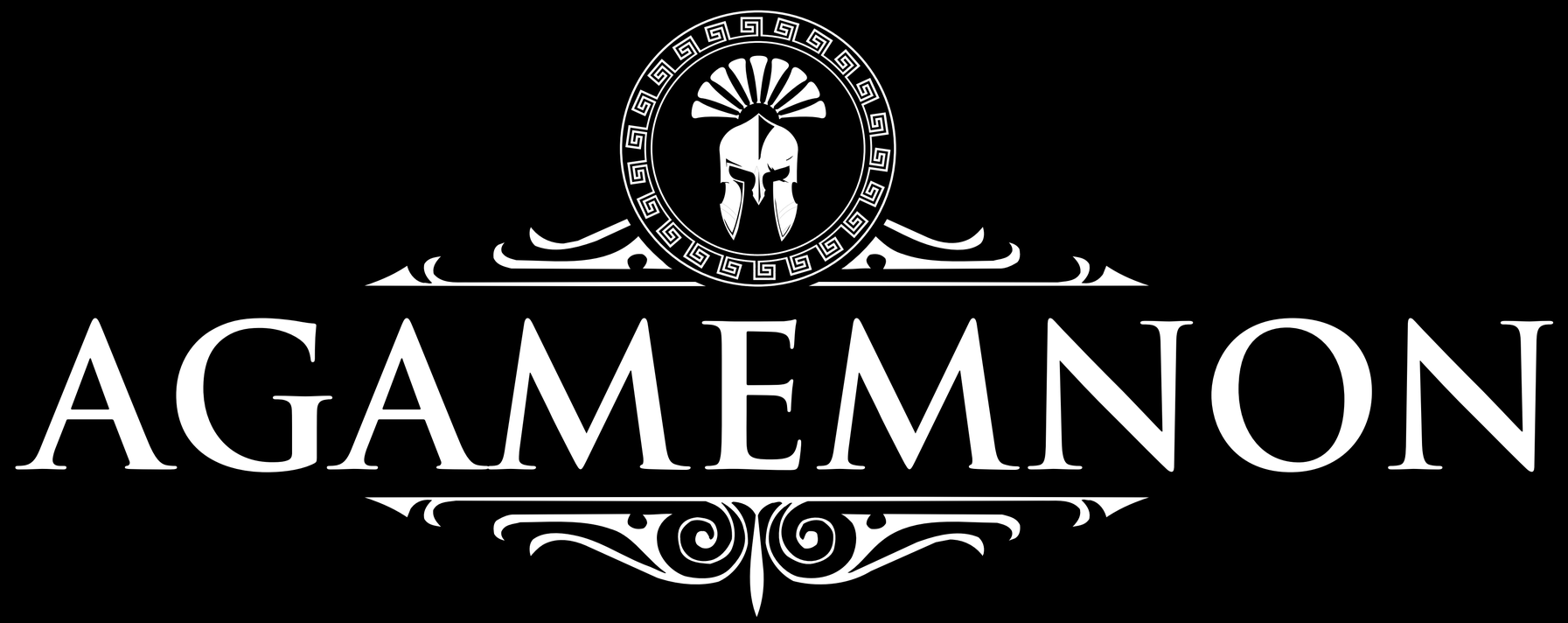
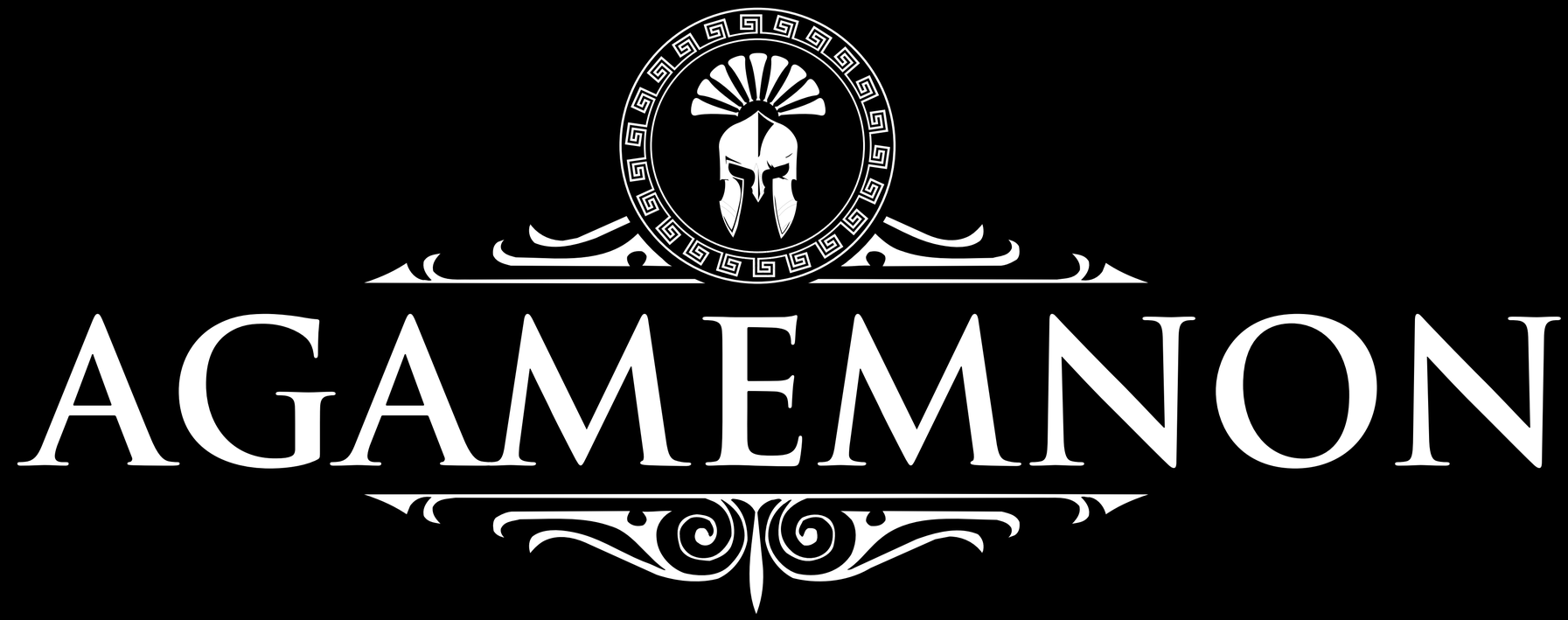

