

# PC4G - University of Waterloo

December 8, 2017

Alice Tutorial  
9:00 - 10:30



Peter McAsh  
@pmcash



BASED ON THE UNTOLD TRUE STORY

MEET THE WOMEN YOU DON'T KNOW,  
BEHIND THE MISSION YOU DO.

# HIDDEN FIGURES



1.13.17

HIDDENFIGURESMOVIE.COM

Source: <https://www.foxmovies.com/movies/hidden-figures>















New Yorker  
April 18, 2015



# Applications vs Programming



## Applications

- make devices useful
- similar applications have similar functionality

## Programming

- instructions that create applications
- concepts are the same in all programming languages

# Applications vs Programming



## Applications

- production:
  - word processing, spreadsheets, presentations
- web browsers:
  - Chrome, Safari, Firefox
- Social Media:
  - Facebook, Twitter, Snapchat, Instagram
- Games:
  - Angry Birds, Solitaire

## Programming

- COBOL, FORTRAN, BASIC
- Java, C, PHP, Python, Ruby, SQL, JavaScript, HTML

# Programming Languages



## Hello World - COBOL

```
//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELOWRLD EXEC COBUCLG,PARM.COB='MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
0065 SPECIAL-NAMES.
0066     CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING-STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100     DISPLAY HELLO-CONST UPON CNSL.
110     STOP RUN.
//LKED.SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//          DD DSN=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//
```

# Programming Languages



## Hello World - FORTRAN

A screenshot of the 'Force 2.0' FORTRAN editor window. The window title is 'Force 2.0 - [Source1.f]'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Run', 'Options', 'Tools', 'Window', and 'Help'. The toolbar contains icons for file operations (new, open, save, print, delete), editing (undo, redo, copy, paste), and execution (run, stop, step through). The main text area shows a FORTRAN program with line numbers 1 through 4 in the left margin. The code is: 

```
1 PROGRAM helloWorld
2 PRINT *, 'Hello World!'
3 WRITE(*,*) 'Hello World!!!'
4 END
```

 The status bar at the bottom indicates '4: 11 Modified Insert'. A file explorer on the left shows 'Source1.f'.

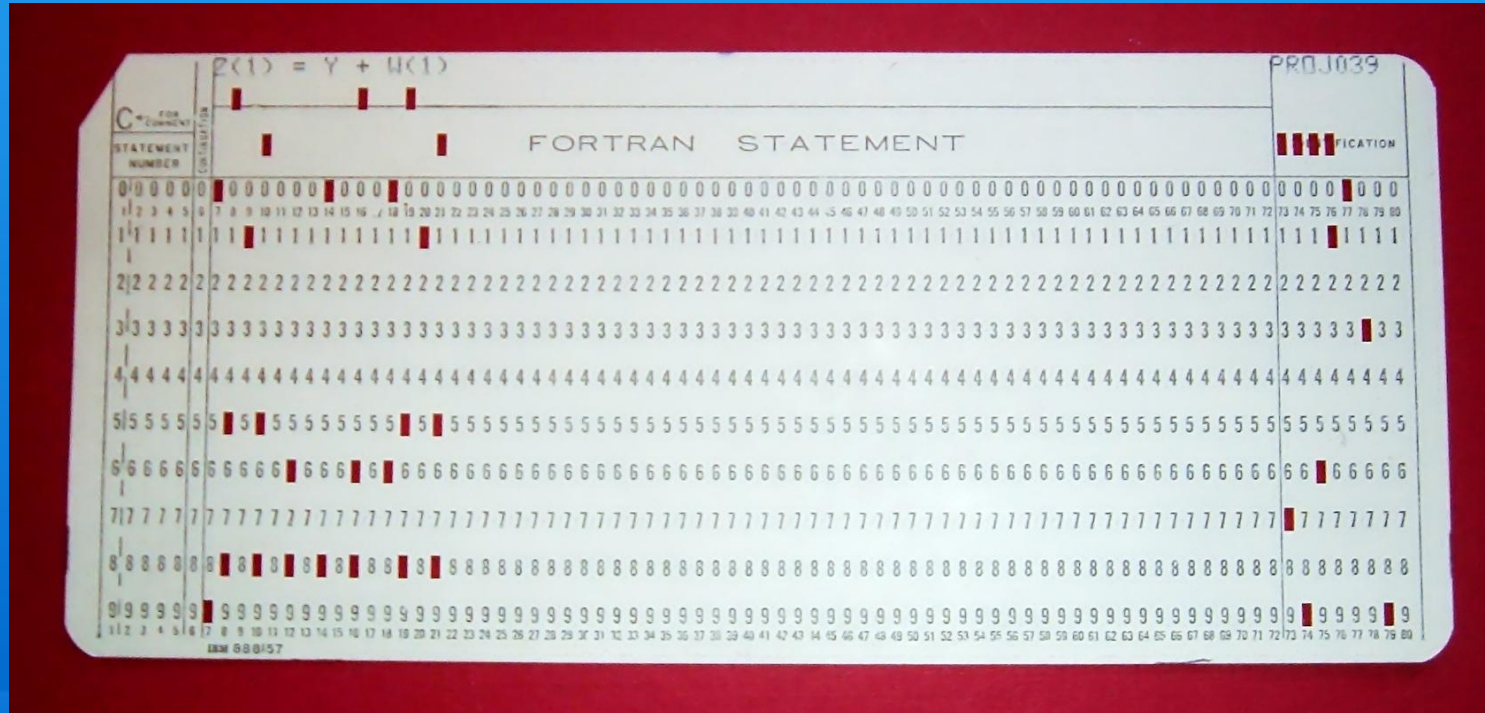
Source: [http://web.ics.purdue.edu/~cs154/lectures/lecture024\\_files/lectur1.jpg](http://web.ics.purdue.edu/~cs154/lectures/lecture024_files/lectur1.jpg)



