PROJECT DOCUMENT – FOR CCPROG1 MP

BRENT JAN F. SOAN

CCPROG1 – S16

EXPLANATION FOR THE DESIGNING OF THE PROGRAM

Before starting to code the program, I focused on separating how things should be. Like, I first started with the loop of printing the selection and the other things needed within the loop. Next was how I would make it look like an actual game. I used ascii text art for this so that it wouldn’t look bland. As for the functions, I did the easiest first up to the hardest because the other two functions needed the rand() function, which was the first time that I used it.

EXPLANATION FOR THE FUNCTIONS

The main function would work because I noticed that coding the functions was going to be tricky. The main function has variables containing what values the pots should have at the beginning. In the beginning, it will ask the user if it wants to start (go to the next part) or exit by pressing either 1 or 0. Entering 1 will print the instructions of the game and how it works, sending you to a loop wherein there are loops and conditions that call the function that is responsible for pouring the pots and the function that randomizes the value for the rain to fill up pots. The main loop only ends when a pot's value is -1.

The fullOrNot function from the name itself determines what pot that isn't full should be chosen. In the loop, there is a variable that gets the value of the rand function, which picks a pot and goes through a series of if-else statements. If the pot is full, then the function x will be 0, and if it is not, then x will be 1, ending the loop.

The pourPot function is for pouring gallons of water from one pot to another. The first three parameters of the function are the gallons in each pot; the other parameters are selecting the pot that will be the pot that will pour onto the chosen pot and vice versa. In the function, there are three if-else statements that are for what pot will be used to fill the other pot, and inside each of the if-else statements there is a switch statement. Each case has an if statement indicating whether the pot to pour has gallons or not. If it is satisfied, then it will proceed to a loop that is responsible for adding value to the pot being poured into. Then the variable j will subtract what value was added to the pot if only the cT is equal to its maximum. The values of each pot will be sent to the main function as their corresponding variable.

The randRain function is responsible for filling up the pots from 1 to 4. Parameters of the function get the values of the three pots. At the start of the function, the srand function will initialize the random number generator algorithm with the time function, so that will start from 0 and continue going up. The fullOrNot function will be called to pick what pot shall be chosen to be filled up. Then the rand function is going to set how many gallons from 1 to 4 are going to fill the pot chosen. The next step is to go through a switch case. It'll be the most important part of the function because of the complexity of what is needed. There are a lot of nested loops, addition, and subtraction; it also calls the fullOrNot function to pick a pot different from the one that was previously chosen.

TEST SCRIPTS

Function	#	Description	Sample Input Data	Expected Output	Actual Output	P/F
pourPot	1	Pour 3rd pot to 1st pot.	*pPot_1: 0 *pPot_2: 0 *pPot_3: 9 cF: 3 cT: 1	POT 1: 4 POT 2: 0 POT 3: 5	POT 1: 4 POT 2: 0 POT 3: 5	P
	2	Pour 3rd pot to 2nd pot.	*pPot_1: 0 *pPot_2: 0 *pPot_3: 9 cF: 3 cT: 2	POT 1: 0 POT 2: 2 POT 3: 7	POT 1: 0 POT 2: 2 POT 3: 7	P
	3	Pour 1st pot to 3rd pot.	*pPot_1: 0 *pPot_2: 0 *pPot_3: 9 cF: 1 cT: 3	POT 1: 0 POT 2: 0 POT 3: 9	POT 1: 0 POT 2: 0 POT 3: 9	P
	4	Pour 2nd pot to 1st pot.	*pPot_1: 0 *pPot_2: 0 *pPot_3: 9 cF: 2 cT: 1	POT 1: 0 POT 2: 0 POT 3: 9	POT 1: 0 POT 2: 0 POT 3: 9	P
	5	Pour 1st pot to 3rd pot.	*pPot_1: 4 *pPot_2: 2 *pPot_3: 3 cF: 1 cT: 3	POT 1: 0 POT 2: 2 POT 3: 7	POT 1: 0 POT 2: 2 POT 3: 7	P
fullOrNot	1	Picks a pot that isn't full.	nGal1: 4 nGal2: 2 nGal3: 3	3	3	P
	2	Picks a pot that isn't full.	nGal1: 3 nGal2: 2 nGal3: 5	1 or 3	3	P
	3	Picks a pot that isn't full.	nGal1: 0 nGal2: 0 nGal3: 9	1 or 2	2	P
	4	Picks a pot that isn't full.	nGal1: 0 nGal2: 0 nGal3: 5	1 or 2 or 3	2	P
	5	Picks a pot that isn't full.	nGal1: 0 nGal2: 2 nGal3: 9	1	1	P
randRain	1	Rain on the pots from a range of 1 to 4 gallons.	*pP1: 0 *pP2: 2 *pP3: 7	POT 1: -1 to 4 POT 2: 1 or 2 POT 3: 6 to 9	POT 1: 1 POT 2: 1 POT 3: 6	P
	2	Rain on the pots from a	*pP1: 0 *pP2: 0	POT 1: -1 to 4 POT 2: -1 to 2	POT 1: 1 POT 2: -1	P

		range of 1 to 4 gallons.	*pP3: 8	POT 3: 7 to 9	POT 3: 9	
	3	Rain on the pots from a range of 1 to 4 gallons.	*pP1: 4 *pP2: 0 *pP3: 6	POT 1: 3 or 4 POT 2: -1 to 2 POT 3: 5 to 9	POT 1: 3 POT 2: -1 POT 3: 7	P
	4	Rain on the pots from a range of 1 to 4 gallons.	*pP1: 2 *pP2: 1 *pP3: 2	POT 1: 1-4 POT 2: 0-2 POT 3: 1-6	POT 1: 1 POT 2: 0 POT 3: 5	P
	5	Rain on the pots from a range of 1 to 4 gallons.	*pP1: 2 *pP2: 2 *pP3: 5	POT 1: 1-4 POT 2: 1 or 2 POT 3: 4 to 9	POT 1: 3 POT 2: 1 POT 3: 4	P