CIS 110: Intro to Computer Programming

Lecture 10

Interaction and Conditionals

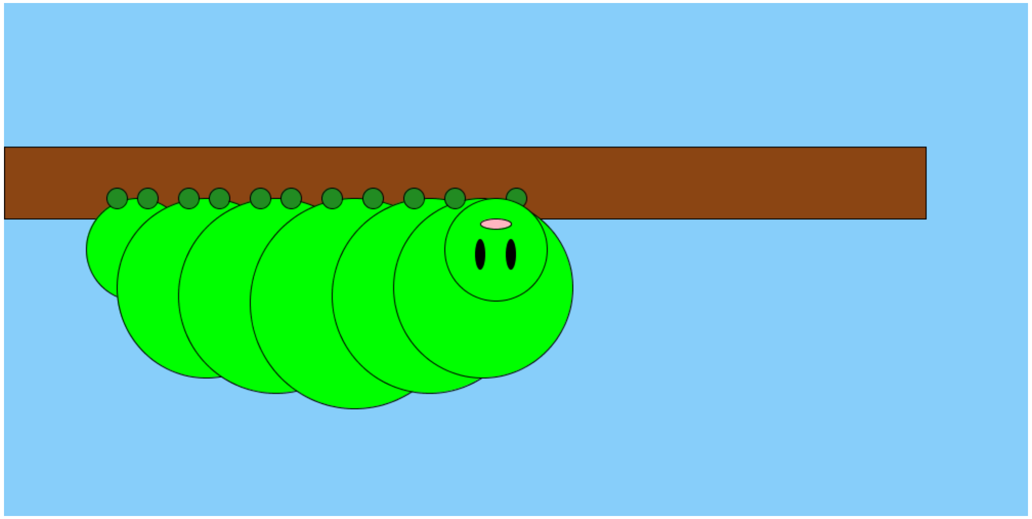
(§ 3.3, 4.1-4.2)

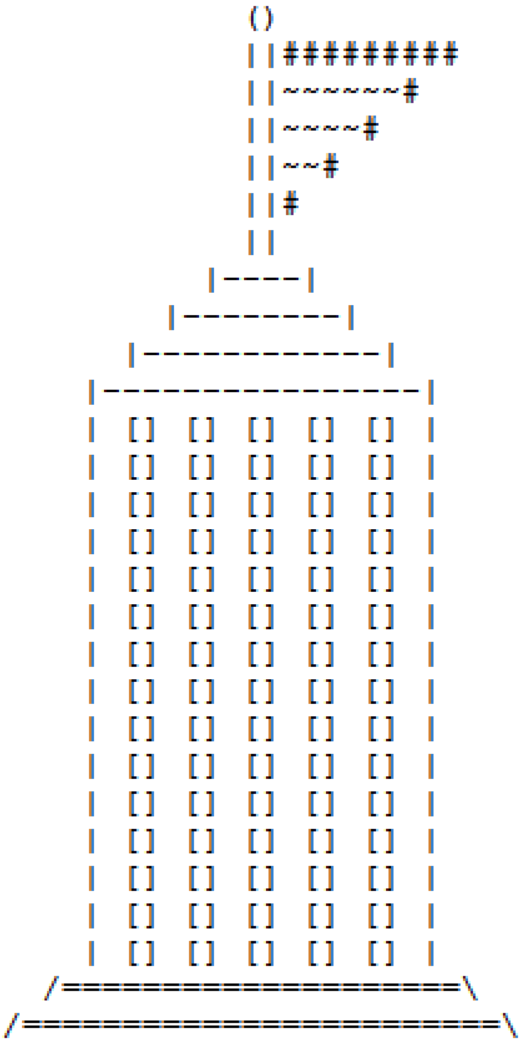
Outline

- The Scanner Object
- Introducing Conditional Statements
- Cumulative Algorithms

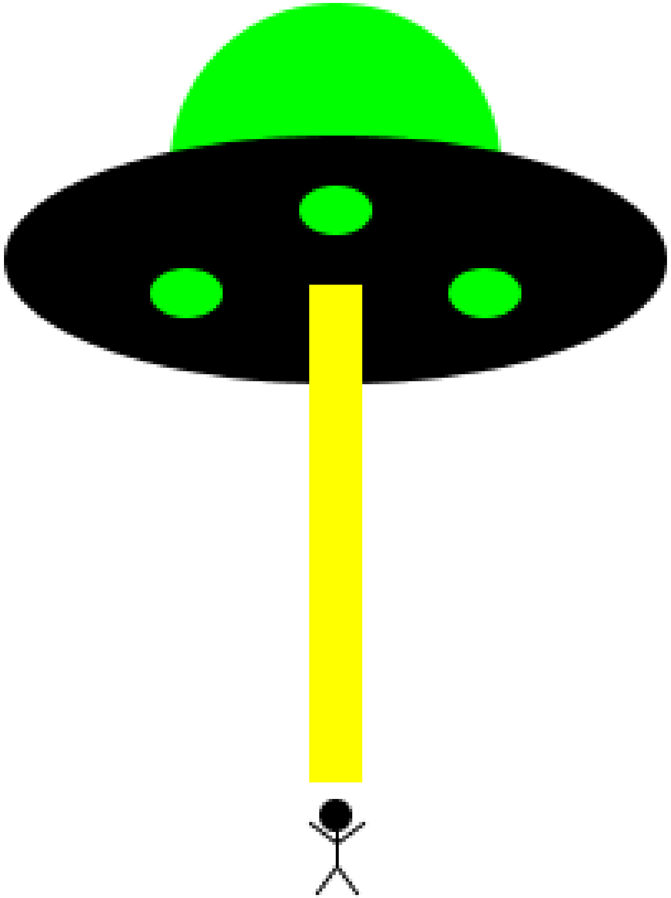
The Scanner Object

What Do Our Programs Look Like?



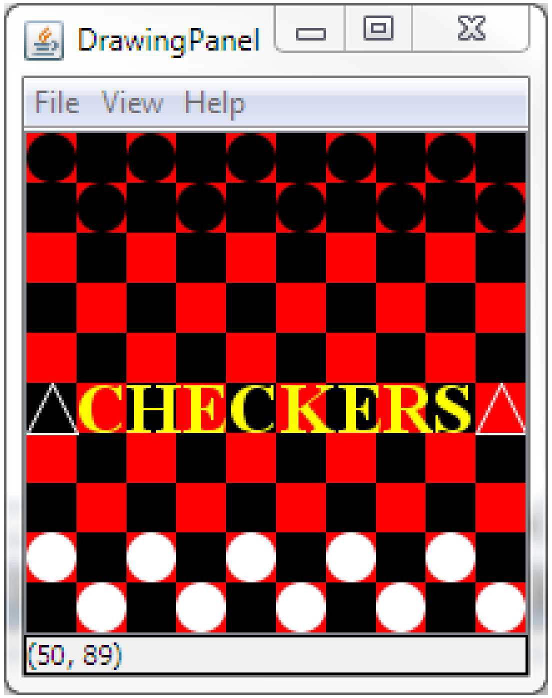


```
            ()
            ||#########
            ||~~~~~~#
            ||~~~~#
            ||~~#
            ||#
            ||
          |----|
        |--------|
      |------------|
    |----------------|
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
    | [] [] [] [] [] |
  /====================\
/========================\
```



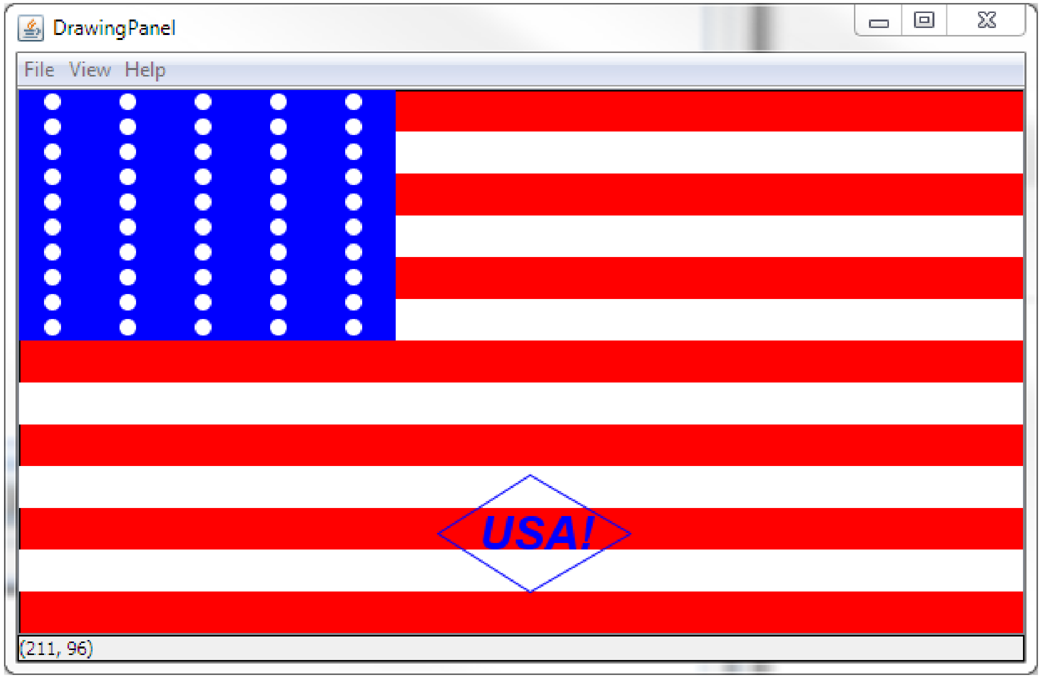
```
This program replicates a drawing
of the sinking Titanic and two
survivors in the water.
****
*****
******
*******
********
*********
**********
***********
***********    \o_   \o/
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
```

```
    - - - - -
   /-/-/-/-/-\
  /-/-/-/-/-\-\
  |-|-|-|-|\-\-\
  |-|-|-|-|-\-\-\
  |-|-|-|-|\-\-\-\
  |-|-|-|-| \-\-\-\
  |-------| |-|-|-|
 / -  -  - \|-|-|-|
/ -   |   - \-|-|-|
|-    o-   -|-|-|-|
\ -       - /-|-|-|
 \ -  -  - /|-|-|-|
  |-------| |-|-|-|
  |-|-|-|-| /-/-/-/
  |-|-|-|-|/-/-/-/
  |-|-|-|-|-/-/-/
  |-|-|-|-|/-/-/
  \-\-\-\-\-/-/
   \-\-\-\-\-/
    - - - - -
```



