CIS214 Chad Botting
Fall 2012
Lab1 Fun with Probability
November 8, 2012

Top Down, Step Wise Refinement with pseudo code

The Top Statement:

To generate statistics to solve the Monty Hall problem

The First Refinement:

Play the game 1000 times and store results without switching doors

Play the game 1000 times and store the result with switching doors

Give statics of each run

The Second Refinement:

```
FUNCTION MainLine
      boolean withOutChange[1000] // ini to false
      boolean withChange[1000]     // ini to false
      boolean doors[3]

      DO (x = 1 TO withOutChange.ArrayLength)
            placePrizes(doors)
            withOutChange[x] = playGame(doors, FALSE)
      END DO

      DO (x = 1 TO withChange.ArrayLength)
            placePrizes(doors)
            withChange[x] = playGame(doors, TRUE)
      END DO

      PRINT "Chances without changing: " + countPrizes(withOutChange) + "%"
      PRINT "Chances with changing   : " + countPrizes(withChange) + "%"

END MainLine

FUNCTION placePrizes(boolean[] doors)
      DO (x = 1 TO doors.ArrayLength)
            doors[x] = FALSE
      END DO
      carDoor =
      doors[carDoor] = TRUE
END placePrizes
```

```
FUNCTION playGame(boolean doors[], boolean changeDoor)
      playerChoice = random interger 1 from 3
      revealedDoor = -1

      DO(x = 1 TO doors.ArrayLength)
            IF ((NOT doors[x]) AND (x NOT EQUAL playerChoice)) THEN
                  revealedDoor = x
                  break
            END IF
      END DO

      IF (changeDoor) THEN
            randomNumber = -1;
            originalChoice = playerChoice
            DO UNTIL ((randomNumber NOT EQUAL originalChoice) AND
(randomNumber NOT EQUAL revealedDoor))
                  randomNumber = random interger 1 from 3
            END DO
            playerChoice = randomNumber
      END IF

      RETURN doesPlayerWin(playerChoice, doors)
END playGame

FUNCTION doesPlayerWin(playerChoice, boolean doors[])
      IF(doors[playerChoice)) THEN
            RETURN TRUE;
      ELSE
            RETURN FALSE;
      END IF
END doesPlayerWin

FUNCTION countPrizes(boolean wins[])
      int winsCount = 0;
      DO(x = 1 TO wins.ArrayLength)
            IF (wins[x] = TRUE) THEN
                  winsCount = winsCount + 1
            END IF
      END DO
      RETURN (winsCount/wins.ArrayLength) * 100.0
END countPrizes
```