Adventure Game
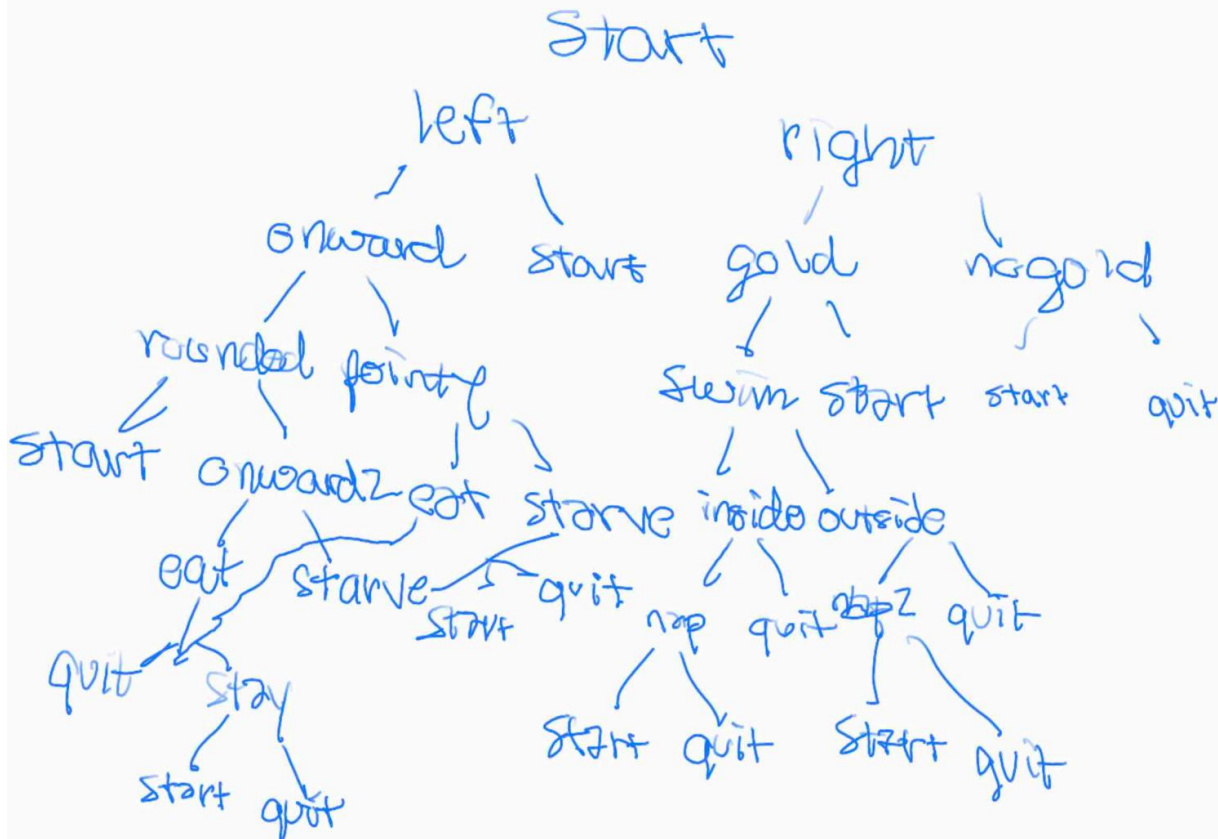
Pseudocode/Documentation

Visual diagram:

define function "getGame":

    pathGame = {

        "start": ["You find yourself facing a fork in the road.", "Go left", "left", "Go right", "right"],

        "left": ["As you stroll, you begin to feel itchy. You quickly realize that this path is full of poison oak.", "Keep going", "onward", "Start over", "start"],

        "right": ["Oh no! A witch has appeared in front of you. She requires 50 gold pieces to pass.", "Give her 50 gold pieces", "gold", "Walk around the witch, pretending not to see her", "noGold"],

        "onward": ["You remember an old pioneer trick your grandma taught you. Rub a specific type of leaf on your rash to clear it up. The problem is... you can't remember

which type. Were the leaves rounded or pointy?", "Rounded", "round", "Pointy", "point"],

    "gold": ["She moves out of the way for you. You can't get much further, though, because there is a river cutting through the pathway.", "Swim across the river", "swim", "Start over", "start"],

    "noGold": ["You made her angry. As you walk ahead of her, she casts a fireball spell in your direction, burning you to cinders. You lose!", "Start over", "start", "Quit", "quit"],

    "round": ["You just rubbed another poison oak leaf onto your rash.", "Start over", "start", "Try again to cure the rash", "onward2"],

    "point": ["With pure, dumb luck, your rash clears up. You're getting hungry, and see a berry bush in the distance.", "Eat the berries", "eat", "Starve", "starve"],

    "swim": ["The water is cold and uninviting. Ahead of you is a warm looking cottage sitting along the river's shoreline.", "Go inside", "inside", "Stay outside", "outside"],

    "onward2": ["You try again with the pointy leaf and, by some miracle, your rash is cured. Now you're hungry, and there is a berry bush in the distance.", "Eat the berries", "eat", "Starve", "starve"],

    "eat": ["The berries are plump and juicy. You have almost reached the end of the path, and it's getting dark. ", "Turn around and go home", "quit", "Stay into the night and keep eating berries", "stay"],

    "starve": ["You are too hungry to continue. You lose!", "Start over", "start", "Quit", "quit"],

    "inside": ["The inside of the cottage is dimly lit by a fireplace in the corner of the room. Nobody is home, but there are plenty of supplies left for you to use.", "Take a nap by the fireplace", "nap", "Turn around and go back home", "quit"],

    "outside": ["The night is cold, dark and humid. ", "Take a nap on the shoreline", "nap2", "Turn around and go back home", "quit"],

    "stay": ["You are too full of berries to continue down the path.", "Start over", "start", "Quit", "quit"],

    "nap": ["You awaken to the smell of cookies in the oven. As it turns out, this is your grandma's vacation home, and she really missed you. You win.", "Start over", "start", "Quit", "quit"],

"nap2": ["Your grandma wakes you and asks you why you are sleeping on the shoreline. She invites you into the cottage nearby, which turns out to be her vacation home. She dries you off and tucks you into bed. You win.", "Start over", "start", "Quit", "quit"],

 }

parameters: none

generates dictionary (shown above: pathGame)

return pathGame

def function "playNode":

parameters: pathGame, currentKey

look up node data from pathGame

create variable "currentNode"; "currentNode" looks up the node from the currentKey (pathGame[currentKey])

(description, menuA, nodeA, menuB, nodeB) = currentNode

print the menu

call {description}

1) {menuA}
2) {menuB}

create variable "choice"; "choice" gets user input (their choice from the menu, 1 or 2)

if choice == "1":

nextKey == nodeA

if choice == "2":

nextKey == nodeB

else:

nextKey == currentKey

return nextKey

define function "main":

        call getGame to get pathGame

        set currentKey to "start"

        use while keepGoing to iterate through pathGame dictionary

        set keepGoing to True

        while keepGoing:

                if currentKey gets "quit",

                        set keepGoing to False (exit loop)

                else:

                        send game and currentKey to playNode function

                        currentKey gets result of playNode

main()