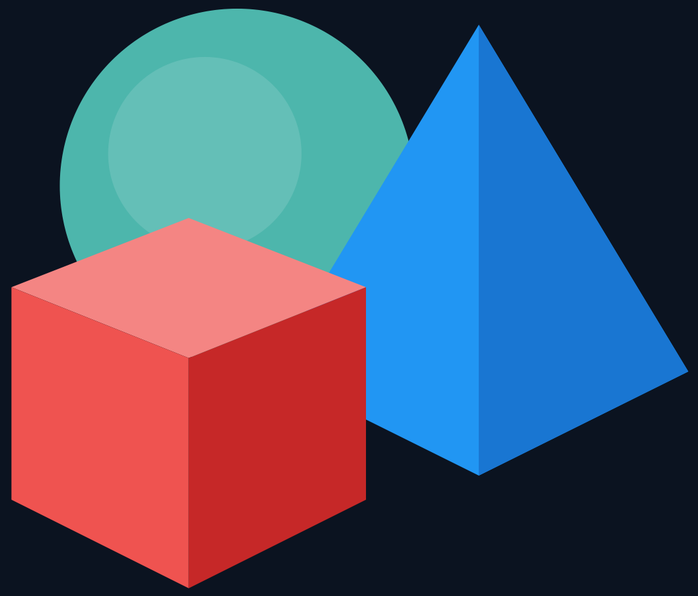
ACM

Teaching Objects-first In Introductory Computer Science

Authors:
Stephen Cooper,
Wanda Dann,
Randy Pausch

Youxin(Tony) Tan | 2023

Any oop struggler?



What’s Object?

- self-contained unit that represents a real-world entity or concept
- game character is an object that you control with methods, data, etc.

Object

- data and method that bundle with an object
- Alice has the method “Run”, “Speak”, etc.

Encapsulation

- Objects are instances of classes
- Elon is an instance of the class “Human”

Classes

Abstract

- Object-first strategy is becoming popular in intro computer science courses
- Paper disccusses the challenges of objects-fist strategy
- Present a new approach that attempt to solve the challenge posed
- Approach Centered on visualization of objects and their behaviors using 3D animation
- Students improved after implementing the approach
- Measured by statistical data and informal observations
- Disccussed comparison of this approach and other relavent work

Introduction

Popular CS1 Approaches in 2003 by ACM:

- Functional-first(Haskell, Racket)
- Objects-first(Java, C++)
- Imperative-first(C)

Classic approaches:

- Gentle learning curve
- Start with simple programs

Objects-First challenges imposed:

- More complex
- Need to learn classes and objects immediately
- Encapsulation(public/price data), method(constructors etc)
- Types, variables, values, references, syntax, functions, states
- Altogether

Problem

Higher Complexity

- Functional-first learns function first then states
- Imperative-first learns states first then function
- Object-first needs to grasp both simultaneously

Aims of the approach

- Reduce the complexity of details
- Provide design-first approach to objects
- Visualizing objects in meaningful context

Methodology

ALICE Program

- Developed by Randy Pausch from Carnegie Mellon University
- Similar to Scratch
- Provide instruction for objects to achieve certain goals
- ~ 2 weeks to a semester course

Method(Cont)

Student Programs

- Student adds 3D objects to virtual world
- Objects encapsulated with own data
- Objects with methods(like Jump, Dance, etc)
- Sample task: Kermit(Frog) hops over the ladybug

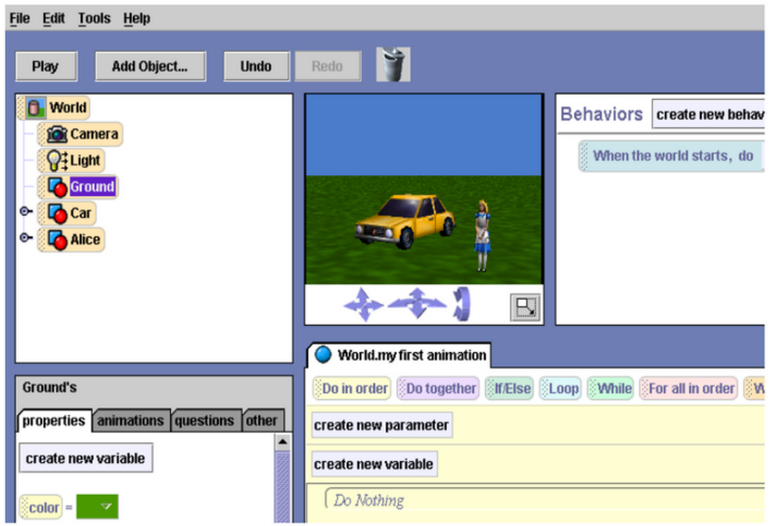


Figure 1. The Alice Interface

Figure 2. An initial scene in an Alice world

Method(Cont)

