Asgn1 Design

Diagram:

Declare Positions (enum)
↓

ask user for # players
↓

if # of player >10 or <2:
    ↙   ↓

  yes   no
  ↓   ↓

invalid   valid
 ↓   ↓

use 2   use #
  ↓   ↙

ask user for seed
     ↓

if seed is valid
  ↙   ↓

  yes   no
  ↓   ↓

valid   invalid
 ↓   ↓

use Seed   use 2021
    ↘   ↙

    Points = 0
    ↓

Player # rolls (while POINTS<101 ∨ $\overline{SIDE}$)
↓   ↓

no   yes
↓   if (POINTS >=100)
roll   ↓
↓

Check points   printf("%s wins w/ %d points", name, POINTS)
↓   break / return 0
if (points >=100)   ↓
↓   else: Next player
printf("%s wins w/ %d points", name, POINTS)
break / return 0

Functions: Roll / Check points / Player turns

```
* Roll() {
    num = random()
    rollSide = pigs[num]
    add points = points[num]

    score += addpoints
    printf(" %s rolled a %s  & scored %d", name, roll side, addpoints)
    totalscore[Player] = score



checkscore () {
    if (score >= 100) {
        printf("%s has won  w/ %d points", name, score)
        break
    else
        return
```

Array: totalScore / Points

```
int  totalScore [numPlayers];

int  points[7] = { 0, 0, 10, 10, 15, 5, 5 }
```

Brainstrom:

- How to switch between players
- Pull from Array & add to array