Matlab OPP

May 31, 2018

0.1 Object Oriented Malab programming

0.1.1 1. Objects

```
In [13]: a = 2
         class(a)
```

```
a =

     2


ans =

    'double'
```

```
In [14]: methods(a)
```

```
Methods for class double:

abs              cscd             hess             polylog
accumarray       csch             hypot            pow2
acos             ctranspose       ichol            power
acosd            cummax           igamma           psi
acosh            cummin           ilu              qrupdate
acot             cumprod          imag             rcond
acotd            cumsum           inv              rdivide
acoth            dawson           isbanded         real
acsc             delete           isdiag           reallog
acscd            det              isfinite         realpow
acsch            diag             isinf            realsqrt
airy             diff             isnan            rectangularPulse
all              dilog            issorted         rem
amd              dirac            issortedrows     reshape
and              display          istril           round
```

```
any             divisors        istriu          sec
asec            dmperm          jacobiP         secd
asecd           ei              jordan          sech
asech           ellipticCE      kummerU         sign
asin            ellipticCK      laguerreL       signIm
asind           ellipticCPi     ldivide         sin
asinh           ellipticE       le              sind
atan            ellipticF       legendreP       sinh
atan2           ellipticK       linsolve        sinhint
atan2d          ellipticNome    log             sinint
atand           ellipticPi      log10           sort
atanh           eps             log1p           sortrowsc
bernoulli       eq              log2            sparse
besselh         erf             logint          sqrt
besseli         erfc            lt              ssinint
besselj         erfcinv         ltitr           superiorfloat
besselk         erfcx           maxk            symrcm
bessely         erfi            mink            tan
betainc         erfinv          minpoly         tand
betaincinv      euler           minus           tanh
bsxfun          exp             mldivide        times
ceil            expm1           mod             transpose
charpoly        find            mpower          triangularPulse
chebyshevT      fix             mrdivide        tril
chebyshevU      floor           mtimes          triu
colon           fresnelc        ne              uminus
conj            fresnels        nnz             uplus
cos             gamma           nonzeros        whittakerM
cosd            gammainc        not             whittakerW
cosh            gammaincinv     nzmax           wrightOmega
coshint         gammaln         or              xor
cosint          ge              ordeig          zeta
cot             gegenbauerC     permute
cotd            gt              plus
coth            harmonic        pochhammer
csc             hermiteH        poly2sym
```

```
In [16]: log10(a)

ans =

    0.3010
```

Examples from Alitrack

```
In [5]: p_folder = '/Users/ryanlab/Desktop/AliT/Data/ALITracker_Data/aj031ro/aj031ro.edf';
        myparticipant = participant(p_folder, 'samples', true);

Importing file /Users/ryanlab/Desktop/AliT/Data/ALITracker_Data/aj031ro/aj031ro.edf...
Loading 70 trials [|         ||        |||       ||||      |||||     ||||||    |||||||   |||||
Truncating samples to exclude empty ones
done!


In [6]: %Object orieneted easy to read code, easy to us
        myparticipant.set_trial_features(1:70,'start_event',"Study_display", 'end_event', "Stu

In [7]: %Making a new object.
        new_trial = myparticipant.gettrial(1)


new_trial =

  trial with properties:

                parent: [1x1 participant]
                  data: []
              trial_no: 1
        trial_fieldname: 'trial_1'
            num_samples: 3992
                  index: [1x3992 uint32]
            sample_time: [1x3992 uint32]
             trial_time: [1x3992 uint32]
                      x: [1x3992 double]
                      y: [1x3992 double]
                    rho: []
                  theta: []
    issaccadeorfixation: []
              fixations: [1x1 struct]
             isfixation: []
               saccades: [1x1 struct]
              issaccade: []
                   rois: [1x1 struct]
              condition: []



In [8]: new_trial.saccades


ans =
```

```
  struct with no fields.
```

```
In [9]: new_trial.number_of_saccade()
        new_trial.saccades
```

```
ans =

  struct with fields:

    rawindex: [3x1 double]
      number: 3
       start: [736000 1869000 2597000]
         end: [744000 1876000 2615000]
```

```
In [10]: new_trial.number_of_fixation()
         new_trial.number_of_saccade()
         new_trial.duration_of_fixation()
         new_trial.duration_of_saccade()
         new_trial.location_of_fixation()
         new_trial.location_of_saccade()
         new_trial.amplitude_of_saccade()
         new_trial.deviation_of_duration_of_fixation
         new_trial.deviation_of_duration_of_saccade
         new_trial.get_polar
         new_trial.get_issaccade
         new_trial.get_isfixation

         new_trial
```

```
new_trial =

  trial with properties:

                parent: [1x1 participant]
                  data: []
              trial_no: 1
       trial_fieldname: 'trial_1'
           num_samples: 3992
                 index: [1x3992 uint32]
           sample_time: [1x3992 uint32]
            trial_time: [1x3992 uint32]
                     x: [1x3992 double]
                     y: [1x3992 double]
```

```
                  rho: [1x3992 double]
                theta: [1x3992 double]
    issaccadeorfixation: []
              fixations: [1x1 struct]
             isfixation: [1x3992 double]
               saccades: [1x1 struct]
              issaccade: [1x3992 double]
                   rois: [1x1 struct]
              condition: []
```

