Name: Samuel Apoya

Course: CS231 Section: A

Date: March 4, 2023

Project 08 Report

# TITLE: USING GRAPH DATA STRUCTURE TO IMPLEMENT THE PURSUIT EVASION GAME

**ABSTRACT**: This project implements graphs as data structure to make the *Pursuit Evasion game.* Different classes were made to maintain the edges and vertices of the graph. Inside these classes, methods were made to handle the movements of the pursuer and the evader. When both the pursuer and the evader lie on the same vertex, it means the pursuer has caught the evader and the pursuer wins.

**RESULTS**: When a method called "Driver.java" is run, a graph is drawn and as the evader tries to evade the pursuer(moves away from the pursuer with the help of a method called MoveAwayPlayerAlgorithm), the pursuer tries to catch the evader(moves towards the evader with the help of a method called "MoveTowardsPlayerAlgorithm").

Exploration

1.

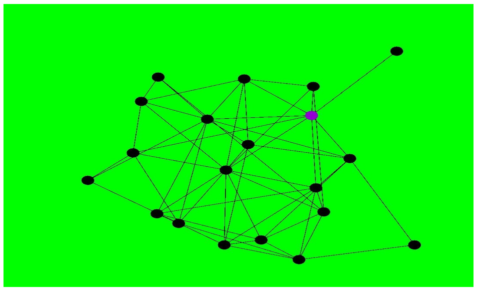

Figure 1. A complete cycle graph with an additional edge attached to it.

If the graph is almost a complete cycle, but has one additional vertex connected to one of the vertices in the circle - let's say we have 5 vertices in a circle (connected to each other) and one extra vertex is connected to ANY ONE of those 5 vertices. If the evader goes into the circle, he won't be caught, but if he goes into THAT ONE EXTRA VERTEX, the pursuer will be right behind him, and the evader will get caught. Figure 1. above illustrates this.

2.

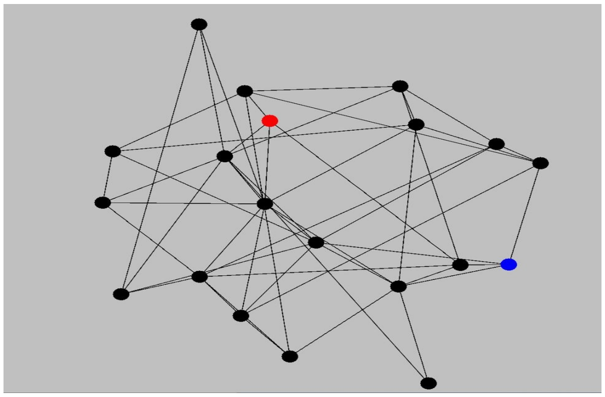

Figure 2. Screenshot showing a case where pursuer moves and pursuer does not move.

If the evader moves and the pursuer does not move, the evader will finally move into the pursuer and get caught. Figure 2 above illustrates this.

**REFLECTION**: A Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes referred to as nodes and the edges are lines that connect any two nodes in the graph. Formally, a Graph is composed of a set of vertices( V ) and a set of edges( E ).

ArrayLists were used to store vertices, edges, and adjacent edges because it's much simpler to implement and also faster in storing and accessing data as compared to a data structure such as LinkedList.

**EXTENSION**: A custom main method was created which allowed the user to enter the density and number of vertices. This ensures that no method is modified when the number of vertices and density need to be changed.

**REFERENCES**: Tutorialspoint - https://www.tutorialspoint.com/differences-between-arraylist-and-linkedlist-in-java#:~:text=ArrayList%20implements%20only%20List.,as%20a%20queue%20as%20well.&text=ArrayList%20is%20faster%20in%20storing%20and%20accessing%20data.

**ACKNOWLEDGEMENTS:** Najam Tariq