Agamemnon
File
Edit
Help
Difficulty
AGAMEMNON
PLAY GAME
PRESENTED BY
LEOPOLD & FANGXU
MacBook Pro

AGAMEMNON

Play Game


AGAMEMNON


AGAMEMNON

Software Designing And Human

Software Designing And Human

Leopold and FangXu Present

Software

Human

```java
    // inputs a tile and then find the paths(represented by connected edges) it forms with the edges.
    //* The core of this task *//
    public static List<ArrayList<String>> findPaths(ArrayList<String> list_tiles, ArrayList<String>
list_edges) {
        // create a list to store all the "paths"
        List<ArrayList<String>> general = new ArrayList<>();
```

```java
    // get the edges **every** tile connected to(except the tile "i")
        for (int x = 0; x <= list_tiles.size() - 1; x++) {
            if (list_tiles.get(x).charAt(1) == 'i') {
              // if it is tile "i", just put it with every edge it connects to
              // into a list, put them into the general list
                for (int y = 0; y <= list_edges.size() - 1; y++) {
                    if (tileIsConnected(list_tiles.get(x), list_edges.get(y))) {
                      // if the tile "i" connected to the edge
                        ArrayList<String> i = new ArrayList<>();
                        i.add(list_tiles.get(x));
                        i.add(list_edges.get(y));
                        general.add(i);
                      // Put this edge with the "i" itself into the list,
                      // put the list into the general list.
                    }
                }
            }

            if (list_tiles.get(x).charAt(1) != 'i') {
                ArrayList<String> a = tile_connect_edges(list_tiles.get(x),list_edges);
                if (a.size() != 0) {
                    general.add(a);
                // put the tile into the list for counting scores
                    a.add(list_tiles.get(x));
                }
            }
        }
```

```java
        return general;
    }
```

```java
        for (int x = 0; x <= general.size() - 1;x ++) {
            for (int y = x + 1; y <= general.size() - 1; y++) {
              //get two paths and find if they are connected
                for (int index_1 = 0; index_1 <= general.get(x).size() - 1; index_1++) {
                    for (int index_2 = 0; index_2 <= general.get(y).size() - 1; index_2++){
                        if (general.get(x).get(index_1).length() > 4
                            && general.get(y).get(index_2).length() > 4){
                          // avoid compare tiles
                            if (general.get(x).get(index_1).equals(general.get(y).get(index_2))
                                && (x != y)) {
                              // if any two edges are repeated, means the two "paths" are connected
                                general.get(x).addAll(general.get(y));// combine the two connected
"paths"
                                general.remove(y);
                                Set<String> edgesSet = new LinkedHashSet<>(general.get(x));
                                general.set(x, new ArrayList<>(edgesSet));// remove the repeating edges
                                y = x + 1;// begin at the new list's next list
                                index_1 = 0;// begin at the new list's first element
                                index_2 = -1;// after combination, the ge.get(y) is new list , begin at
this list's first element.
                            }
                            if (general.size() == 1 || y == general.size()) return general;// if there
is only one "path" or the last two lists combine, just return
                        }
                    }
                }
            }
        }
```

Steve Jobs

“Finally, it’s about human.”

Collaboration

Team

What is a partner?