# Programming Languages



## Hello World - FORTRAN - punch card



# Programming Languages



## Hello World - BASIC

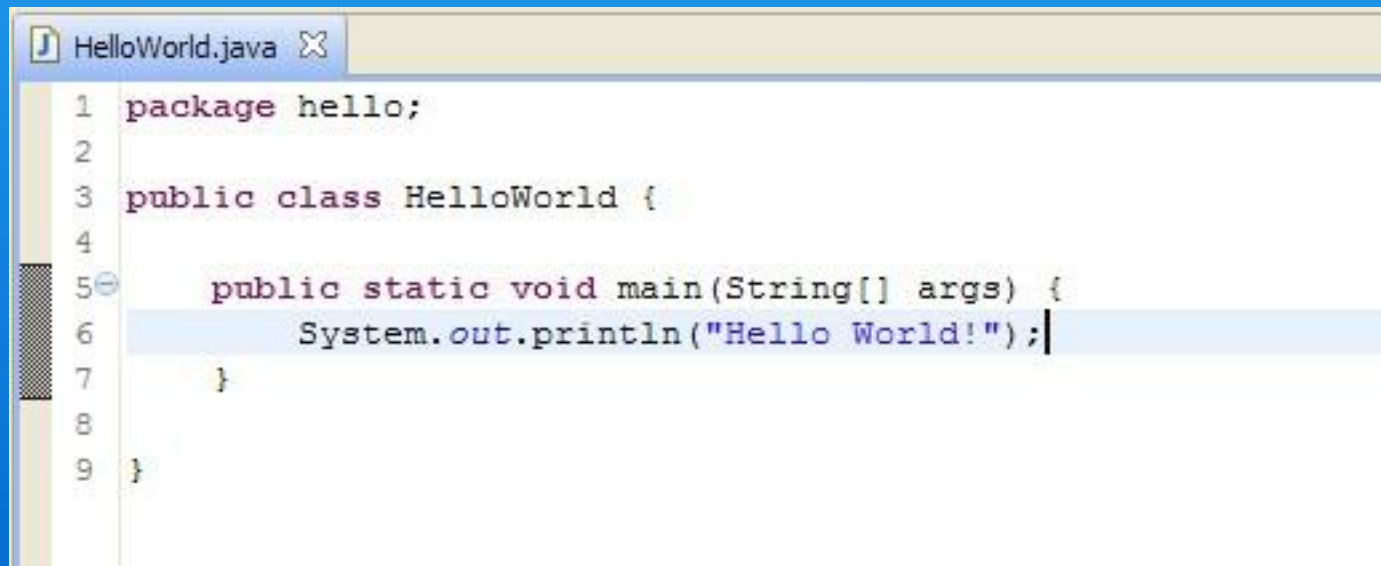
A screenshot of a BASIC interpreter window. The window has a title bar with three small circles on the left and the text 'XL/XE 128K' on the right. The main area of the window is dark blue with white text. The text shows the following sequence of commands and output:

```
READY
LIST
10 PRINT "Hello World!"
READY
RUN
Hello World!
READY
█
```