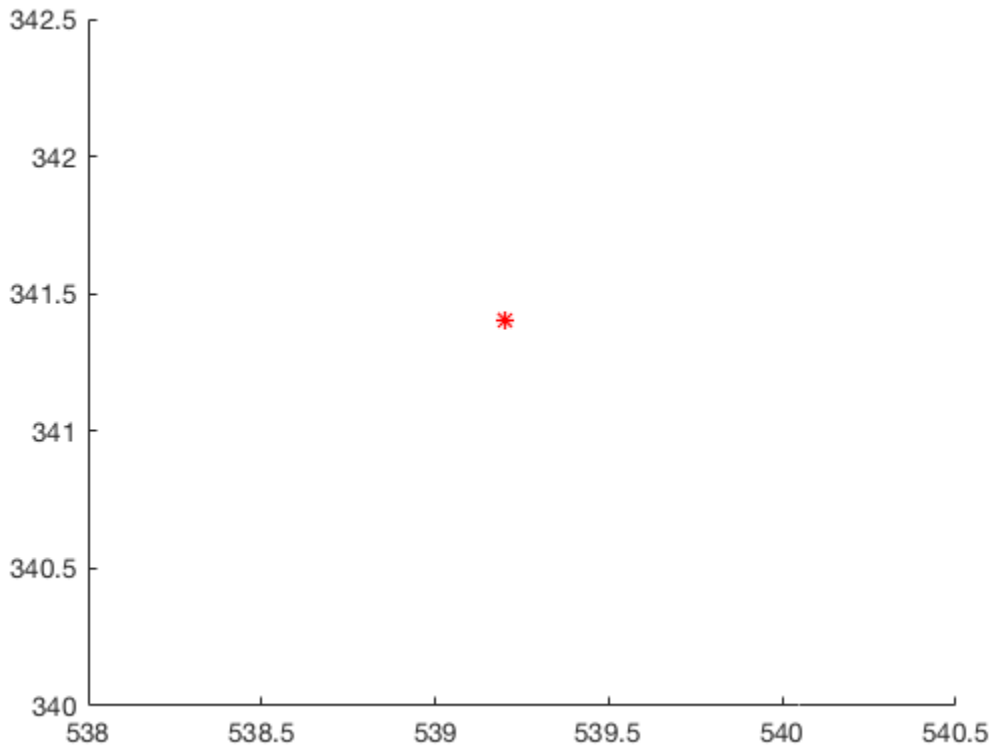
```
In [11]: new_trial.saccades
```

```
ans =

  struct with fields:

              rawindex: [3x1 double]
                number: 3
                 start: [736000 1869000 2597000]
                   end: [744000 1876000 2615000]
              duration: [1123000 719000 1375000]
           start_gazex: [3x1 double]
           start_gazey: [3x1 double]
             end_gazex: [3x1 double]
             end_gazey: [3x1 double]
             amplitude: [3x1 double]
    duration_variation: [0.1531 -1.0677 0.9146]
```

```
In [12]: new_trial.animate
```



0.1.2 2. Creating Classes

excerpt from matlab's website MATLAB classes use the following words to describe different parts of a class definition and related concepts.

Class definition — Description of what is common to every instance of a class.

Properties — Data storage for class instances

Methods — Special functions that implement operations that are usually performed only on instances of the class

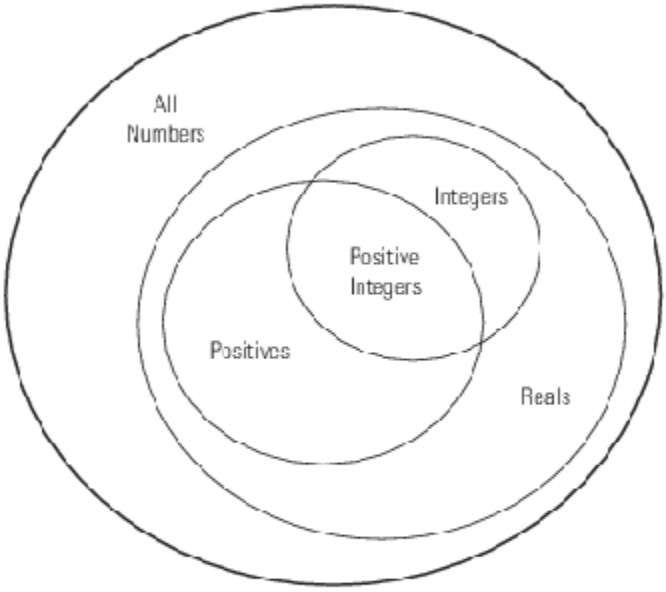
Events — Messages defined by classes and broadcast by class instances when some specific action occurs

Attributes — Values that modify the behavior of properties, methods, events, and classes

Listeners — Objects that respond to a specific event by executing a callback function when the event notice is broadcast

Objects — Instances of classes, which contain actual data values stored in the objects' properties

```
In [19]: imshow('1.object_oriented/assets/classes_matlab.png')
```



Subclasses — Classes that are derived from other classes and that inherit the methods, properties, and events from those classes (subclasses facilitate the reuse of code defined in the superclass from which they are derived).

Superclasses — Classes that are used as a basis for the creation of more specifically defined classes (that is, subclasses).

Packages — Folders that define a scope for class and function naming

```
In [ ]: myparticipant
```

```
myparticipant =

  participant with properties:
```

```
            id: []
       address: []
        trials: {1x70 cell}
          subs: {[31]}
          data: {[1x70 struct]}
           raw: []
    subject_var: 'ID'
        screen: [1x1 struct]
          rois: [1x1 struct]
           log: [1x1 struct]
```

0.1.3 3. Handles

In [24]:
```
% Example directory from matlab's website

%function myFunc(var)
%   var = var + 1;
%end

myFunc = @(x) x+1;

x = 12;
myFunc(x);
x
```

```
x =

    12
```

0.1.4 3. Resources

External relevant links

- Referenced mit lecture slides
- Reference mit lecture video
- Matlab documentation

My personal links

- Javascript javascript code
- Referenced matlab code
- this talk's directory on github

written by Alireza Tajdod