L17. Structures

Simple Structures
Structure Arrays
Structures with Array Fields
Other Possibilities

Data is Often Related
A point in the plane has an x coordinate and y coordinate.

If a program manipulates lots of points, there will be lots of x's and y's.

Anticipate clutter. Is there a way to "package" the two coordinate values?

Packaging Affects Thinking

Our Reasoning Level:
P and Q are points. Compute the midpoint M of the connecting line segment.

Behind the scenes we do this:
\[ M_x = \frac{P_x + Q_x}{2} \quad M_y = \frac{P_y + Q_y}{2} \]

Seen This Before

Functions are used to "package" calculations.

Elevates the level of our reasoning.

Critical for problem solving.

Packaging

Functions "package" calculations.
Structures "package" data.

Simple Example

\[
P1 = \text{struct('x',3,'y',4);}\\
P2 = \text{struct('x','-1','y',7);}\\
D = \text{sqrt((P1.x-P2.x)^2 + (P1.y-P2.y)^2);}\\
\]

Distance between two points.

P1.x, P1.y, P2.x, P2.y participating as variables—because they are.
Initialization

\[ P1 = \text{struct}('x',3,'y',4); \]

\( P1 \) is a structure.
The structure has two fields.
Their names are \textit{x} and \textit{y}.
They are assigned the values 3 and 4.

How to Visualize \( p1 \)

Accessing a Field

\[ A = p1.x + p1.y \]
Assigns the value 7 to \( A \).

Assigning to a Field

\[ p1.x = p.y^2 \]
Will assign the value 16 to \( p1.x \).

Another Example

\[ A = \text{struct}('name','New York',…
    'capital', 'Albany',…
    'Pop',15.5) \]
Can have combinations of string fields and numeric fields.

Legal/Illegal Maneuvers

\[ P = \text{struct}('x',3,'y',4); \]
\[ Q = \text{struct}('x',5,'y',6); \]
\[ R = Q \quad \% \text{Legal. } R \text{ is copy of } Q \]
\[ S = (P+Q)/2 \quad \% \text{Illegal.} \]
Legal/Illegal Maneuvers

% Illegal...
P = struct('x',3,'y')
P.y = 4

% Legal
P = struct('x',3,'y',[])
P.y = 4

Using the Empty array as a place holder

A Function Can Have Inputs that are Structures

function d = dist(P,Q)
% P and Q are points.
% d is the distance between them
D = sqrt((P.x-Q.x)^2 + ...
    (P.y-Q.y)^2);

A Function Can Return a Structure

function P = MakePoint(x,y)
% P is a point with P.x and P.y
% assigned the values x and y.
P = struct('x',x,'y',y);

Good Style. Highlights the structure's definition.

Functions and Structures

function DrawLS(P,Q,c)
% P and Q are points.
% Draws a line segment connecting
% P and Q. Color specified by c
plot([P.x Q.x],[P.y Q.y],c)

Pick Up Sticks Script

s = 'rgbmcy';
for k=1:100
    P = MakePoint(randn(1),randn(1));
    Q = MakePoint(randn(1),randn(1));
    c = s(ceil(6*rand(1)));
    DrawLS(P,Q,c)
end

Generates two random points and chooses one of six colors randomly.
Structure Arrays

An array whose components are structures.
And all the structures are the same.

Example: An array of points...

Use this “Make” Function

function P = MakePoint(x,y)
% P is a point with P.x and P.y
% assigned the values x and y.
P = struct('x',x,'y',y);

An Array of Points

P(1) = MakePoint(.50,.86)

An Array of Points

P(2) = MakePoint(-.50,.86)

An Array of Points

P(3) = MakePoint(-1.0,0.0)

An Array of Points

P(4) = MakePoint(-.50,-.86)
An Array of Points

\[ P(5) = \text{MakePoint}(0.50, -0.86) \]

\[ P(6) = \text{MakePoint}(1.00, 0.00) \]

A Function that Returns an Array of Points

\[
\begin{align*}
\text{function } & P = \text{CirclePoints}(n) \\
\theta & = 2 \pi / n; \\
\text{for } & k = 1:n \\
& c = \cos(\theta k); \\
& s = \sin(\theta k); \\
& P(k) = \text{MakePoint}(c, s);
\end{align*}
\]

Structures with Array Fields

Let’s develop a structure that can be used to represent a colored disk.