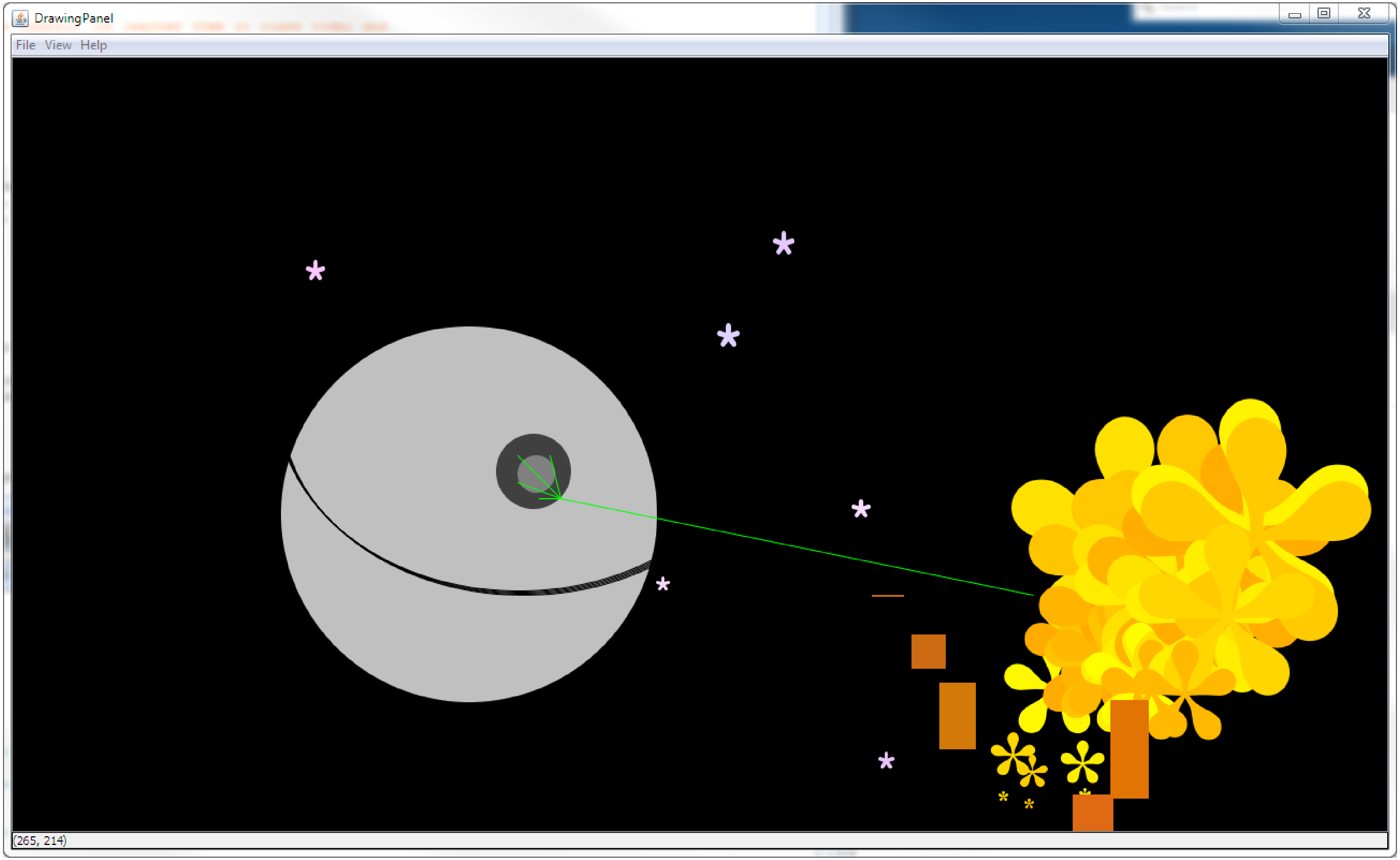
(50, 89)

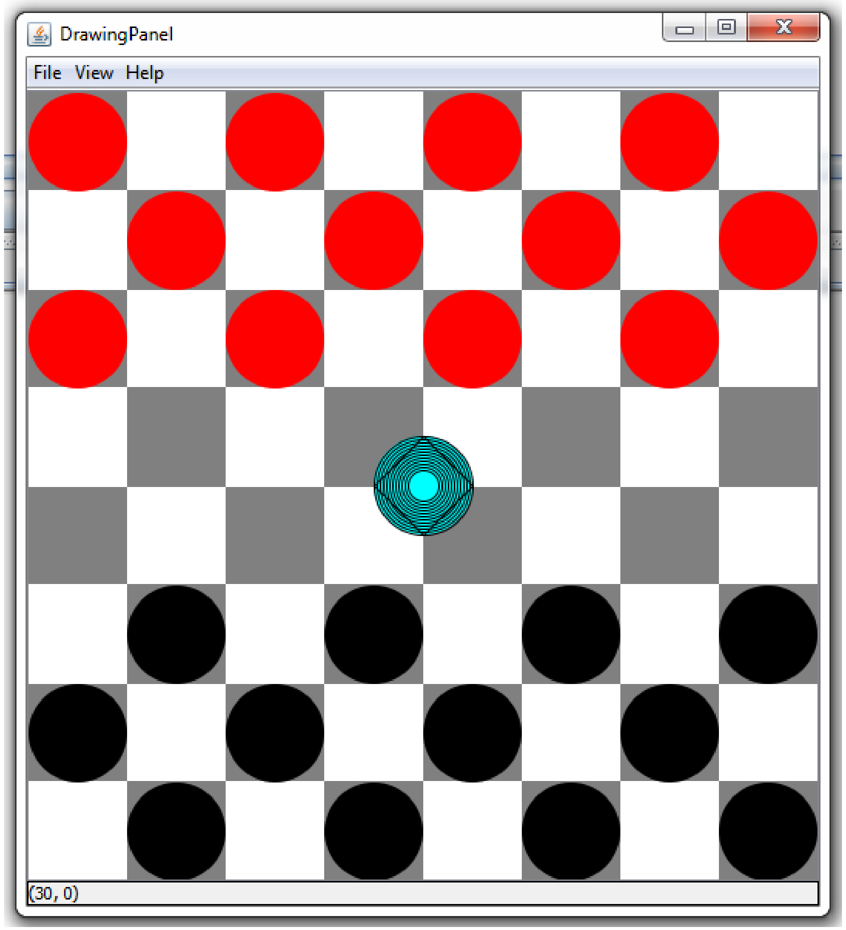
```
 =======================================
|                                       |
|  [oooooo] [oooooo] [oooooo] [oooooo]  |
|  [ [oo] ] [ ]oooo  [ ]oo[ ] [ ]oo[ ]  |
|  [ ]ooooo [  oooo] [ ]  [ ] [ ]  [ ]  |
|  [ ]      [ ]oooo  [ ]  [ ] [ ]  [ ]  |
|  [_]      [_o_o_o] [_]  [_] [_]  [_]  |
|                                       |
| ------------------------------------- |
|                                       |
|          ------------------           |
|          |            //   \          |
|          --          //     \         |
|            |      (-- -)     |        |
|            |      |****|    /         |
|            |      /****|   /          |
|            |     //----)  /           |
|            |    //    ----            |
|            |   //     |               |
|            |  //      |               |
|            | //       |               |
|            |//        |               |
|            |/         |               |
\           /            \              /
 \          |             |            /
  \         (-------------)           /
   \                                 /
    ---------------------------------
```

