|  |  |  |
| --- | --- | --- |
| Functions | Variables | Comment |
| **Game function**  print initial text & short rules  run **print table function**  ask for game mode (1 player vs. pc or 2 players)  while end\_game = false  while (input\_valid = false) or (end\_game=false)  run **input function**  if end\_game = false  run input **validity check function**  if end\_game = false  run **implement move on table function**  run **print table function**  run **check winner function**  if end\_game = true  print “actual player has won”  else  set actual\_player = other  set input\_valid = false  else  print “actual\_player has ended game”  game ends here | game\_mode = 1/2  end\_game = false  input\_valid = false  actual\_player = 1  input = <intitialize>  table = numpy array  headers = list  dict = {a:0, b:1….g:6}  -> input = value   or end\_game = true  -> input\_valid = T/F  -> end\_game = T/F  -> actual\_player = 1/2  -> input\_valid = false | **Test**  **Test** |
| **Print table function**  print(header)  print(table) |  | separate function might not be needed |
| **Input function**  if game\_mode = 1 and actual\_player = 2  input = random letter a - g  else:  print ‘actual player enter the column where you drop your stone or x to end game‘  temp\_variable = str(input())  if temp\_variable = x  end\_game = true  else  input = temp\_variable | -> input  -> end\_game = T/F  -> input | here would be the place to add intelligent input from pc |
| **Validity check function**  if input in dict = false  print ‘no such column, pick a letter between a – g’  elif top item in the column(input) not = 0  if (game mode = 1 and actual player = 1) or game\_mode = 2  print ‘column already full, pick another between a-g ’  else  input\_valid = true | -> input\_valid = T/F |  |
| **Implement move on table function**  column index = dict[input]  check which is the lowest row in array, where in this column the value is still 0  change this value = actual player |  |  |
| **Check winner function**  set winning sequence based on actual player (1111 or 2222)  check all the rows in array if they contain winning sequence  check all the columns - “ –  check all diagonals – “ –  flip the table and check all reverse diagonals – “ –  if any of those true:  end\_game = true | -> end\_game = T/F |  |