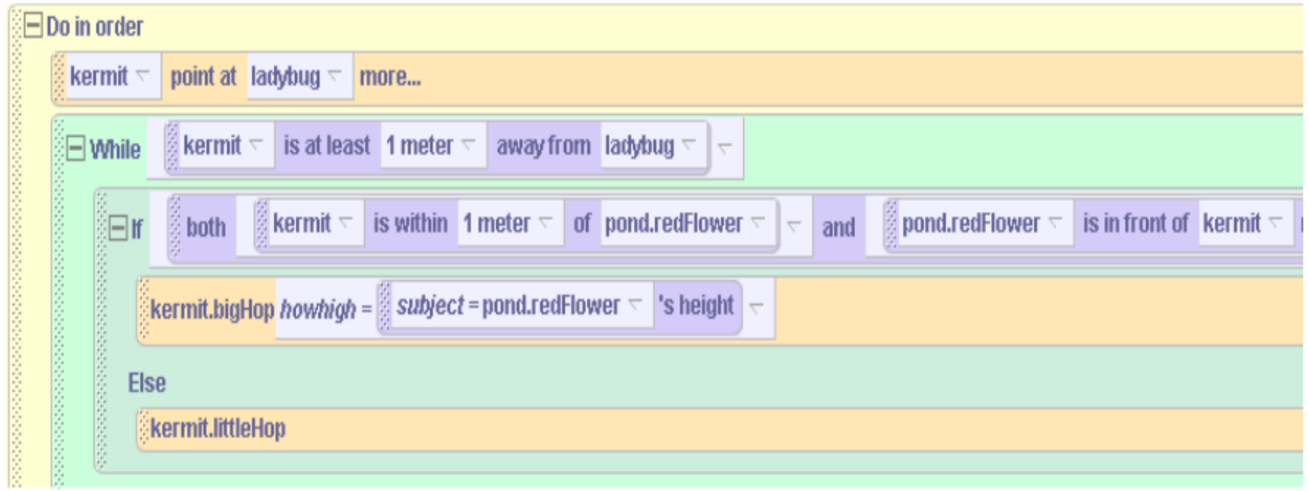


Figure 3. The code to have kermit hop over to the ladybug

Benefits

3D provides a sense of realism where students can receive meaningful context for understand classes, objects, methods and events.

Using Animated 3D Graphics to Prepare Novices for CS1(2003)

```
Getting Started
        * Why Alice?
        * Alice Virtual Worlds (Demo)
        * Using The Alice Interface

Types of Animations
        * DoInOrder vs. DoTogether
        * Movie vs. Interactive

Classes and Objects in a Virtual World
        * Object Properties and Parts
        * Orientation and Degrees of
Freedom
        * Motion in 3D Space

Building An Animation Program
        * Scenarios and Tasks
        * Problem Solving Concepts
        * Storyboards
        * Behavioral Methods
        * Parameters
        * Class vs. world methods
        * Vehicles
        * Events and Responses
        * Sound
Powerful motion techniques
        * Expressions
        * Decisions:  If … Else
Statements
        * Random Motion
        * MoveTo:  Location Vector

Repetition
        * Loop
        * Recursion
            * Recursive Function Call
            * Structural vs. Generative
        * While
        * Infinite Loops

Collections of Objects
        * A List of Objects
        * Sorting Objects

Projects
        * Collision Detection
```

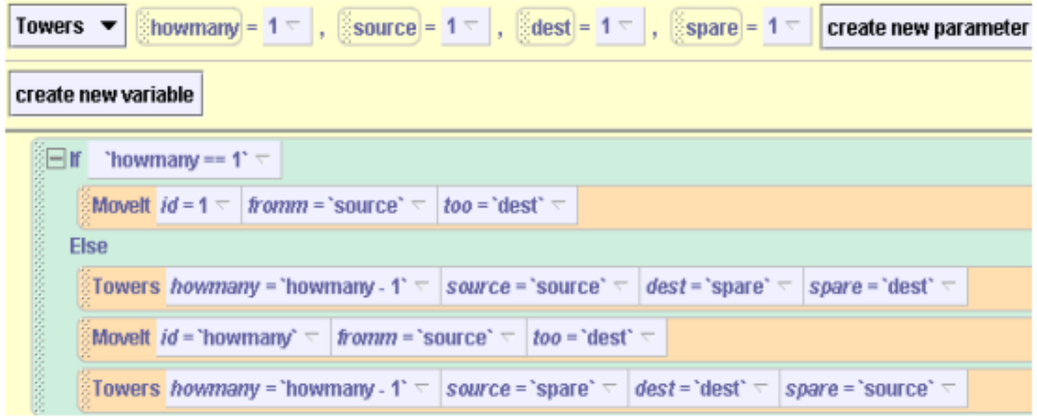


Figure 11. Towers of Hanoi Animation Method

- Full course for college student with no experience
- RHS is an example of teaching recursion
- Move everything from one tower to another
- Allow student to discovery the recursion pattern starting from base case