# Programming Languages



## Hello World - Java

A screenshot of a Java IDE window titled 'HelloWorld.java'. The code is as follows:

```
1 package hello;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World!");
7     }
8
9 }
```

# Programming Languages



## Hello World - Python

```
hello.py
print "Hello World!"
```

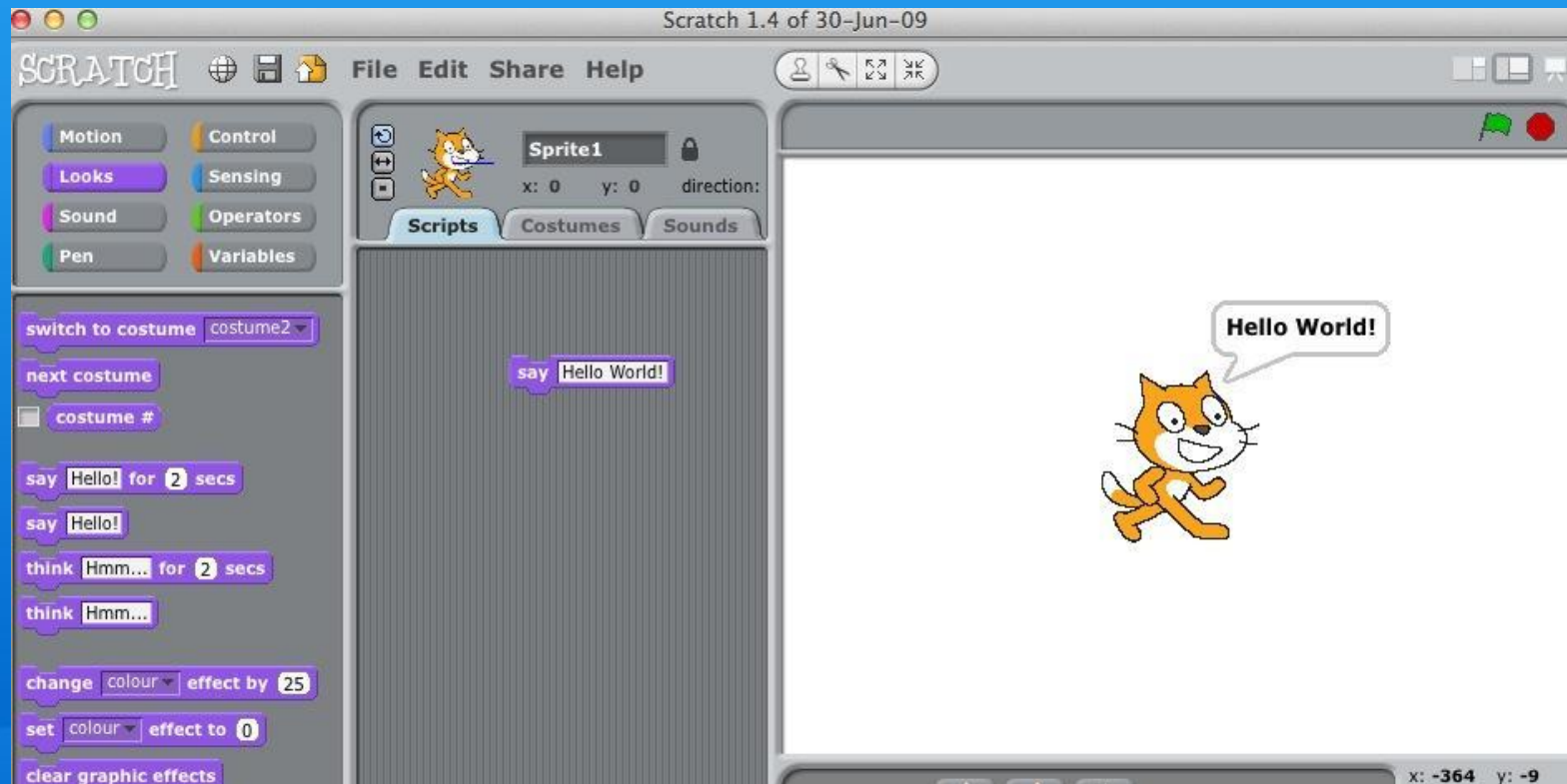
Source: <https://static.digit.in/fckeditor/uploads/file/python-hello%20world.png>