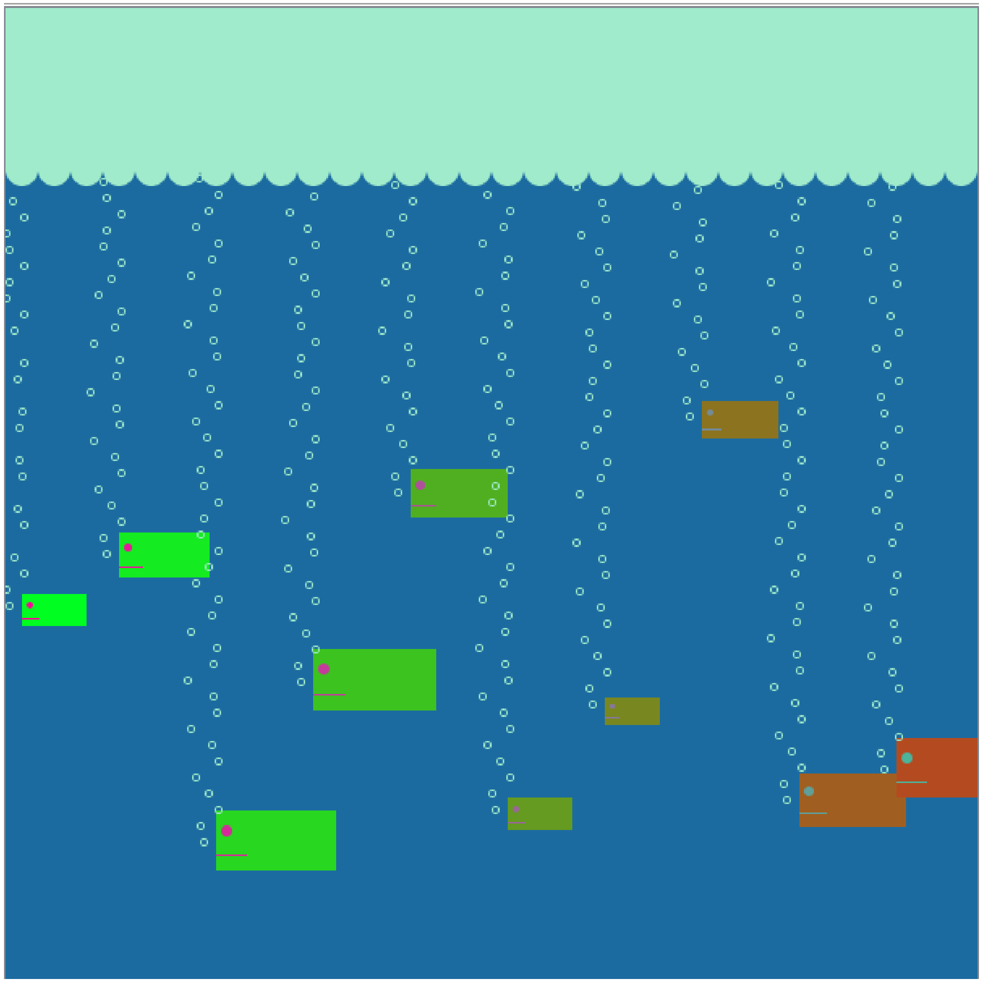


```
      (\_/)
      ('_')
    /""""""""""""\======[===]D
/""""""""""""""""""\
\ _@_ @_ @_ @_ @_  /
```

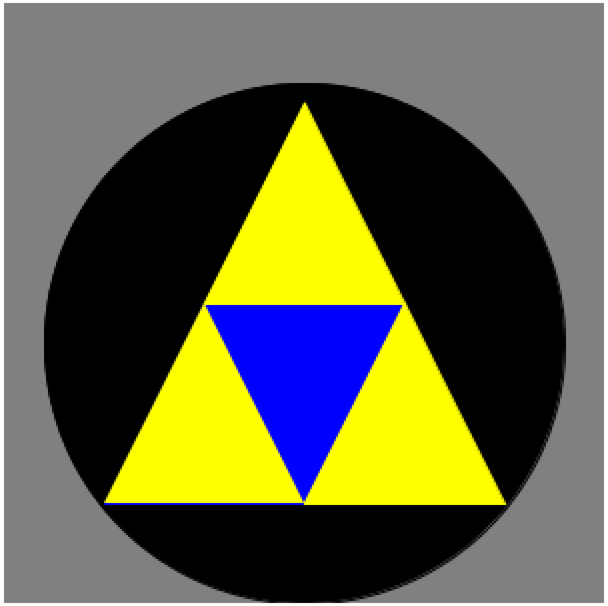
```
================================
%%============================%%
%% { | }    /~~\_   Y   { | } %%
%% {_|_}   %*     \ /   {_|_} %%
%%============================%%
%%             /\             %%
%%           /####\           %%
%%         /##    ##\         %%
%%       /###      ###\       %%
%%     /######    ######\     %%
 %%  /####################\  %%
 %%/###    ####/\####    ###\%%
  %%##      ##/  \##      ##%%
   %%##    #/      \#    ##%%
    %%####/          \####%%
      %%/              \%%
        %%            %%
          %%        %%
            %%%__%%%
```

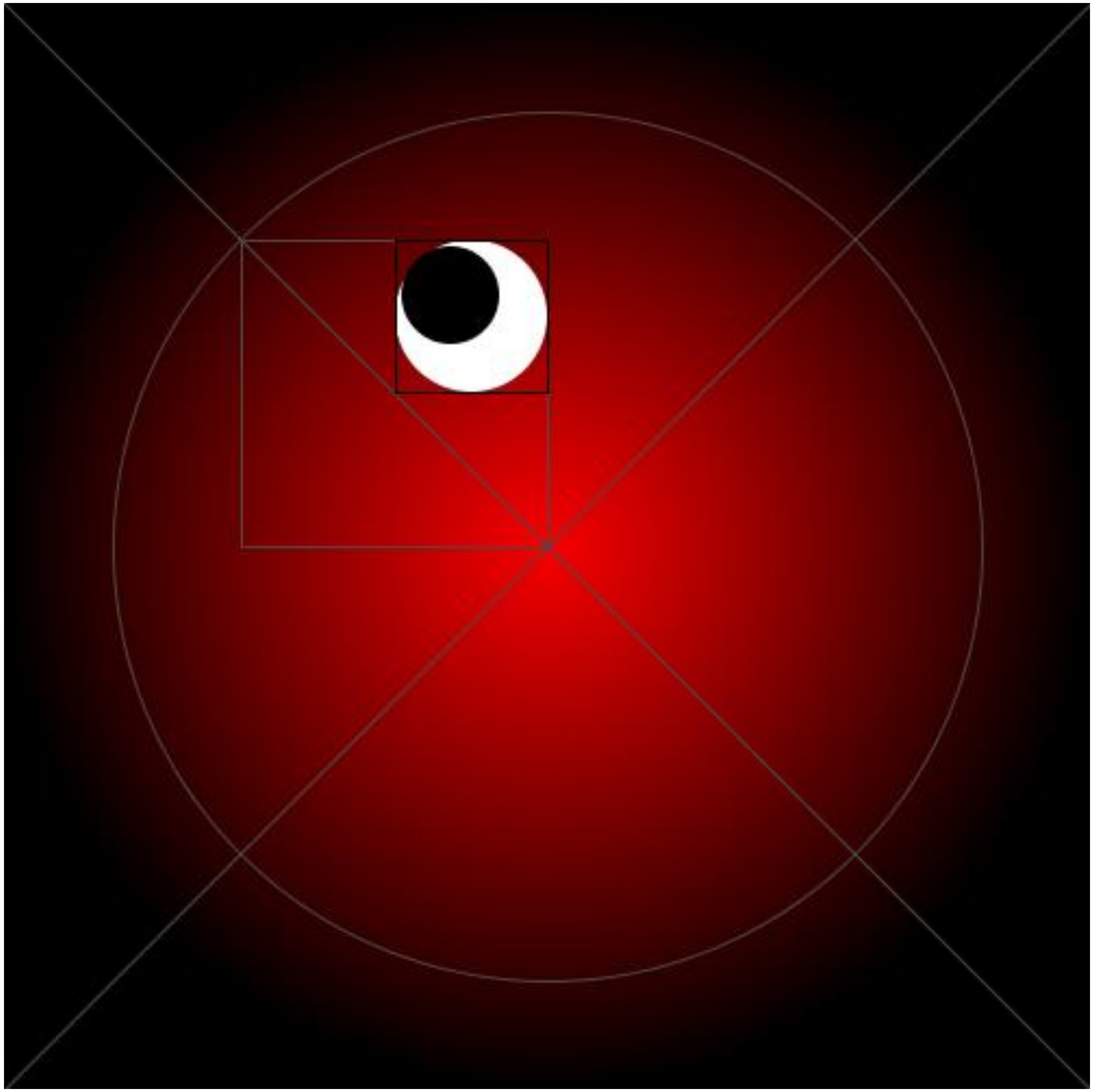

DrawingPanel
File View Help
(265, 214)