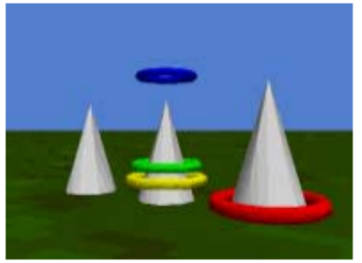


Figure 12. Towers of Hanoi Animation in Progress

Observations

Students developed:

- Strong sense of design
- Able to contextualize objects
- Appreciation of trial and error(each instruction cases visible change in the animation)
- Intuition concerning encapsulation(“move()” method can change an object’s position defined)
- Sense of inheritance(students write to create more powerful classes)
- An sense of program state(variables, memory, etc)

Observations

Weakness:

- Do not develop a detailed sense of syntax
- No support for mismatched braces, semicolons, etc
- Tho students tend to master them quickly after transition to C++/Java

Results

Statistics	All	Test	Control
# Students	49	11	10
Mean	2.49	2.8	1.3
Median	2.75	3	1.25
Variance	1.62	0.75	1.22

Table 1: Students taking Alice, 2001-2002

Control v.s. Test

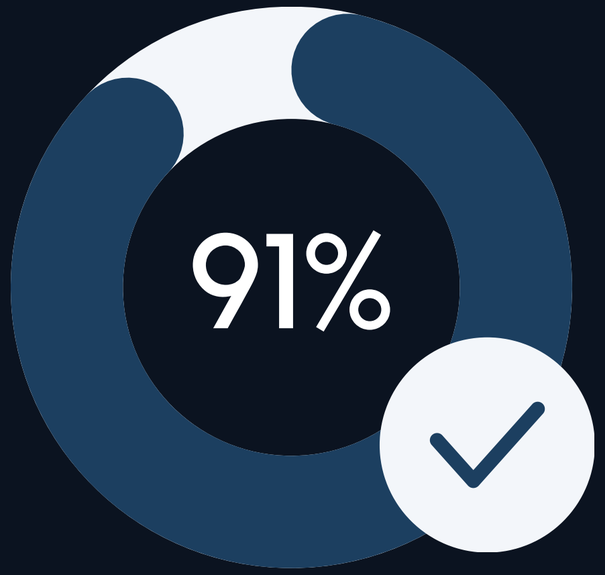
- Weakest 21 CS majors selected
- 10 excused out, 11 joined the test group
- GPA scale is out of 4

Interpretation

- Test group did better in CS1 than the total group(including the “stronger” CS majors
- Lower variance: less students performed poorly

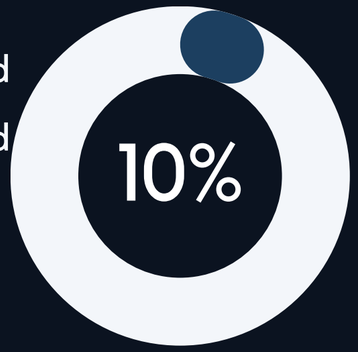
Results(CS 2 enrollment)

From test group



- 65% of all students took CS1 continued on to CS2
- 91% of students who participated the course with ALICE continued with CS2

Control Group



- Only 10% from the control group enrolled in CS2

Discussion & comparisons

1.

Syntax Complexity:

- Other similar graphics libraries for teaching requires student to write correct Java/C++ program from scratch
- BlueJ and JPT approaches still requires students to modify existing code with correct syntax but ALICE doesn't

2.

Cocurrency supports:

- ALICE does support concurrency where providing primitives for performing actions simultaneously(2 frogs jump at the same time)

3.

Examples development:

- Developing large collection of examples is time-consuming
- Larger size of examples helps with teaching(used as lecture materials, assignments questions, exam questions, instructional aids, etc)

Conclusion

- As per the authors, ALICE approach is the best way to aid in students understanding if the course is OOP in CS1
- Significantly increases CS major retention rate
- Improved weaker students' grade by a large extend
- ALICE presented an intuitive understanding of objects to students

Recommendations

1. Using Animated 3D Graphics to Prepare Novices for CS1: https://www.tandfonline.com/doi/epdf/10.1076/csed.13.1.3.13540?needAccess=true

2. Dann, W., Cooper, S., & Pausch, R. Using visualization to teach novices recursion. In Proceedings of the 6th annual conference on Innovation and Technology in Computer Science Education (Canterbury, England, June, 2001), 109-112

3. Dann, W., Cooper, S., & Pausch, R. Making the connection: programming with animated small worlds. In Proceedings of the 5th annual conference on Innovation and Technology in Computer Science Education (Helsinki, Finland, July, 2000), 41-44.

Thank You

For Your Attention