Tyler Cullen
CSC 344
2/4/2023

Racket Programming Assignment
First Interactions

Abstract

The assignment Racket Programming First Interactions will introduce us to the basics of Racket programming which includes installing Racket/DrRacket and creating a LEL sentence generator using nouns, verbs, qualifiers and articles.

Racket 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
 )
```

Racket Demo for LEL 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 ) )
'blue
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'red
> (pick '( Racket Prolog Haskell Rust))
'Rust
> (pick '( Racket Prolog Haskell Rust))
'Haskell
> (pick '( Racket Prolog Haskell Rust))
'Rust
> (noun)
'(hat)
> (noun)
'(baby)
> (noun)
'(hat)
> (noun)
'(dog)
> (verb)
'(hornswoggled)
> (verb)
'(protected)
> (verb)
'(kissed)
> (verb)
'(protected)
> (article)
'(the)
> (article)
'(the)
> (article)
```

```
'(a)
> (article)
'(a)
> (qualifier)
'()
> (qualifier)
'(barking)
> (qualifier)
'(dancing)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(dancing)
> (qualifier)
'(barking)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(laughing)
> (qualifier)
'(barking)
> (qualifier)
'(talking)
> (qualifier)
'(barking)
> (qualifier)
'()
> (qualifier)
'(talking)
> (qualifier)
'(laughing)
> (noun-phrase)
'(a baby)
> (noun-phrase)
'(a dog)
```

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

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