Temitope Emokpae 2/5/2023

CSC 344

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

Learning Abstract

This programming assignment is about typing the program presented in appendix 1 into the definitions area and mimicking the demo in appendix 2.

Main Task #1: Mindfully Type a Program into the Definitions Area

Demo:

```
#lang racket
;-------------------------------------------------
; LEL sentence generator, with helper PICK,
; serveral 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 todler 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 )
   )
```

Main Task #2: Generate a Demo by Mimicking a Given Demo

Demo:

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> ( pick '( red yellow blue ) )
'red
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'blue
> ( pick '( red yellow blue ) )
'red
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( pick '( Racket Prolog Haskell Rust ) )
'Racket
> ( pick '( Racket Prolog Haskell Rust ) )
'Rust
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( noun )
'(robot)
> ( noun )
'(robot)
> ( noun )
'(hat)
> ( noun )
'(baby)
> ( verb )
'(protected)
> ( verb )
'(hugged)
> ( verb )
'(kissed)
> ( verb )
'(hugged)
> ( article )
'(the)
> ( article )
'(a)
> ( article )
'(the)
```

```
> ( article )
'(the)
> ( qualifier )
'(laughing)
> ( qualifier )
'()
> ( qualifier )
'(dancing)
> ( qualifier )
'(laughing)
> ( qualifier )
'()
> ( qualifier )
'(howling)
> ( qualifier )
'()
> ( qualifier )
'(howling)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'(talking)
> ( qualifier )
'(howling)
> ( qualifier )
'()
> ( qualifier )
'(dancing)
> ( qualifier )
'(barking)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'()
> ( noun-phrase )
'(the howling baby)
```

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

```
> ( ds )
a dancing todler hugged the baby
> ( ds )
the talking baby hornswoggled the laughing dog
> ( ds )
the laughing todler hornswoggled the dog
>
```