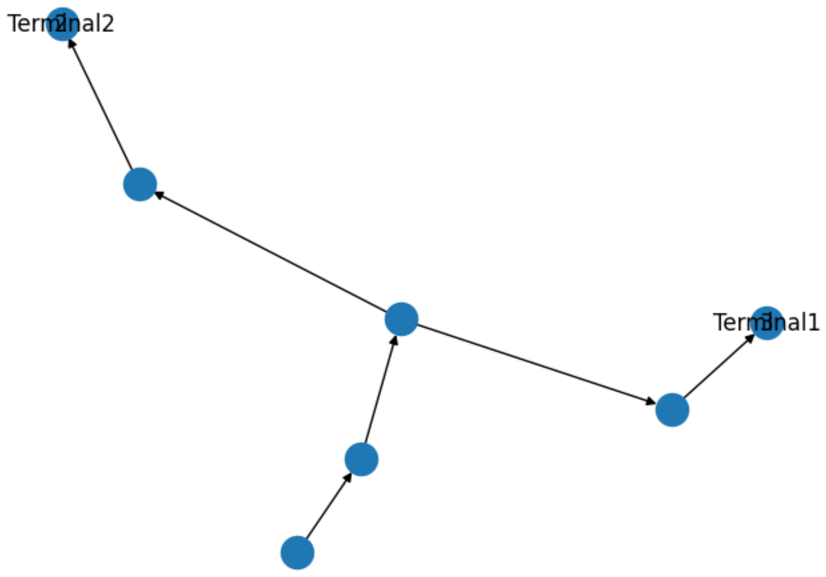
```
import networkx as nx
import matplotlib.pyplot as plt


def create_game_tree():
    # Create an empty directed graph
    G = nx.DiGraph()

    # Add root node
    G.add_node("Root")

    # Add player 1's decision nodes and edges
    G.add_node("A1")
    G.add_edge("Root", "A1")

    G.add_node("A2")
    G.add_edge("A1", "A2")

    # Add player 2's decision nodes and edges
    G.add_node("B1")
    G.add_edge("A2", "B1")

    G.add_node("B2")
    G.add_edge("A2", "B2")

    # Add terminal nodes and edges
    G.add_node("Terminal1", utility=3)
    G.add_edge("B1", "Terminal1")

    G.add_node("Terminal2", utility=2)
    G.add_edge("B2", "Terminal2")

    return G


def visualize_game_tree(G):
    pos = nx.spring_layout(G)
    labels = {node: node if "Terminal" in node else "" for node in G.nodes()}

    utilities = {node: G.nodes[node].get("utility", "") for node in G.nodes()}

    nx.draw(G, pos, with_labels=True, labels=labels)
    nx.draw_networkx_labels(G, pos, utilities)

    plt.show()


if __name__ == "__main__":
    game_tree = create_game_tree()
    visualize_game_tree(game_tree)
```


Terminal2
2
Terminal1
3