

GEORGIA INSTITUTE OF TECHNOLOGY
COLLEGE OF ARCHITECTURE
SCHOOL OF ARCHITECTURE

Design Scripting

ARCH 6502 MS

Semester Fall 2012
CRN Number 88048
Day/Time Tuesday/Thursday 12:05pm-1:25pm
Credit 3 Hours

Matthew Swarts
Research Scientist
Digital Building Lab

This is the syllabus for the Fall 2012 Design Scripting course taught by Matthew Swarts in the School of Architecture for Georgia Institute of Technology. This document outlines the methodology, grading, and schedule for the semester long course.

Course Methodology

Who should take this course?

Anyone with an interest in algorithmic design processes, unpacking of design relationships, building hierarchies of design objectives, design task automation, and rationalization of design thinking. This course offers you the opportunity to go beyond the restrictions of high level interfaces of common design tools and take command of the machine. Knowledge gained in this course will equip you with skills that are highly useful to your academic design explorations and professional future. This course presumes no prior knowledge in programming, and will take you along from basic computer science to modeling and simulation of complex systems through form genesis, simulation, and analysis. Those with prior experience in computer programming will also find the course helpful in exploring how to use algorithms for design.

What tools will be used?

There are three types of tools: software, languages, and techniques (algorithms).

Software: Rhino 4.0, Grasshopper 0.8, and Processing 1.5

Languages: VBScript, RhinoScript, Grasshopper, Java

Techniques: Parametric Function Graphs, L-Systems, Fractals, Shape Grammars, Cellular Automata, Agent-based Simulation, Line Integral Convolution, Circle Packing, Diffusion Limited Aggregation...

How will it be taught?

As much of the material depends upon writing text-based scripts, the majority of lectures are delivered through the Tegrity video lecture management system. The lectures are viewed as homework before class. Activities during class give you the opportunity to exercise the knowledge gained from the lectures. The instructor guides individuals during class as needed. Additionally there are web-based practice activities to be completed throughout the course. The midterm test covers basic knowledge of programming languages. The final project, proposed by each student, demonstrates a new technique for design genesis, simulation, and/or analysis.

Criteria for Grading

Student performance will be graded upon motivation and commitment in course procedures and in the quality of projects and presentations. Commitment entails attendance in all lectures, labs, on-time completion of requirements and participation in discussions. Grading is as follows:

"A" is reserved for work that is extremely sound and motivated; on outcome that clearly and consistently demonstrate inspired exploration of superior quality.

"B" grade signifies good work, i.e., work which clearly shows development although some problems may be noted. Work reflects a solid commitment to learning and an understanding of the issues.

"C" grade is satisfactory. It signifies work that meets the basic goals of the exercises, is presented in a complete manner satisfying the requirements and contains no serious errors of judgment or omission. In addition to that attendance, engagement in the course and timely completion of work is expected.

"D" denotes unsatisfactory work, not passing; obvious shortcomings include but are not limited to any of the above; failure to complete assignments.

"F" denotes failing work; low level attendance.

"I" (Incomplete) will be given only within Institute policy.

Lab Submissions	50%
Online Practice Problems	05%
Attendance	05%
Midterm	15%
Final Project	25%

Course Requirements:

- Attendance for In-class Lab
- In-Class Lab
- Online Practice Problems
- Video Lectures
- Midterm Exam
- Final Project Presentation

Most lab submissions will be due at 5pm after the next class. Submissions will be accepted up until 10pm with a 10% late penalty. Submissions after this time will not be accepted without prior email approval.

Course References

Design Scripting	http://designscripting.gatech.edu/
College of Architecture	http://coa.gatech.edu/
Rhino	http://www.rhino3d.com/
Monkey	http://wiki.mcneel.com/developer/monkeyforrhino4
RhinoScript	http://wiki.mcneel.com/developer/rhinoscript101
VBScript	http://msdn.microsoft.com/en-us/library/t0aew7h6(v=vs.85).aspx
Grasshopper	http://www.grasshopper3d.com/
	http://www.grasshopper3d.com/page/library-algorithms-and
VB.NET	http://msdn.microsoft.com/en-us/vbasic/ms789056
Processing	http://processing.org/
Java	http://www.oracle.com/technetwork/java/index.html
Online Coding	http://www.codingbat.com
GT VPN	http://www.coa.gatech.edu/coa/resources/network/offcampus
CoA Virtual Lab	https://coavlab.gatech.edu/
CoA Building Access	http://www.coa.gatech.edu/coa/resources/buildings/access_request

