Department of Computing

COMP1127 – Introduction to Computing II

The Chase

"The Chase" is a game played with two dice and an even number of players. The players sit around a table and the game begins with two players sitting opposite to one another having one dice each. On each turn, the two players with dice roll them. If a player rolls a 1, she passes the dice to her neighbour on the left; if she rolls a 6, she passes the dice to her neighbour on the right; otherwise, she keeps the dice for the next turn. The game ends when one player has both dice after they have been rolled and passed; that player has then won. We will write the code that simulates the game play, so let's begin. The fundamental component of the game is the dice. Let's represent a dice as an abstract data type (ADT). Complete the following,

1. Write a function called `roll` that generates and returns a random integer between 1 and 6 inclusive. `roll` does not accept any arguments. For example,

```
>>> roll()
6
>>>
```

2. Write an ADT for a dice using the following representation,

```
('dice',[])
```

and the functions (their type signatures are listed below),

- constructor: `makeDice -> Void -> dice`
  Takes no arguments and returns a dice and its current face value. Its current face value is determined by calling `roll` and assigning the returned value to the dice ADT's list element. For example,

```
>>> dice1=makeDice()
>>> dice1
('dice', [4])
>>>
```

- predicate: `isDice -> dice -> Boolean`
  Accepts a dice and returns `True` if the argument is a tuple, the argument's first element is the tag 'dice' and its second element is a list. For example,

```
>>> isDice(dice1)
True
>>> isDice('dice')
False
```

- selectors: `face -> dice -> List`

  Accepts a dice and returns its list. Note, a dice's list should always have only one element. For example,

  ```
  >>> face(dice1)
  [4]
  >>>
  ```

  `facevalue -> dice -> Integer`

  Accepts a dice and returns its face value. For example,

  ```
  >>> facevalue(dice1)
  4
  >>>
  ```

- mutator: `rollDice -> dice -> Void`

  Accepts a dice and simulates rolling that dice by generating a random number between 1 and 6 and changing the stored face value of the dice. For example,

  ```
  >>> rollDice(dice1)
  >>> dice1
  ('dice', [3])
  >>>
  ```

You must include checks that ensure arguments passed to the ADT functions are of type `'dice'`

3. We next need to write a function called `displayAndRoll`. This function accepts two players and two dice and displays which players have which dice. It then rolls both players' dice and displays what the new face value is. For example,

```
>>>
Players 0  and  2  each have a dice. Going to roll each of them
Dice 1: Player  0  has rolled  4
Dice 2: Player  2  has rolled  3
...
```

4. Write a function called `passDice` that accepts the face value of a dice. If the face value is either 1 or 6 it returns `True`, otherwise it returns `False`. For example,

```
>>> passDice(1)
True
>>> passDice(4)
False
>>>
```

5. Now write a function called `nextPlayer`. This function accepts a player, a face value and the total number of players as arguments. Based on the face value and the game's rules, it passes the dice to another player i.e. if the face value is 1 the dice is passed to the player to the left of the current player, if the face value is a 6, the dice is passed to the player to the right of the current player, otherwise the current player keeps the dice. For example,

```
>>> nextPlayer(3,1,4)
2
>>> nextPlayer(3,6,4)
0
>>> nextPlayer(3,3,4)
3
>>>
```

6. We can now put things together so that the gameplay can be simulated. The `play` function does not accept any arguments. The gameplay is simulated as follows:
   - two dice, `dice1` and `dice2`, are created
   - the user is asked to enter the total number of players. `play` checks that the number entered is even. It does not continue until an even number has been entered
   - two players, in opposite positions on the table are initially assigned the dice
   - while one player is not in possession of both dice, the function does the following:
     - it displays and rolls the players and the dice's face values
     - if either dice must be passed to another player, the new player is determined and the dice passed on. If a dice is passed to a new player, the user is informed
     - after each round, the user is offered the option to quit by entering the letter `q`, or continuing by pressing any key other than `q`
   - play ends, as does the game, when one player is in possession of both dice.

   A run of the game is shown below,

```
>>> play()
How many players (even number only)?4
Players 0  and  2  each have a dice. Going to roll each of them
Dice 1: Player  0  has rolled  2
Dice 2: Player  2  has rolled  1
...
Player  1  now has dice 2
press any key to roll again or 'q' to stop
Players 0  and  1  each have a dice. Going to roll each of them
Dice 1: Player  0  has rolled  4
Dice 2: Player  1  has rolled  1
...
Player  0  now has dice 2
press any key to roll again or 'q' to stop
Player 0  has won
>>>
```

That's it. By the time you have finished, you will have a fully functioning simulation. There are some improvements that can be made, for example, it can be enhanced to be completely graphical. ***N.B.*** throughout this assignment, no abstractions are to be violated. If you do so, marks will be deducted.

- This assignment is worth 15% of your coursework mark
- You are to work in pairs
- Ensure both team member's ID numbers are included in the submitted code
- Submit well documented, original code using the container on OurVLE
- Name your file by concatenating the ID# of the person that submitted the code prefixed with the characters ”A1127”. For example: A1127_62001234.py
- The deadline for submission is April 11, 2014 at 11pm

Have fun!