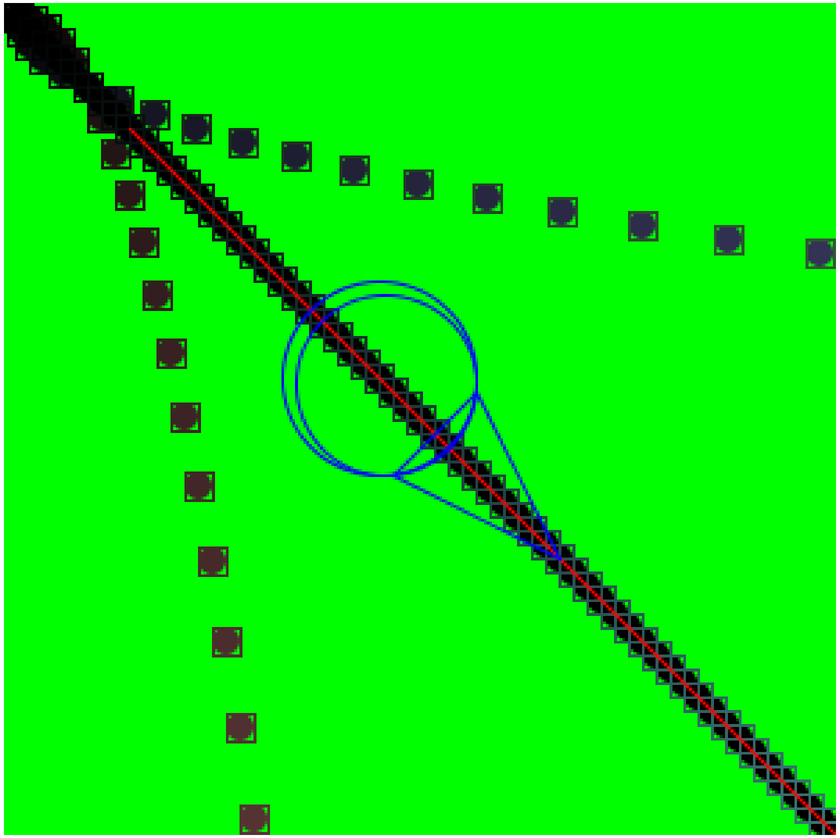

DrawingPanel
File View Help
(30, 0)

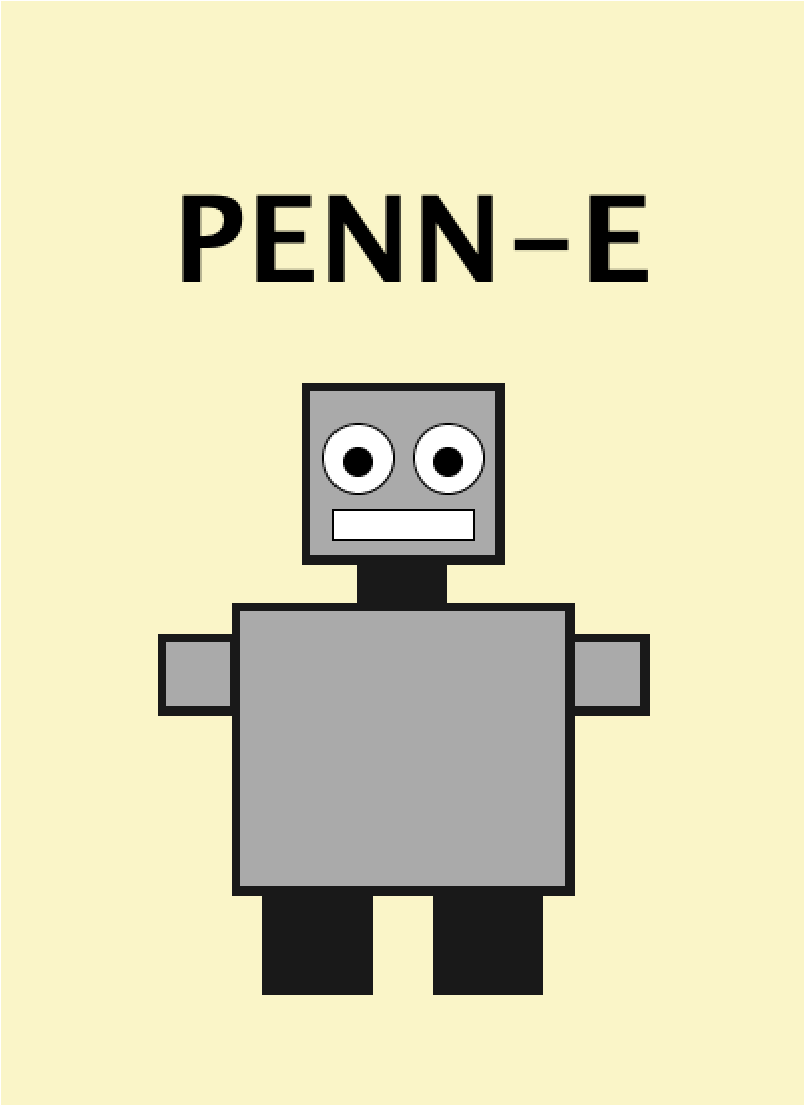


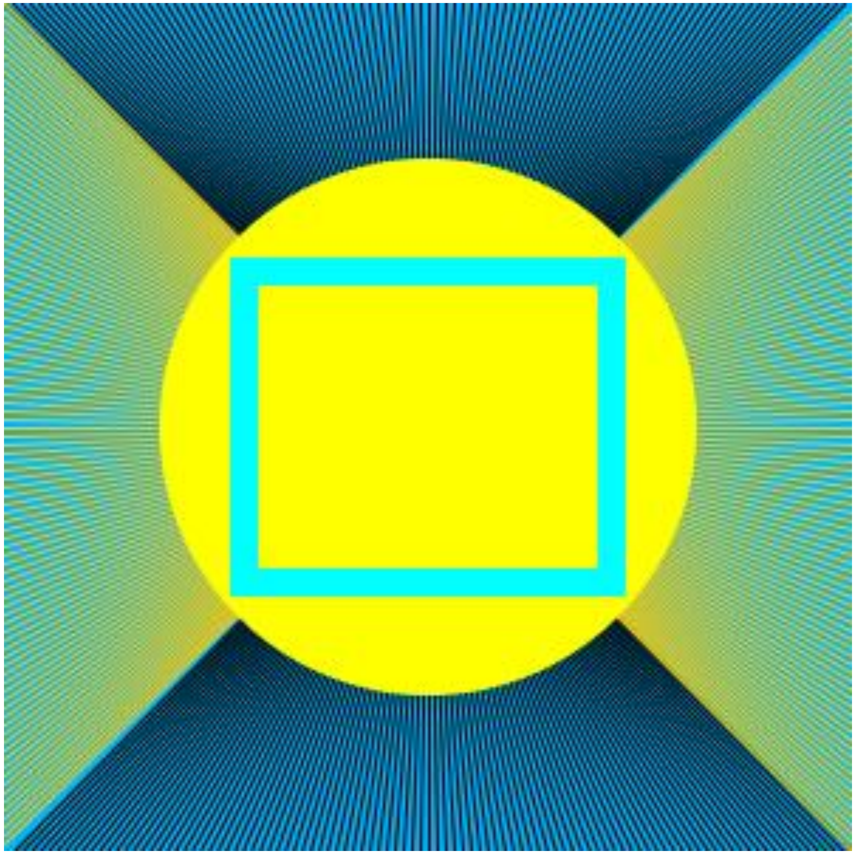
```
       / \     / \
       \./     \./
      -------------
   \/  (o)    (o)  \/
  \/       <>       \/
  ^                  ^
 ^    ************    ^
^    *  /\    /\  *    ^
^   * /\   /\   /\ *   ^
^   * /\   /\   /\ *   ^
 ^  *              *  ^
  ^  *            *  ^
   ^  *          *  ^
      -------------
```

