Sam Ghent

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

Abstract:

The goal of this assignment was to mindfully type code in DrRacket in order to understand the basics of the Racket programming language. We typed code the created a sentence generator and took notice of how our actions effected the output of the program.

Code:

```
#lang racket
;------------------------------------------
; LEL sentence generator, with helper PICK,
; several applications of APPEND, several
; application 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:

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

```
> ( pick '( red yellow blue ) )

'yellow

> ( pick '( red yellow blue ) )

'blue

> ( pick '( Racket Prolog Haskell Rust ) )

'Racket

> ( pick '( Racket Prolog Haskell Rust ) )

'Prolog

> ( pick '( Racket Prolog Haskell Rust ) )

'Racket

> ( pick '( Racket Prolog Haskell Rust ) )

'Racket

> ( noun )

'(dog)

> ( noun )

'(robot)

> ( noun )

'(hat)

> ( noun )

'(toddler)

> ( verb )

'(hugged)

> ( verb )

'(chased)

> ( verb )

'(chased)

> ( verb )

'(hornswoggled)
```

```
> ( article )
'(the)
> ( article )
'(the)
> ( article )
'(a)
> ( article )
'(a)
> ( qualifier )
'(howling)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'(happy)
> ( qualifier )
'(happy)
> ( qualifier )
'(laughing)
> ( qualifier )
'(happy)
> ( qualifier )
'(dancing)
> ( qualifier )
'(happy)
> ( qualifier )
'(talking)
```

```
> ( qualifier )
'(howling)
> ( qualifier )
'(howling)
> ( qualifier )
'(howling)
> ( qualifier )
'()
> ( qualifier )
'(happy)
> ( qualifier )
'()
> ( noun-phrase )
'(the barking robot)
> ( noun-phrase )
'(the laughing toddler)
> ( noun-phrase )
'(a dancing hat)
> ( noun-phrase )
'(the dancing dog)
> ( noun-phrase )
'(the toddler)
> ( noun-phrase )
'(a howling hat)
> ( noun-phrase )
'(a happy robot)
> ( noun-phrase )
'(a barking hat)
```

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

```
> ( ds )
the howling robot hugged a baby
> ( ds )
the dancing robot kissed the barking toddler
>
```