ASGN 1 ALI SALEHI

GAME LOGISTICS

- 2-10 PLAYERS
- 3$ STARTING AMOUNT
- ROLL 1 DICE PER DOLLAR OWNED (MAX 3)
  - PASS IF NO MONEY

· 3 DICE } EACH DICE
50% •
$16.\bar{6}\%$ L, R, C

- MOVE -> NEXT PLAYER TO THE RIGHT
- LEFT PLAYER = (you-1) % (# players)
- RIGHT PLAYER = (you + 1) % (# players)

PROGRAM STRUCTURE:

- rand() For RANDOM #'s —FOR DICE ROLL
- srand() CREATES RANDOM SEED + INTAKES # OF PLAYERS
- INFINITE LOOP -> BREAKS

1: SETUP

- GENERATE SEED -# OF PLAYERS IN?
- INITIALIZE ARRAY OF PLAYERS
- DETERMINE # OF PLAYERS + RANDOM NUMBERATOR
- INITIALIZE BANK ARRAY
- CREATE POT FOR MONEY
- INITIALIZE DICE ARRAY

HELPER FUNCTIONS

1) FIND LEFT PLAYER

2) FIND RIGHT PLAYER

3) TAKE TURN

↳ ROLL X DICE (X BASED ON BANK)

→ ROLL DIE w/ PARAMETER RUNNING MORE OR LESS

END CONDITION

- LEAVE INFINITE LOOP
- PRINT WINNER + pot money + amount in bank

- USE names[# % player_num] TO ACCOUNT FOR VARYING # OF PLAYERS
  - USE SAME NUMBER FOR BANK

PSUEDO CODE

```
- ALL THE VARIABLE DECLARATIONS & FUNCTION
  PROTOTYPES HERE        * DO PLAYERNUM ERROR CHECK

MAIN {
    - PRINT PROMPTS AND TAKE IN USER INPUT

    WHILE(1){
        - PRINT CURRENT PLAYER MESSAGE
        - CHECK IF CURRENT PLAYER IS IN GAME (IF SO:)
            - TAKE TURN():
            - LOOP X # OF TIMES (X = 3 IF $ >3 ELSE $)
        *     — ROLL DICE — FOLLOWED BY IF BRANCH
                  - DO MONEY/BANK ADJUSTMENTS
                  - PRINT RESULT (SPACE FIRST)
                  -> PRINT A NEW LINE
        - ELSE:
            - CHECK IF GAME SHOULD END
                - IF SO BREAK LOOP + PRINT END MES.
            - INCREMENT PLAYER POSITION + RE-LOOP.

if( die[rand() % 6] == LEFT) {
    - left prompt
} ELSE IF (die[rand() % 6] == RIGHT) {
    - RIGHT PROMT
} ...
```