Title: **Racket Assignment #1: Getting Acquainted with Racket/DrRacket + LEL Sentence Generation**

Abstract

This assignment helped acquaint myself with Racket and the DrRacket IDE by reproducing an lel sentence generator. The sentence generator introduced some useful syntax like append, list, and map. Once the lel sentence generator was written a short demo was concluded and is included in this assignment.

Code for the LEL Sentence Generator

```
#lang racket

;---------------
; LEL sentence generator, with helper PICK,
; several applications of APPEND, several
; applications of LIST, and one use of MAP
; with a LAMBDA function.

( define ( pick list )
   ( list-ref list ( random ( length list ) ) )
)

( define ( noun )
   ( list ( pick '( robot baby toddler hat dog ) ) )
)

( define ( verb )
   ( list ( pick '( kissed hugged protected chased hornswoggled ) ) )
)

( define ( article )
   ( list ( pick '(a the ) ) )
)

( define ( qualifier )
   ( pick '( ( howling ) ( talking ) ( dancing )
             ( barking ) ( happy ) ( laughing )
```

```
            () () () () () ()
          )
  )
)

( define ( noun-phrase )
  ( append ( article ) ( qualifier ) ( noun ) )
)

( define ( sentence )
  ( append ( noun-phrase ) ( verb ) ( noun-phrase ) )
)

( define ( ds ) ; display a sentence
  ( map
    ( lambda ( w ) ( display w ) ( display " " ) )
    ( sentence )
  )
  ( display "" ) ; an artificial something
)
```

Demo for the LEL Sentence Generator

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'red
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( Racket Prolog Haskell Rust ) )
'Prolog
> ( pick '( Racket Prolog Haskell Rust ) )
'Prolog
> ( pick '( Racket Prolog Haskell Rust ) )
'Racket
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( noun )
```

'(dog)
> (noun)
'(toddler)
> (noun)
'(robot)
> (noun)
'(hat)
> (verb)
'(hornswoggled)
> (verb)
'(hornswoggled)
> (verb)
'(kissed)
> (verb)
'(protected)
> (article)
'(the)
> (article)
'(a)
> (article)
'(the)
> (article)
'(a)
> (qualifier)
'(howling)
> (qualifier)
'(happy)
> (qualifier)
'(dancing)
> (qualifier)
'()
> (qualifier)
'(talking)
> (qualifier)
'()
> (qualifier)
'(dancing)
> (qualifier)
'(barking)
> (qualifier)
'()
> (qualifier)
'(howling)
> (qualifier)

'()
> (qualifier)
'()
> (qualifier)
'(laughing)
> (qualifier)
'(barking)
> (qualifier)
'()
> (qualifier)
'(happy)
> (noun-phrase)
'(a baby)
> (noun-phrase)
'(a dancing baby)
> (noun-phrase)
'(the toddler)
> (noun-phrase)
'(the happy dog)
> (noun-phrase)
'(a happy robot)
> (noun-phrase)
'(the talking toddler)
> (noun-phrase)
'(a laughing dog)
> (noun-phrase)
'(the toddler)
> (sentence)
'(a happy robot kissed a barking toddler)
> (sentence)
'(the howling dog hugged the talking dog)
> (sentence)
'(a laughing dog protected a happy dog)
> (sentence)
'(the happy toddler chased the baby)
> (sentence)
'(a toddler kissed the dog)
> (sentence)
'(the happy toddler hugged a barking dog)
> (sentence)
'(a robot protected the hat)
> (sentence)
'(a robot protected the howling toddler)
> (ds)

a laughing dog hugged a hat
> (ds)
the happy hat hugged a dog
> (ds)
the baby kissed a laughing baby
> (ds)
the laughing hat kissed the robot
> (ds)
a robot hornswoggled the happy hat
> (ds)
the toddler kissed a happy baby
> (ds)
a happy dog chased the howling dog
> (ds)
the howling baby protected the happy robot
> (ds)
the hat hornswoggled the happy dog
> (ds)
a toddler protected the dancing dog
> (ds)
the toddler hornswoggled a happy baby
> (ds)
the dog protected the laughing dog
>