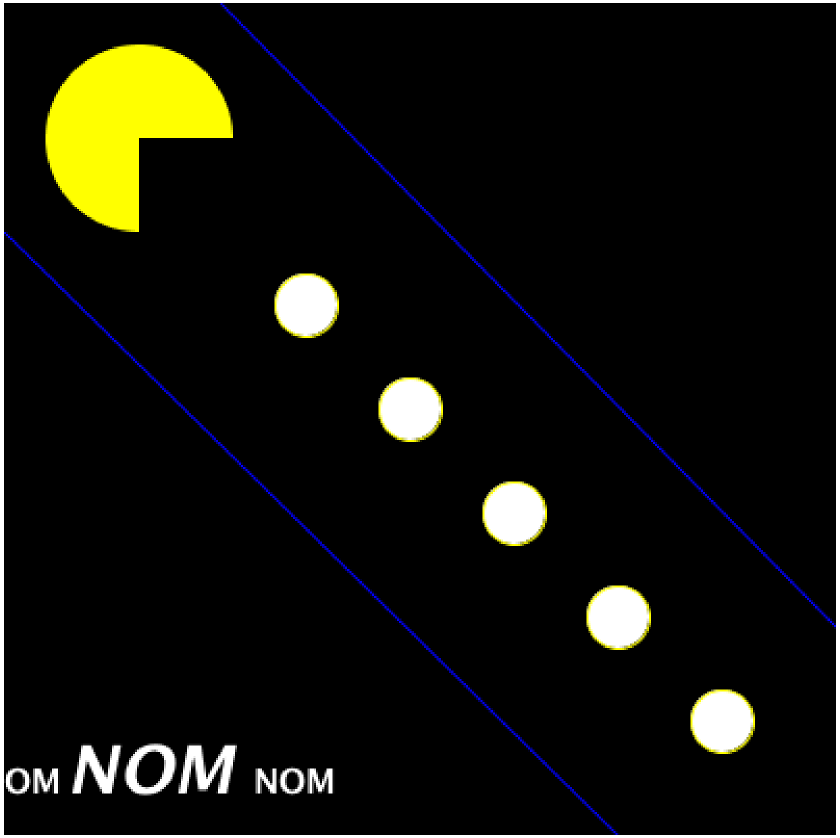




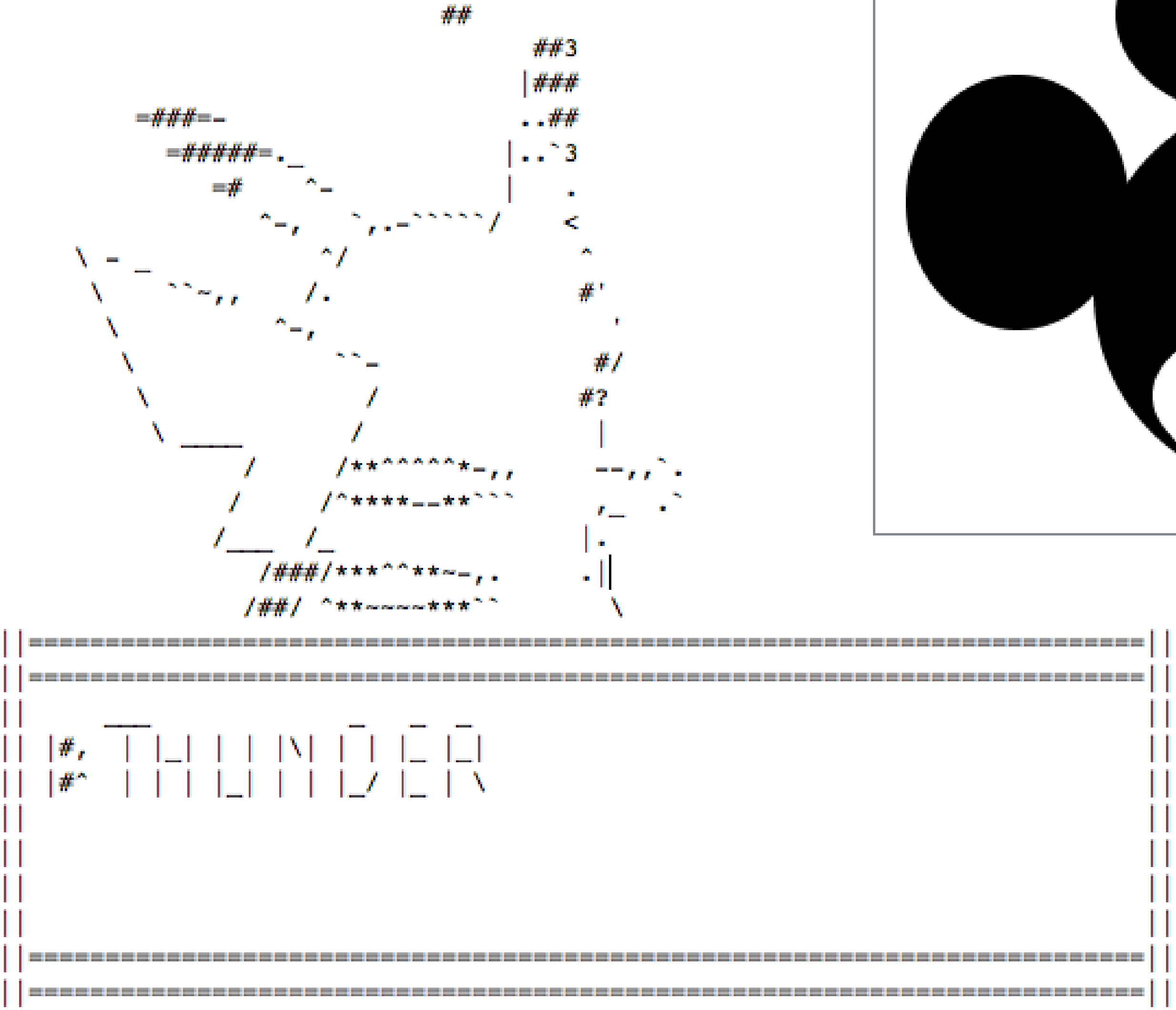



PENN-E

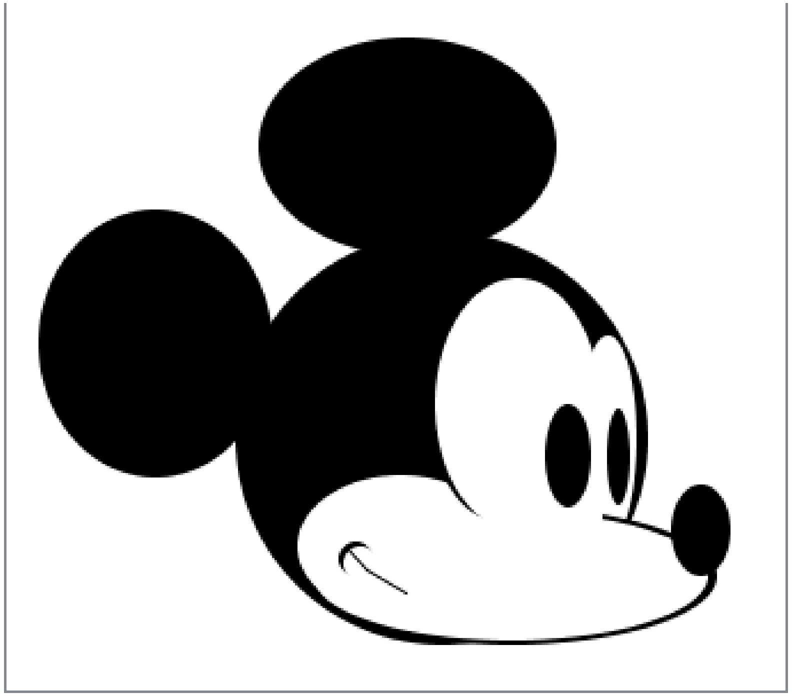


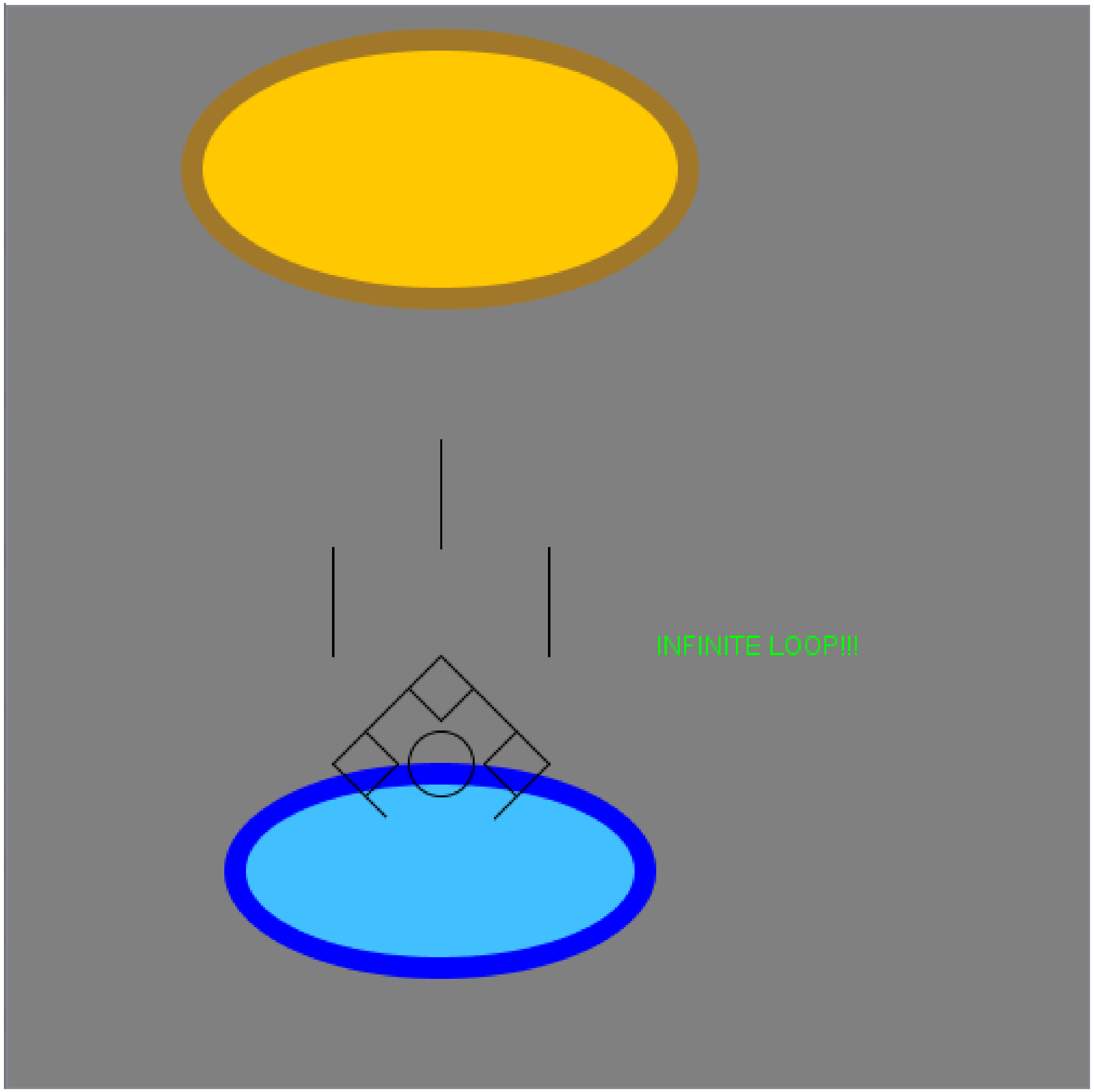


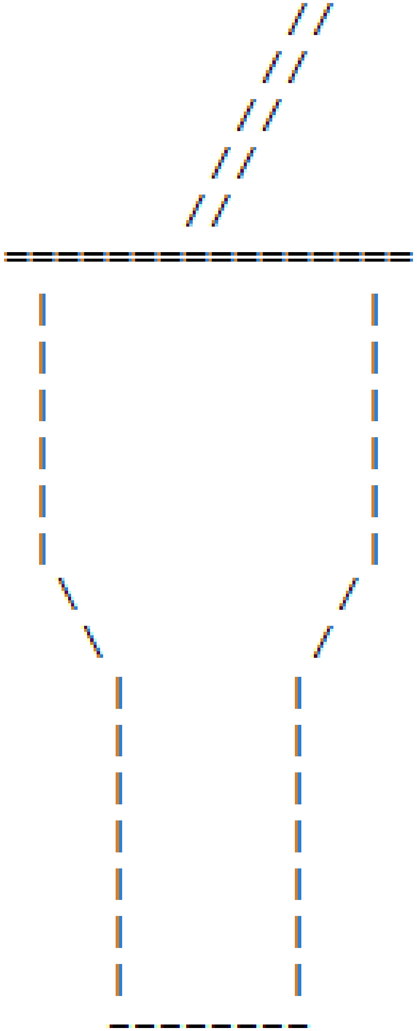
OM NOM NOM



```
                             ##
                                   ##3
                                  |###
         =###=-                   ..##
           =#####=._             |..`3
              =#    ^-           |   .
                 ^-,   `,.-`````/    <
     \ - _           ^/               ^
      \    ``~,,    /.                #'
       \          ^-,                   '
        \             ``-              #/
         \              /             #?
          \ ____       /               |
                /     /**^^^^^*-,,     --,,`.
               /     /^****__**```     ,_  .`
              /____ /_                |.
                 /###/***^^**~-,.     .|
                /##/ ^**~~~~***``       \
||===================================================================================||
||===================================================================================||
||                                                                                   ||
||     ___             _   _  _                                                      ||
|| |#,  | |_| | | |\| | | |_ |_|                                                     ||
|| |#^  | | | |_| | | |_/ |_ | \                                                     ||
||                                                                                   ||
||                                                                                   ||
||                                                                                   ||
||                                                                                   ||
||===================================================================================||
||===================================================================================||
```



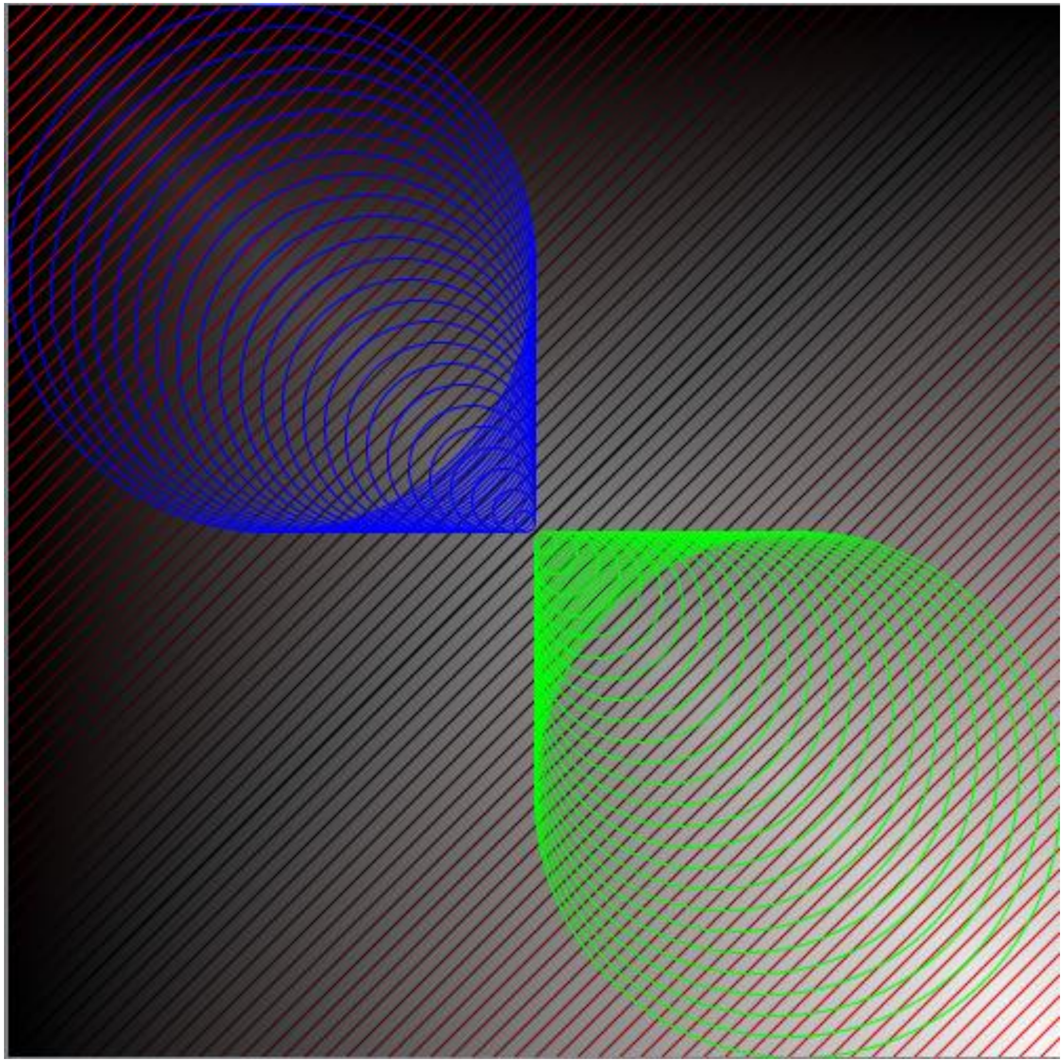

INFINITE LOOP!!!

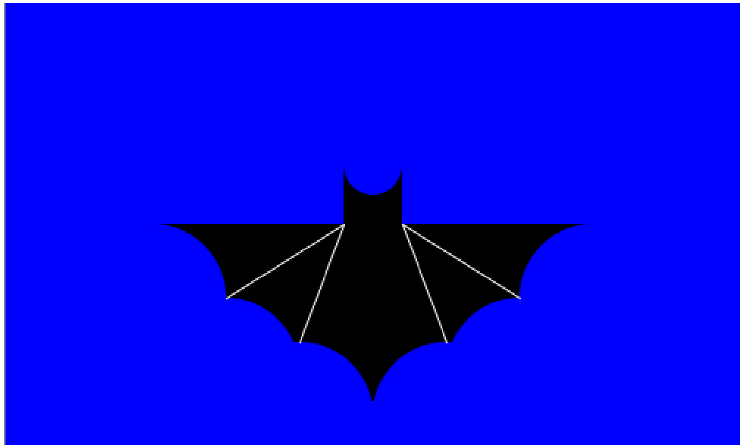


```
            ||
            ||
            ||
            ||
           /||\
        /:::||:::\
     /::::::||::::::\
  /:::::::::||:::::::::\
|""""""""""""""""""""""""|
\_/\/\/\/\/\/\/\/\/\/\/\_/