Four fields:
- \( x_c \): x-coordinate of center
- \( y_c \): y-coordinate of center
- \( r \): radius
- \( c \): rgb color vector

Examples

\[
\begin{align*}
D1 & = \text{struct}(\{x_c’,1’,y_c’,2’,r’,3’,\ldots\}’c’,[1\ 0\ 1]) \\
D2 & = \text{struct}(\{x_c’,4’,y_c’,0’,r’,1’,\ldots\}’c’,[.2\ .5\ .3])
\end{align*}
\]

Problem

Assume \( D1 \) and \( D2 \) are colored disks. Let’s compute their “average”.

\[
\begin{align*}
r & = (D1.r + D2.r) / 2; \\
xc & = (D1.xc + D2.xc)/2 \\
yc & = (D1.yc + D2.yc)/2 \\
c & = (D1.c + D2.c) / 2; \\
D & = \text{struct}(\{x_c’,xc’,y_c’yc’,r’,r’,c’,c\})
\end{align*}
\]
A Structure Can Have a Field That is a Structure

A = MakePoint(2,3)
B = MakePoint(4,5)
L = struct('P',A,'Q',B)

(One way to represent a line segment with endpoints P and Q.)

S = L.P.y assigns 3 to S.

More on Structures and Arrays

Let's encode a collection of n random points.

Method 1. An array of points.

for k=1:n
  u = randn(); v = randn();
  P(k) = MakePoint(u,v).
end

Method 2. A structure with array fields

x = randn(n,1); y = randn(n,1);
P = struct('x',u,'y',v)
More on Structures and Arrays

Let's encode a collection of \( n \) random points.

Method 1. An array of points.

Method 2. A structure with array fields

Choice sets a "computational stage."

Illustrate Ramifications

% Method 1 Centroid Computation
\[
\begin{align*}
x_s &= 0; \\
y_s &= 0; \\
\text{for } k=1:n \\
& \quad x_s = x_s + P(k).x; \\
& \quad y_s = y_s + P(k).y; \\
\text{end} \\
x_{Bar} &= x_s/n; \; y_{Bar} = y_s/n;
\end{align*}
\]

Illustrate Ramifications

% Method 2 Centroid Computation
% Vectorized version...
\[
\begin{align*}
x_{Bar} &= \text{sum}(P.x)/n; \\
y_{Bar} &= \text{sum}(P.y)/n;
\end{align*}
\]

Illustrate Ramifications

% Method 2 Centroid Computation
% Vectorized version...
\[
\begin{align*}
x_{Bar} &= \text{sum}(P.x)/n; \\
y_{Bar} &= \text{sum}(P.y)/n;
\end{align*}
\]

A Structure Array with Components Whose Fields Are Arrays

function \( P = \text{MakeOutcome}(s,n) \)
% \( s \) a string that names the player.
% \( n \) the number of dice rolls.
\[
T = \text{ceil}(6*\text{rand}(1,n));
\]
\[
P = \text{struct}('name',s,'throws',T);
\]
% \( P \) is an outcome
Appreciate the Hierarchy

G       a structure array
G(2)    component in the structure array
G(2).throws a field in the component
G(2).throws(3) a component of the field

What did the 2nd player throw on the 3rd dice roll?

Designing Structures

function P = MakeOrbit(Name,P,A,phi,psi,rho,N)
% Name = planet/asteroid name (string)
% P = perihelion
% A = aphelion
% phi rotation in the plane of ecliptic
% psi tilt from the plane of the ecliptic
P = struct('x',x,'y',y,'z','z','t',t,...
'name',Name,'P',P,'A',A,...}