# Programming Languages



## Hello World - Scratch

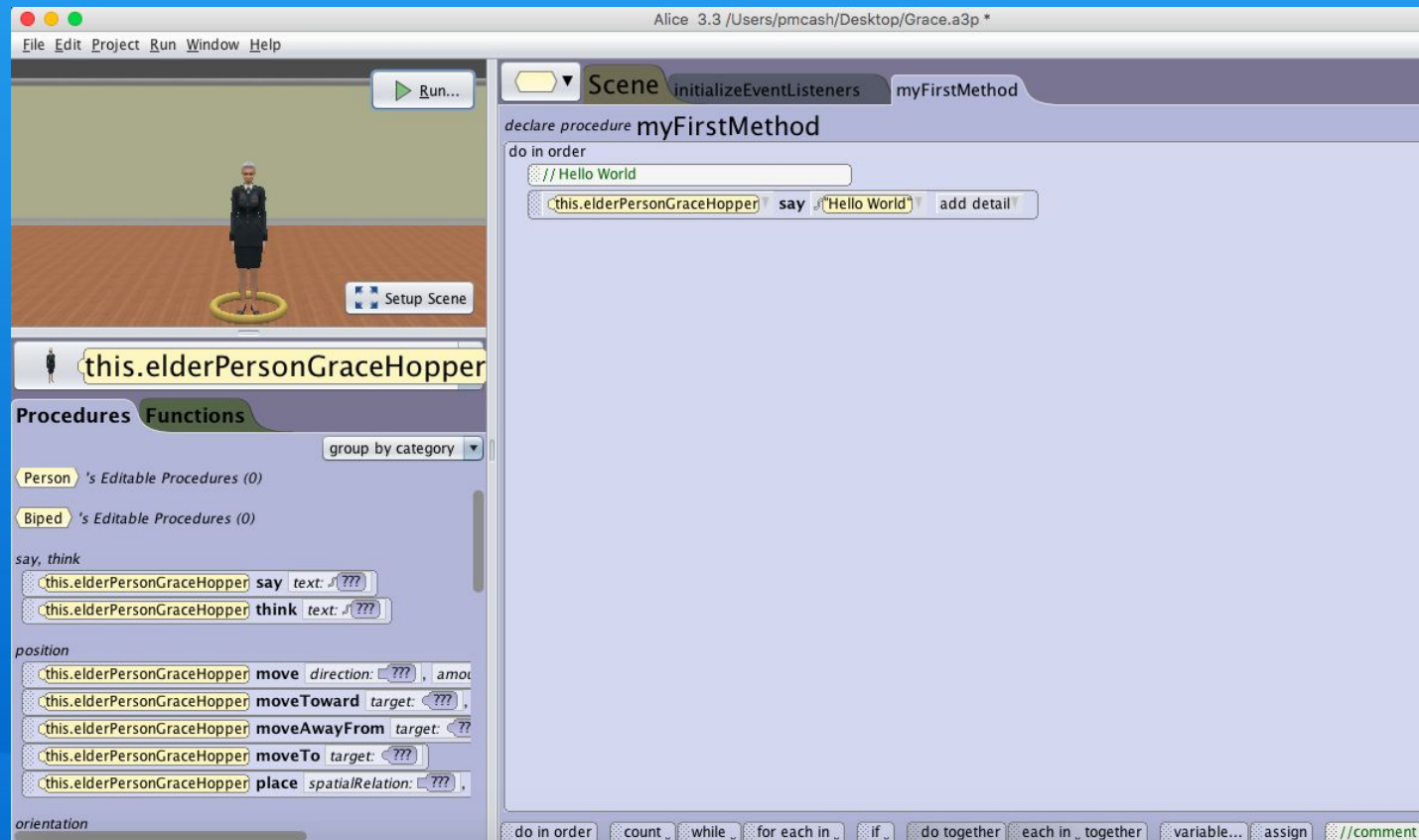


Source: <http://scratch.duncanmoran.net/helloworld.jpg>

# Programming Languages



## Hello World - Alice3



# Programming Languages

Alice is a programming language developed at Carnegie Mellon University designed to teach programming concepts.



# Alice 3 - Exercise #1 - Penguin Bowling



## Exercise #1 - Penguin Bowling

- Problem: A penguin is bowling on the arctic ice cap. The penguin pushes a bowling ball towards a bowling pin, knocking the pin over. Surprisingly, the pin stands up, and pushes the bowling ball back at the penguin. The penguin is knocked over.
- start Alice on your laptop
- File System (tab) → browse...
- navigate to the thumb drive and select PenguinBowling.a3p
- click OK
- click Run





# Alice 3 - Exercise #1 - Penguin Bowling



## Exercise #1 - Penguin Bowling

- Storyboard



Penguin pushes the ball



Ball strikes the pin  
and the pin falls  
over with a "Thud"



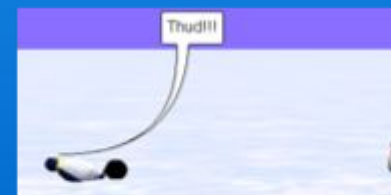
Pin stands up



Pin pushes the ball



Ball strikes the  
penguin



Penguin falls down  
with a "Thud"

# Alice 3 - Exercise #1 - Penguin Bowling



## Exercise #1 - Penguin Bowling

- Alice 3 program code solution

The screenshot displays the Alice 3.3 software interface. The top window shows a 3D scene of a penguin on an ice rink, a bowling ball, and a bowling pin. The bottom window shows the code for the scene, organized into tabs: Scene, initializeEventListeners, and myFirstMethod.