  \_/\/\/\/\/\/\/\/\/\_/
    \_/\/\/\/\/\/\/\_/
      \_/\/\/\/\/\_/
            ||
            ||
            ||
            ||
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
         |%%||%%|
           /||\
        /:::||:::\
     /::::::||::::::\
  /:::::::::||:::::::::\
```

```
......................................................................
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
P.................................PP.................................P
P                              .PPPPPP.                              P
P                            .PPPPPPPPPP.                            P
P                          .PPPPPPPPPPPPPP.                          P
P                        .PPPPPPPPPPPPPPPPPP.                        P
P                      .PPPPPPP.      .PPPPPPP.                      P
P                    .PPPPPPP.          .PPPPPPP.                    P
P                  .PPPPPPPPP.          .PPPPPPPPP.                  P
P                .PPPPPPPPPPPP.        .PPPPPPPPPPPP.                P
P              .PPPPPPPPPPPPPPPP......PPPPPPPPPPPPPPPP.              P
.P           .PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.           P.
 P         .PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.         P
 .P      .PPPPP.    .PPPPPPPPPPPPPPPPPPPPPPPPPPPP.    .PPPPP.      P.
  P.   .PPPPP.        .PPPPPPPPPP.  .PPPPPPPPPP.        .PPPPP.   .P
  .P..PPPPPPP.        .PPPPPPPP.      .PPPPPPPP.        .PPPPPPP..P.
   .PPPPPPPPP.        .PPPPPP.          .PPPPPP.        .PPPPPPPPP.
    .PPPPPPPPPP......PPPPPP.              .PPPPPP......PPPPPPPPPP.
     .PPPPPPPPPPPPPPPPPPP.                  .PPPPPPPPPPPPPPPPPPP.
      .PPPPPPPPPPPPPPPP.                      .PPPPPPPPPPPPPPPP.
       .PPPPPPPPPPPPP.                          .PPPPPPPPPPPPP.
        .PPPPPPPPP.                                .PPPPPPPPP.
         .PPPPPP.                                    .PPPPPP.
            .PP.                                      .PP.
              .PP.                                  .PP.
                .PP.                              .PP.
                   .PP.                        .PP.
                     .PP..                  ..PP.
                        .PPP..          ..PPP.
                            .PPPPPPPPPPPPPP.
```

