStation 5 Games': 1,

'PlayStation Not Xbox Games': 2,

'Switch Games': 3,

'Switch Only Games': 4,

'Xbox Not PlayStation Games': 5,

'Xbox One Games': 6,

'Xbox Series X/S Games': 7},

'PlayStation 5 Games': {'PlayStation 5 Games': 0,

'PlayStation 4 Games': 1,

'PlayStation Not Xbox Games': 1,

'PC Games': 2,

'Switch Games': 2,

'Switch Only Games': 3,

'Xbox Not PlayStation Games': 4,

'Xbox One Games': 5,

'Xbox Series X/S Games': 6},

'PlayStation Not Xbox Games': {'PlayStation Not Xbox Games': 0,

'Switch Games': 1,

'PlayStation 5 Games': 1,

'PlayStation 4 Games': 2,

'Switch Only Games': 2,

'Xbox Not PlayStation Games': 3,

'PC Games': 3,

'Xbox One Games': 4,

'Xbox Series X/S Games': 5},

'Switch Games': {'Switch Games': 0,

'Switch Only Games': 1,

'PlayStation Not Xbox Games': 1,

'Xbox Not PlayStation Games': 2,

'PlayStation 5 Games': 2,

'PlayStation 4 Games': 3,

'Xbox One Games': 3,

'Xbox Series X/S Games': 4,

'PC Games': 4},

'Switch Only Games': {'Switch Only Games': 0,

'Xbox Not PlayStation Games': 1,

'Switch Games': 1,

'Xbox One Games': 2,

'PlayStation Not Xbox Games': 2,

'Xbox Series X/S Games': 3,

'PlayStation 5 Games': 3,

'PlayStation 4 Games': 4,

'PC Games': 5},

'Xbox Not PlayStation Games': {'Xbox Not PlayStation Games': 0,

'Xbox One Games': 1,

'Switch Only Games': 1,

'Xbox Series X/S Games': 2,

'Switch Games': 2,

'PlayStation Not Xbox Games': 3,

'PlayStation 5 Games': 4,

'PlayStation 4 Games': 5,

'PC Games': 6},

'Xbox One Games': {'Xbox One Games': 0,

'Xbox Series X/S Games': 1,

'Xbox Not PlayStation Games': 1,

'Switch Only Games': 2,

'Switch Games': 3,

'PlayStation Not Xbox Games': 4,

'PlayStation 5 Games': 5,

'PlayStation 4 Games': 6,

'PC Games': 7},

'Xbox Series X/S Games': {'Xbox Series X/S Games': 0,

'Xbox One Games': 1,

'Xbox Not PlayStation Games': 2,

'Switch Only Games': 3,

'Switch Games': 4,

'PlayStation Not Xbox Games': 5,

'PlayStation 5 Games': 6,

'PlayStation 4 Games': 7,

'PC Games': 8}},

3.3333333333333335)

1. Graph\_Construction analysis:

Nodes: Each node represents a gaming platform, labeled with the number of upcoming games as an attribute.

Edges: Simple connections are made sequentially from one platform to the next.

1. Six Degrees of Separation:

The concept of "Six Degrees of Separation" is illustrated quite effectively in the network connectivity among gaming platforms, as shown by an average path length of 3.33. This implies that in the network of gaming platforms, any two platforms can be connected through approximately three intermediary steps on average. This degree of interconnectedness suggests a closely-knit network where information or influence can potentially travel quickly from one platform to another due to the relatively short distances separating them.

In more detailed terms, the shortest path lengths between various pairs of platforms highlight both the direct and indirect connections that facilitate this small-world phenomenon. For instance, the direct connection between "PC Games" and "PlayStation 4 Games" represented by a single step indicates a strong, immediate link which could be due to similar or shared gaming libraries, cross-platform compatibility, or cooperative arrangements between these platforms.

On the other hand, the longest path in the network, consisting of 8 steps from "PC Games" to "Xbox Series X/S Games", although atypical, illustrates that even the most distant platforms are still relatively accessible to each other. This longer path could be indicative of fewer direct relationships or partnerships, possibly due to competing business models or different target markets. However, they remain part of the same network, connected through a series of intermediary platforms that bridge the gap between them.

These connections within the gaming platform network not only facilitate the movement of gamers from one platform to another based on their evolving preferences and technological upgrades but also allow for the spread of gaming trends, development techniques, and innovations across platforms. This interconnectedness enhances the robustness of the gaming industry, enabling it to adapt and grow by leveraging the strengths and specialties of different platforms.

In conclusion, the network analysis of gaming platforms with an average path length of 3.33 underscores a highly interconnected community where no platform is too isolated. This ensures a vibrant exchange of ideas, technology, and gaming experiences, which enriches the global gaming ecosystem and continually shapes the dynamics of game development and consumer engagement.

1. Degree Distribution and Centrality Measures:

Degree Distribution:

PC Games" and "Xbox Series X/S Games" are endpoints with a degree of 1.

All other platforms have a degree of 2, indicating they are connected to two other platforms.

Centrality Measures:

Closeness Centrality: Measures how close a node is to all other nodes in the network. "Switch Games" has the highest closeness centrality, indicating it is, on average, closest to all other nodes.

Betweenness Centrality: Indicates the extent to which a node lies on paths between other nodes. "Switch Games" also has the highest betweenness centrality, suggesting it acts as a critical bridge within the network.

1. Graph density:

Graph density offers insights into how closely nodes are linked within a network. It's calculated as the ratio of actual edges to possible edges. In a two-node network like "PlayStation 4 Games" and "PC Games," density calculation is simple, as the maximum possible edges is one.

A density of 1 between these two nodes signifies a perfect connection, indicative of a strong link with potential real-world implications for the gaming community and industry:

The strong link between "PlayStation 4 Games" and "PC Games" could enable cross-platform interaction, significantly enhancing user experience by facilitating connectivity across different platforms. Additionally, games released on both platforms might share development strategies and target demographics, which could ease transitions for gamers and foster a more integrated gaming environment. This connection could also reflect collaborative partnerships between developers, potentially leading to synchronized technological advancements and coordinated game releases. Marketers might leverage this data to tailor promotions and content strategically to appeal to a combined user base, optimizing outreach and engagement.

Recognizing such a dense connection within the gaming network can inform industry strategies, potentially sparking innovations that capitalize on this strong link to benefit both platforms and their users.

1. Conclusion:

The analysis of the gaming platform network using graph theory reveals essential insights into connectivity and strategic positions within the network. By implementing Breadth-First Search, we assessed connectivity and average path lengths, indicating close links between platforms. The examination of Degree Distributions and Centrality Measures helped identify the most interconnected and strategically important platforms. Furthermore, identifying the Densest Subgraph, particularly between "PlayStation 4 Games" and "PC Games," underscored robust connections that could significantly influence cross-platform collaboration and marketing strategies. These analyses collectively provide valuable insights for stakeholders aiming to optimize network interactions and strategic decisions in the gaming industry.

Dataset Link and citations:

1. , Clement. “Major Upcoming Video Games by Platform 2024.” Statista, 15 June 2023, [www.statista.com/statistics/1391589/major-upcoming-video-games-by-platform/.](http://www.statista.com/statistics/1391589/major-upcoming-video-games-by-platform/.)

"Exploringfly." CSDN, 26 Oct. 2017, blog.csdn.net/exploringfly/article/details/78652762.