**Scene Tab:**

- Scene:** initializeEventListeners, myFirstMethod
- Procedures:** Penguin's Editable Procedures (0), Flyer's Editable Procedures (0)
- Functions:** say, think, position, orientation

**myFirstMethod Tab:**

```
declare procedure myFirstMethod
do in order
  // Exercise #1 - Penguin Bowling
  // PC4G - University of Waterloo - 2017 12 08
  // Peter McAsh

  // Scene created by D Slater PC4G
  // Exercise adapted from Anne Philpott PC4G
  // Penguin pushes the ball
  this.penny move FORWARD, {this.penny getDistanceTo this.ball} - 0.2, add detail
  // Ball moves and strikes the pin
  this.ball move FORWARD, {this.ball getDistanceTo this.bowlingPin} - 0.2, add detail
  // The pin falls over with a "Thud"
  this.bowlingPin turn BACKWARD, 0.2, add detail
  this.bowlingPin say "Thud", add detail
  // The pin gets back up and pushes the ball at the penguin
  this.bowlingPin turn FORWARD, 0.2, add detail
  this.bowlingPin move BACKWARD, 0.5, add detail
  this.bowlingPin move FORWARD, 0.5, add detail
  // Ball moves and strikes the penguin
  this.ball move BACKWARD, {this.ball getDistanceTo this.penny} - 0.2, add detail
  // The penguin falls over with a "Thud"
  this.penny turn BACKWARD, 0.2, add detail
  this.penny say "Thud", add detail
```

# Alice 3 - Exercise #1 - Penguin Bowling



Your turn!

