

This worksheet has two aims - to give you your first experience of working in a team, and to strengthen your skills with writing functions in R. These two aims will be achieved through your group working together to produce one or more short games, playable using R.

Two New Functions

I showed you how the **while** and **for** functions worked earlier in the workshop, but here is a little more information on them which you might find useful when designing your game.

1. The function **for** performs an operation a number of times; the number of times has to be specified. We specify through the use of a counter, often but not necessarily denoted **i**. We refer to this process as “a **for** loop”. If we try:

```
a<-0
```

```
for(i in 1:10){a<-a+i}
```

we get a value **a**, initially equal to 0, but then has the values 1, 2, 3,... 10 added to it, for a total of 55. In contrast, if we try

```
a<-0
```

```
for(i in 1:10){a<-a+1}
```

we'll get a value for **a** of 10. This is because **i** is not contained in the **for** loop. R still knows how many times you want the loop to run for, though, so it still does what you ask it to 10 times.

2. The **while** function has some similarity with the **for** function, in that it runs loops for us. The big difference is that the **for** function needs you to specify in advance how many loops to run. The **while** function will keep running loops so long as a given property holds.

So for instance, we can write a function that tells us how many times we can divide an integer by 2, as we see on the next page.

```
divideby2<-function(x){  
  count<- -1  
  status<-1  
  while(status==1)  
  {count<-count+1  
    if(x%%2!=0){(status<-0)}  
    x<-x/2  
  }  
  return(count)  
}
```

Here, the **while** loop keeps dividing the input by 2, until such time as that doesn't give us a whole number. At that point the program exits the **while** loop. By adding one to the count each time the **while** loop runs, we learn how many times dividing by two was possible.

Gang And Game

Okay! It's time to design a game. I'll start this section by explaining what I want each group to produce, and after that, I've given some examples of code you can use to construct a game to play in the Workshop 2 folder on Ultra - feel free to use parts of this code in your own game. I'll leave it up to each group as to how simple or complicated you want your game to be. Bear in mind the benefits of starting with something simple and making it more complicated once you're sure the simple version works!

There are three primary components to produce in this exercise:

1. A name for your game.
2. A word-processed document which gives the rules to the game.
3. R code to be used in playing the game. The game does not have to be completely playable in R (the first example I give below just uses R to generate dice rolls), but at least some part of your game should be playable using R code that you have written.

It is up to your group to decide what the game involves, how complicated to make it, whether it should come in multiple stages (perhaps modular programming would be a good idea here). While this is a good opportunity to apply what you've learned so far about R, the primary focus of this task is taking your first steps in learning to work together. Discuss the task, and come to an agreement on what you want to try. Decide who will participate in each task and how, and how to keep each other informed of how things are going.

Remember what we just learned about teams vs groups. Will each of you have a different task, or will some/all tasks be done together? Do you want more than one person attempting to code something, so they can help each other, or are people going to work on different parts of code, in order to bring it together at the end. What will you do if someone gets stuck?

I won't be asking you to submit your game, but I **will** be releasing a short reflective exercise at 6pm today for you to go through. This exercise will not be for credit, but it is an expectation that you complete the exercise before the next workshop on 19.10.23.