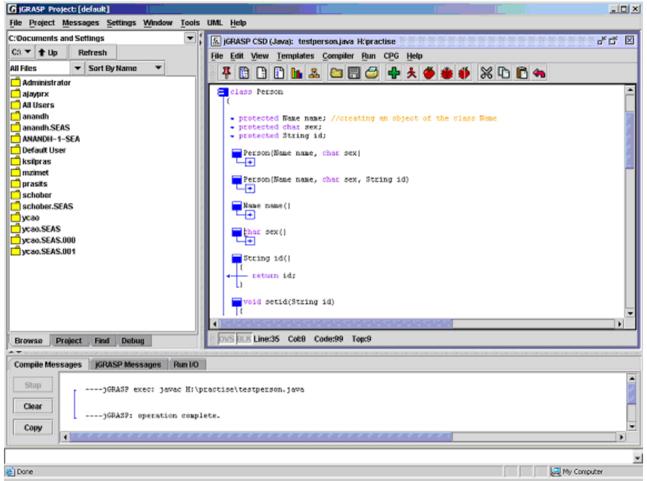
GOTTA CATCH 'EM ALL





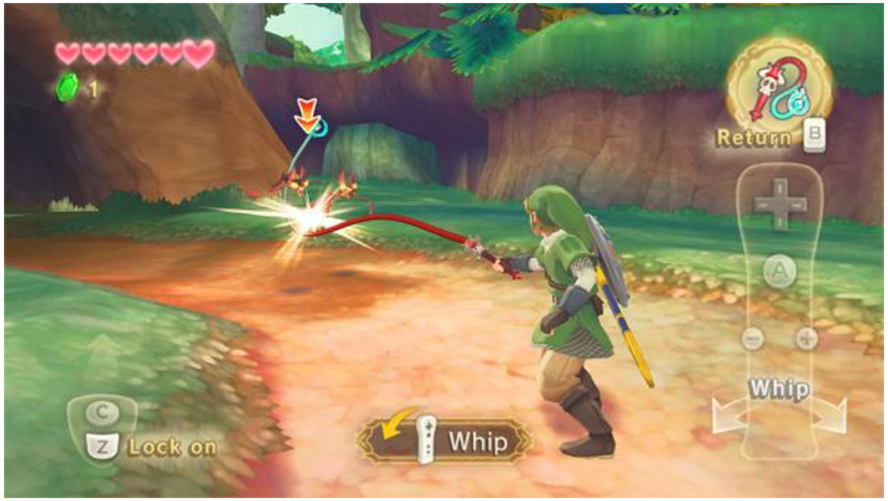
its rollie pollie ollie!

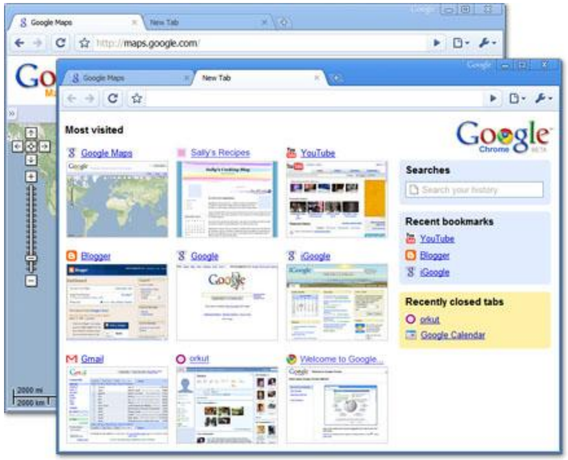
Real World Programs Out There


jGRASP Project: [default]
File Project Messages Settings Window Tools UML Help
C:\Documents and Settings
Up Refresh
All Files Sort By Name
Administrator
ajayprx
All Users
anandh
anandh.SEAS
ANANDH-1-SEA
Default User
ksilpras
mzimet
prasits
schober
schober.SEAS
ycao
ycao.SEAS
ycao.SEAS.000
ycao.SEAS.001
Browse Project Find Debug
jGRASP CSD (Java): testperson.java H:\practise
File Edit View Templates Compiler Run CPG Help
OVS BLK Line:35 Col:0 Code:99 Top:9
Compile Messages jGRASP Messages Run I/O
Stop
Clear
Copy
Done
My Computer

```
class Person
{

 protected Name name; //creating an object of the class Name
 protected char sex;
 protected String id;

  Person(Name name, char sex)

  Person(Name name, char sex, String id)

  Name name()

  char sex()

  String id()
  {
   return id;
  }

  void setid(String id)
  {
```

```
----jGRASP exec: javac H:\practise\testperson.java

----jGRASP: operation complete.
```





Real programs are **INTERACTIVE!**

Introducing the Scanner Class

- Object that lets you read input from the user.

Scanner is in the "java.util" package.

```
import java.util.*;

Scanner in = new Scanner(System.in);
System.out.println(
   "Echo: " + in.nextLine());
```

Creates a new Scanner that reads from some source.

"The keyboard"

Grabs the next line of input from the Scanner.

Reading Input From The User

```
// Reads in a double
double d = in.nextDouble();
// Reads in an integer
int n = in.nextInt();
// Reads in an entire line
String line = in.nextLine();
// Reads a token
String token = in.next();
```

Tokens

- *Tokens* are "chunks" of an input separated by a *delimiter* (here, whitespace).

```
This is a string    with
 "some tokens" in         it!
```

- Tokens: this, is, a, string, with, "some, tokens", in, it!
  - Includes punctuation (e.g., quotes and bangs).

Aside: Packages and import

- Classes are bundled into sets called *packages*.
- The *import declaration* says that you wish to use classes found in a particular package.

```
// Make available all classes in java.util
import java.util.*;
// Make available just the Scanner class
import java.util.Scanner;
```

Conditional Statements

Problem: Making Decisions Based on User Input

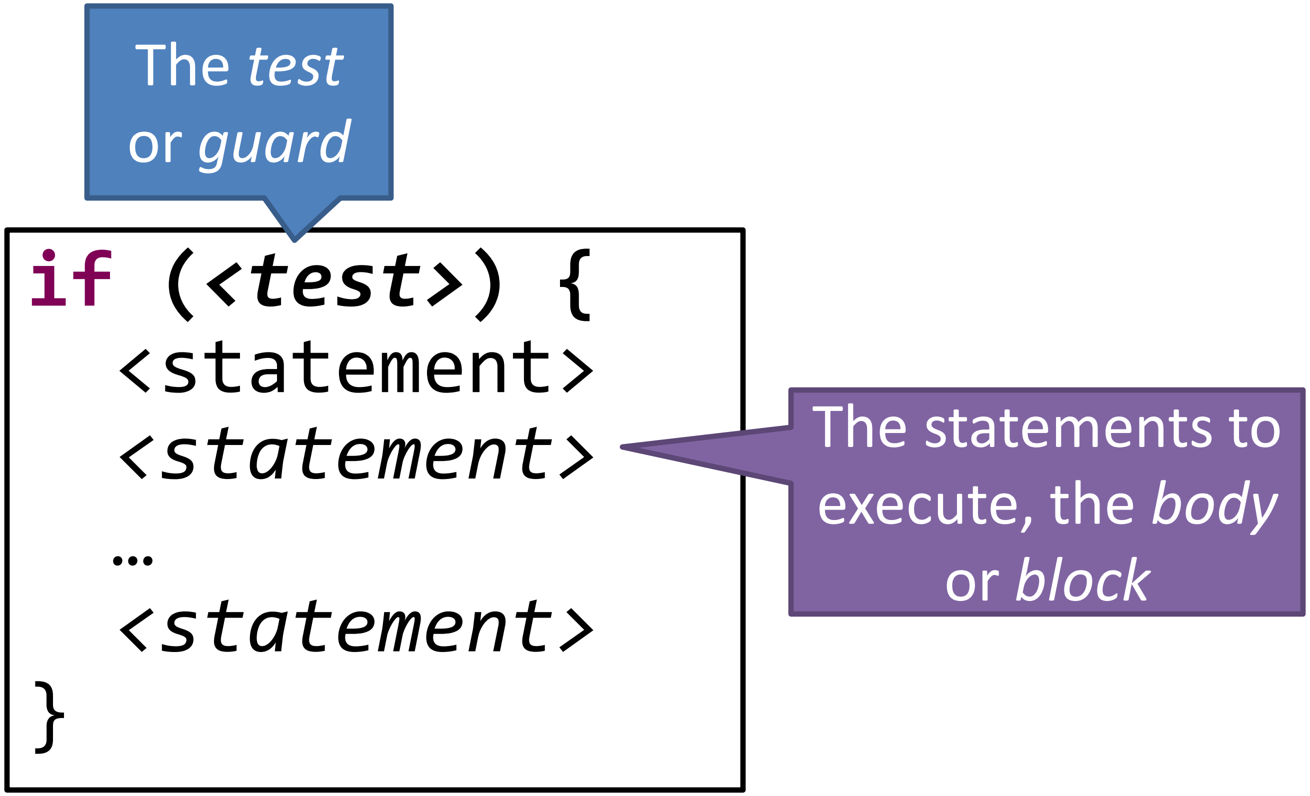
```
Scanner in = new
Scanner(System.in);
double savings = in.nextDouble();
// If amount is greater than 100,
// print a congratulations msg!
```