- Learn through experimentation
- Drag and drop things
- Click on things and try making changes
- There is always “undo” - or you can close and load the original program again
- After you make a change - “Run” to see what effect your changes have had



# Programming Concepts

- 4 steps in problem solving
  - Define
  - Design
  - Implement
  - Test
- Simple problems can often be solved with little or no formal design.
- As problem complexity increases, the need to design increases.
- When an individual reaches the point where she/he needs to design varies by individual.



# Programming Concepts

- Problem Definition:
  - Written description of the problem to be solved.
- Pseudocode:
  - A list of actions, in the proper sequence, required to solve the problem. They are very close to (but not quite) actual program code.
- Computer Program:
  - A set of computer instructions to complete a task.
- Bugs:
  - Errors in computer programs. The process of identifying and removing bugs is referred to as debugging a program.

# Programming Concepts

- Comments:
  - Statements within the computer program code.
  - Intended for computer programmers.
  - Explain what the program does.
  - A well written program includes comments.

# Programming Concepts

- All programs involve:

- Sequence

- Instructions are executed in the order they are found within the program code.

- Selection

- Execution of instruction(s) based on a condition being true or false.

- Repetition

- Repeated execution of a section of code, containing at least one instruction.



# Programming Concepts

- Object Oriented Programming

- Class:

- A particular kind of object.
    - Class names begin with a capital letter, no spaces, subsequent words are also capitalized.

- Object:

- An instance of a class.
    - Object names begin with a lower case letter, no spaces, subsequent words are capitalized.

- All objects of the same class share some commonality.

- Although each object belongs to a class, each object is unique in its own way.

# Programming Concepts

- Object Oriented Programming

- Methods:

- A sequence of instructions that will be carried out when requested.
    - Abstraction:
      - Once a method is written it allows us to think about an overall task instead of the the small actions that were needed to complete the task.
    - Parameter(s):
      - One or more pieces of information that are referred to the method.  
Example: the distance an item is to move.



# Alice 3 - Object Oriented Programming



Object

Method

Parameter

The screenshot shows the Alice 3.3 interface. The top window displays a 3D scene with a penguin on an ice floe and a black ball. Below the scene is a dropdown menu showing 'this.penny' as the selected object. The bottom-left pane shows the 'Procedures' tab with a list of objects and their editable procedures. The bottom-right pane shows the 'myFirstMethod' procedure editor, which contains a sequence of actions for the penguin, ball, and bowling pin. The title bar of the window reads 'Alice 3.3 /Users/pmcash/Dropbox/PC4G/2016/For The Girls/PenguinBowling.a3p'.

File Edit Project Run Window Help

Run...

Setup Scene

this.penny

Procedures Functions

group by category

Penguin's Editable Procedures (0)

Flyer's Editable Procedures (0)

say, think

this.penny say text: ???

this.penny think text: ???

position

this.penny move direction: ???, amount: ???

this.penny moveToward target: ???, amount: ???

this.penny moveAwayFrom target: ???, amount: ???

this.penny moveTo target: ???

this.penny place spatialRelation: ???, target: ???

orientation

Scene initializeEventListeners myFirstMethod

declare procedure myFirstMethod

do in order

// Exercise #1 - Penguin Bowling

// PC4G - University of Waterloo - 2017 12 08

// Peter McAsh

// Scene created by D Slater PC4G

// Exercise adapted from Anne Philpott PC4G

// Penguin pushes the ball

this.penny move FORWARD, this.penny getDistanceTo this.ball - 0.2 add detail

// Ball moves and strikes the pin

this.ball move FORWARD, this.ball getDistanceTo this.bowlingPin - 0.2 add detail

// The pin falls over with a "Thud"

this.bowlingPin turn BACKWARD, 0.2 add detail

this.bowlingPin say "Thud" add detail

// The pin gets back up and pushes the ball at the penguin

this.bowlingPin turn FORWARD, 0.2 add detail

this.bowlingPin move BACKWARD, 0.5 add detail

this.bowlingPin move FORWARD, 0.5 add detail

// Ball moves and strikes the penguin

this.ball move BACKWARD, this.ball getDistanceTo this.penny - 0.2 add detail

// The penguin falls over with a "Thud"

this.penny turn BACKWARD, 0.2 add detail

this.penny say "Thud" add detail

do in order count while for each in if do together each in together variable... assign //comment

Class

Methods (Alice Procedures)

# Alice 3 - Exercise #2



## Exercise #2 - Sequence Selection Repetition

- Problem: Three students and their teacher are in a classroom. The teacher asks the students “What are the 3 basic components of all programs?” The first girl answers “Sequence”, the second “Selection”, and the third “Repetition”. The teachers says “Bravo”. In celebration, all three girls jump up and down at the same time.