Introducing Conditional Statements

"Execute this block only if savings is greater than 100."

```
Scanner in = new Scanner(System.in);
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratulations!");
}
```

Syntax of Conditional Statements

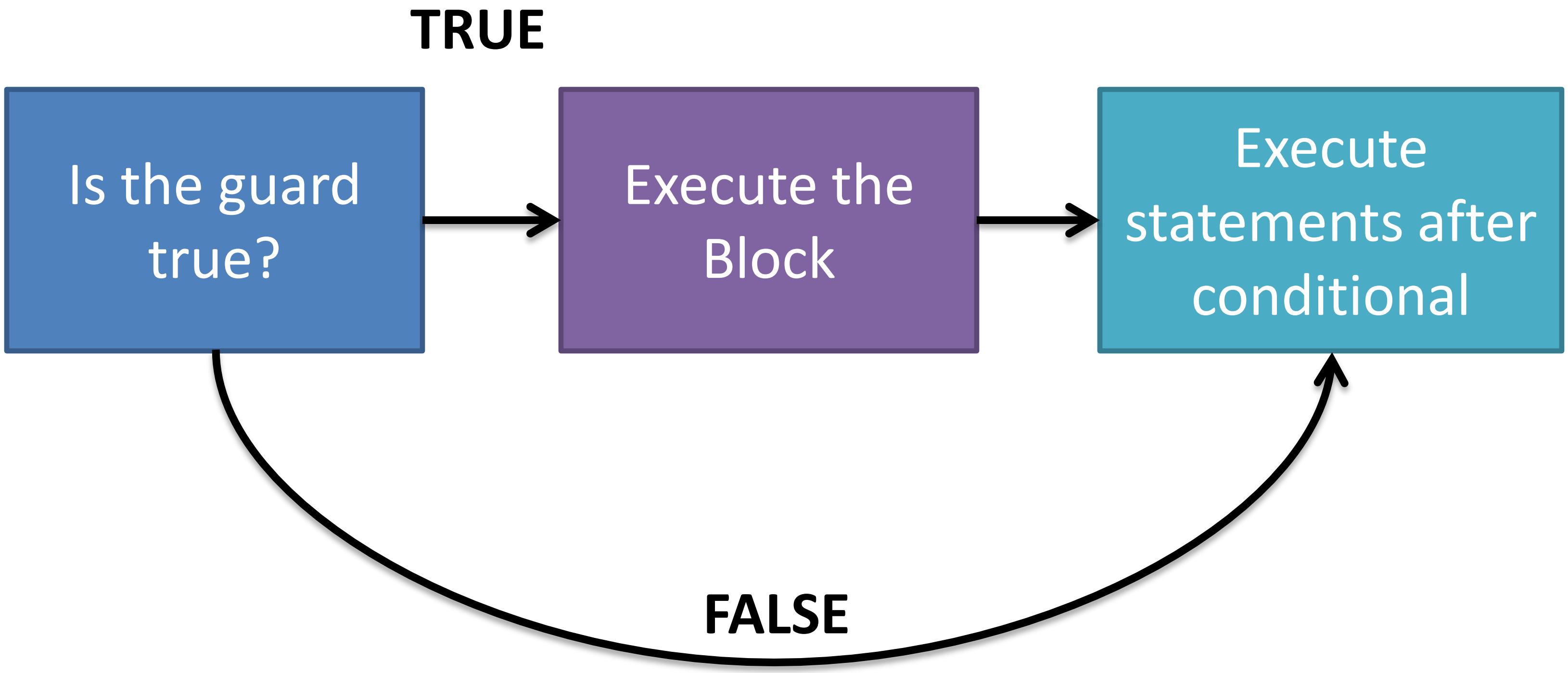


The *test* or *guard*

```
if (<test>) {
  <statement>
  <statement>
  …
  <statement>
}
```

The statements to execute, the *body* or *block*

Semantics of Conditionals



Else Branches

```java
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratuations!");
} else {
  System.out.println("You need more money!");
}
```

"Execute this block if we don't go into the first block" (i.e., when savings is less than or equal to 0).

Else-if Branches

```java
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratuations!");
} else if (savings > 50) {
  System.out.println("That's decent.");
} else {
  System.out.println("Need more!");
}
```

"Else" = if the previous guard fails, try this one!

Relational Operators

```
  >         /* greater than */
  <         /* less than */
  >=        /* greater than or equals */
  <=        /* less than or equals */
  ==        /* equals */
  !=        /* not-equals*/

// Syntax: <expr> <op> <expr>, e.g., 1 != 2
```

- Only works on primitive data.
  - We'll discuss what to do for objects, e.g., Strings, later.

Operator Precedence

```
    |     ++, --, +, -                // Unary operators
    |     *, /, %                     // Multiplication operators
    |     +, -                        // Addition operators
    |     <, >, <=, >=                // Relational operators
    |     ==, !=                      // Equality operators
    |     =, +=, -=, *=, /=, %=       // Assignment operators
    V
  Lower
Precedence
```

Mutually exclusive branches

```java
if (savings < 100) {

}
if (savings >= 50) {

}
if (savings == 75) {

}
```

```java
if (savings < 100) {

} else if (savings >= 50) {

} else if (savings == 75) {

}
```

```java
double savings = 75;
```

`else if` gives you *true mutually exclusive branches*.

Object Equality

```
Scanner in = new Scanner(System.in);
String s = in.nextLine();
// Will never be true.  == only works for primitive types.
if (s == "yes") {
  System.out.println("s1 is yes!");
}
// Need to use the equals method to check equality for objects.
if (s.equals("yes")) {
  System.out.println("s1 is really yes!");
}
```

Multiple Conditions

```java
Scanner in = new Scanner(System.in);
String name = in.nextLine();
double amount = in.nextDouble();
// Logical AND: true if both conditions are true
if (name.equals("McScrooge") && amount > 1000) {
  System.out.println("Y U SO RICH!?");
}
// Logical OR: true if one of the conditions is true
if (name.equals("Peter") || amount < 10) {
  System.out.println("Y U SO POOR!?");
}
```

Cumulative Algorithms

Problem: Interactive Sum

- Can you write a program that computes the sum of numbers from 1 to the user's input?

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println(
  "Sum of 1 to " + n + " is " + sum + ".");
```

Interactive Sum Trace (1)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

Interactive Sum Trace (2)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

Interactive Sum Trace (3)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

in

…

Interactive Sum Trace (4)

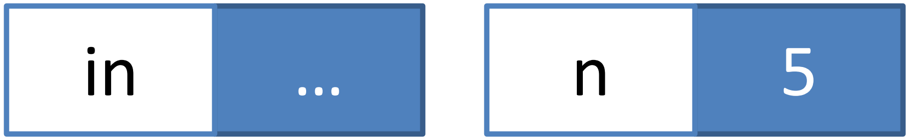
```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

in ...

n 5

Interactive Sum Trace (5)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```



in | ...

n | 5

Interactive Sum Trace (6)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

in	…	n	5	sum	0

Interactive Sum Trace (7)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	0
in	…
n	5
sum	0

Interactive Sum Trace (8)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	0
in	…
n	5
sum	0

Interactive Sum Trace (9)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i	1
in	…
n	5
sum	0

Interactive Sum Trace (10)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	1
in	…
n	5
sum	0

Interactive Sum Trace (11)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	1
in	…
n	5
sum	<u>1</u>

Interactive Sum Trace (12)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	2
in	…
n	5
sum	1

Interactive Sum Trace (13)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i	2
in	…
n	5
sum	3

Interactive Sum Trace (13)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	**3**
in	…
n	5
sum	3

Interactive Sum Trace (14)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

i	3
in	…
n	5
sum	6

Interactive Sum Trace (15)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i	4
in	…
n	5
sum	6

Interactive Sum Trace (16)

```
01  Scanner in = new Scanner(System.in);
02  System.out.print("n? ");
03  int n = in.nextInt();
04  System.out.println();
05
06  int sum = 0;
07  for (int i = 0; i < n; i++) {
08    sum += i;
09  }
10 System.out.println("Sum of 1 to " +
      n + " is " + sum + ".");
```

i	4
in	…
n	5
sum	10

Interactive Sum Trace (17)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i	<u>5</u>
in	…
n	5
sum	10

Interactive Sum Trace (18)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
  n + " is " + sum + ".");
```

in	…	n	5	sum	<u>10</u>