# Alice 3 - Exercise #2

## Exercise #2 - Sequence Selection Repetition

- Scene



# Alice 3 - Exercise #2



## Exercise #2 - Sequence Selection Repetition

- Pseudocode:

Teacher says "What are the 3 basic components of all programs?"

Student one says "Sequence"

Student two says "Selection"

Student three says "Repetition"

Teacher says "Bravo"

do together

Student one moves up

Student two moves up

Student three moves up

do together

Student one moves down

Student two moves down

Student three moves down

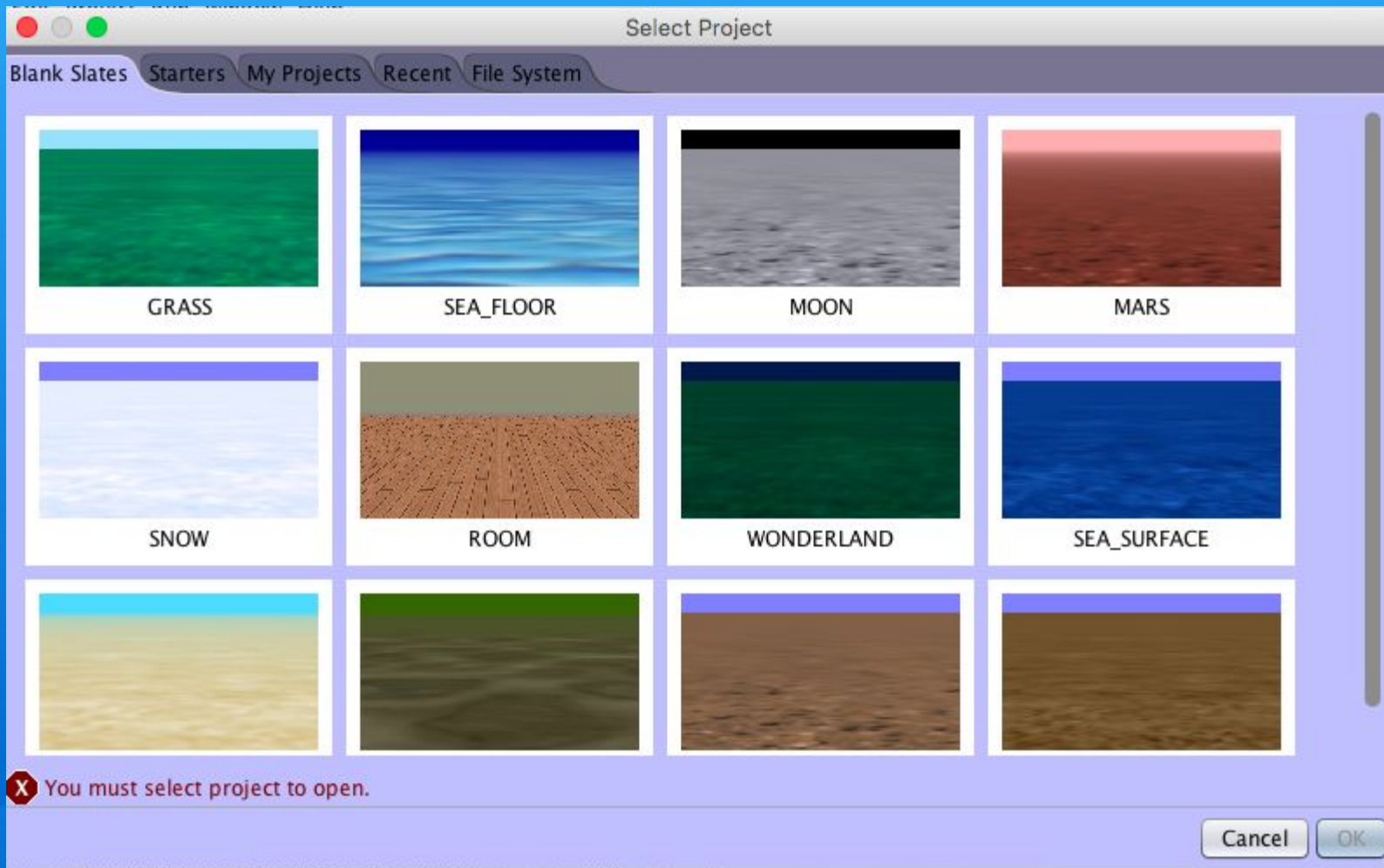
- navigate to the thumb drive and select  
SequenceSelectionRepetition.a3p





# Alice 3 - Creating your own scene

Start a new project:

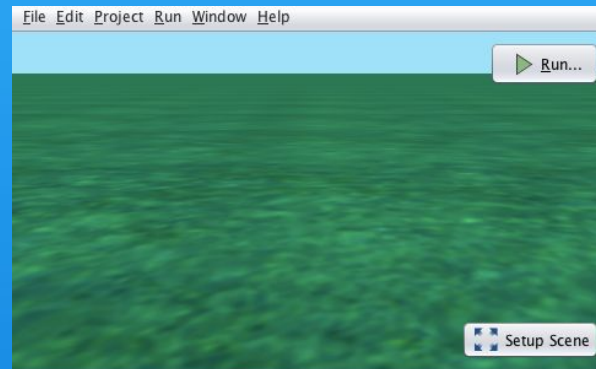




# Alice 3 - Creating your own scene



Click Setup Scene to add object(s) / edit the scene



100's of Classes to choose from to add objects to the scene



# Alice 3 - Creating your own scene



Add objects to the scene

- name starts with lower case, no spaces, capitalize subsequent words



# Alice 3 - Creating your own scene

Position objects in the scene



# Alice 3 - Creating your own scene

## Adjust camera position



Move Camera  
Up/Down Left/Right

Tilt Left

Tilt Right

Move Camera  
Forward/Back

Tilt Forward/Backward



# Alice 3 - Exercise #3



## Exercise #3 - Create your own project

- Create a new project
- Edit the scene to add objects and set the initial position of the camera
- Add code to myFirstMethod to make your scene come alive





# Alice 3 - Biped



## Biped Class

- objects consist of multiple parts
  - each part is an object
  - methods can work with the entire object OR with a specific part
- 
- start a new project and create an object from the Biped Class



# Alice 3 - Biped



## Biped Class

- spine
  - moves whole upper body - good for bows, bending over, etc.
- shoulder (left and right)
  - moves arms individually
- knee (left and right)
  - move leg below the knee
- hip (left and right)
  - move entire leg
- straightenOutJoints



# Alice 3 - Biped

## Biped Class

- navigate to the thumb drive and select Biped.a3p

## Biped Methods

- written by PC4G organizers



# Alice 3 - Last couple of things



Everything in a scene is an object

- manipulate anything you want with your code

Make an object invisible

- opacity - 1 is fully visible - 0 is invisible

setVehicle method

- object1 setVehicle object2
- when object2 moves, object 1 moves too
- movement of object1 has no impact on object2
- end the connection - assign a new vehicle to the object
  - object1 setVehicle this

Functions

- provide the answer to a question - example: distance to another object



# Alice 3 - Exercise #4



## Exercise #4 - Final Practice

- Final opportunity to practice before the “Challenges”
- work with  
Bipeds.a3p





# Challenges



There are 3 challenges:

- first two provide you with the problem and a storyboard to follow - the scene is created (you don't need to add any objects)
- 3rd challenge requires you to create a storyboard and then code your storyboard - the scene is created (you may wish to add objects)
- not all groups will get to third challenge
- appropriate procedures from PC4G organizers are included in all today's challenges

Include comments in your code

Save your work frequently

- Alice can crash
- this version of Alice has backups - better saved than sorry



New Yorker  
April 18, 2015