Online Coding Problem Sets

The online problem sets are performed through Nick Parlante's CodingBat site. Students will need to create an account on www.codingbat.com and link their TeacherShare in their account preferences to:

matthew.swarts@coa.gatech.edu

Additionally the list of problems are located here:

http://codingbat.com/home/matthew.swarts@coa.gatech.edu/designscripting_fall2012

They must be completed by the end of the semester (Saturday, December 15th, 2012 at 1:00am).

The following is a list of the problems and a suggested schedule of completion:

Suggested Order	Category	Problem Title	Problem ID	Suggested Date for Completion
1	Warmup-1	sleepIn	p187868	08/31/2012
2	Warmup-1	diff21	p116624	08/31/2012
3	Warmup-1	intMax	p101887	08/31/2012
4	Warmup-1	stringE	p173784	08/31/2012
5	String-1	helloName	p171896	09/07/2012
6	String-1	endsLy	p103895	09/07/2012
7	String-1	firstTwo	p163411	09/07/2012
8	String-1	withoutX	p151940	09/07/2012
9	Logic-1	caughtSpeeding	p157733	09/14/2012
10	Logic-1	teaParty	p177181	09/14/2012
11	Logic-1	twoAsOne	p113261	09/14/2012
12	Logic-1	greenTicket	p120633	09/14/2012
13	Array-1	sum3	p175763	09/21/2012
14	Array-1	makePi	p167011	09/21/2012
15	Array-1	makeEnds	p101230	09/21/2012
16	Array-1	maxTriple	p185176	09/21/2012
17	Logic-2	makeBricks	p183562	09/28/2012
18	Logic-2	loneSum	p148972	09/28/2012
19	String-2	countHi	p147448	10/05/2012
20	String-2	xyBalance	p134250	10/05/2012
21	Array-2	countEvens	p162010	10/12/2012
22	Array-2	centeredAverage	p136585	10/12/2012
23	String-3	maxBlock	p179479	10/19/2012
24	Array-3	seriesUp	p104090	10/26/2012
25	Recursion-1	factorial	p154669	11/02/2012

Class Lab Schedule

Week 01 Class 1 - Tuesday 2012.08.21

Rhino Modeling

Lecture	Lab
Geometry Rhino: Interface Rhino: Point Curve Surface Rhino: Boolean Rhino: Sweep Rhino: Section	Create castle from Rhino Tutorials. Create Flashlight from Rhino Tutorials.

Week 01 Class 2 - Thursday 2012.08.23

Rhino Modeling

Lecture	Lab
Rhino: Work Plane Rhino: History Rhino: STL 3D Print Preparation Rhino: Rendering	Create rubber ducky from Rhino Tutorials.

Week 02 Class 1 - Tuesday 2012.08.28

Rhino Macros

Lecture	Lab
1D Periodic Pattern 2D Periodic Pattern Symmetry – Reflection + Rotation Macros Rhino Macros Custom Rhino Macro Button	Create macro button that creates a snowflake pattern using a macro.

Week 02 Class 2 - Thursday 2012.08.30

Rhinoscript

Lecture	Lab
Monkey Interface Rhinoscript: Structure VBScript: Declaring Variables Rhinoscript: Create 3D Points Rhinoscript: Point Array VBScript: Random Generator User Input	Create a user defined point. Create a random point.

Week 03 Class 1 - Tuesday 2012.09.04

Rhinoscript

Lecture	Lab
Parametric Functions VBScript: For-loops Rhinoscript: Create Line from points Rhinoscript: Create Curve from points Rose Curve Function Lissiduous Curve Function	Create a lissiduous curve. Create a rose curve.

Week 03 Class 2 - Thursday 2012.09.06

Rhinoscript

Lecture	Lab
Rhinoscript: Create 3D surface of points Rhinoscript: Create Surface from Curves Rhinoscript: Create points on surface	Create an array of rose curves between a given range and step size from user input. Create a 3D surface of points. Create a swept surface with two curves from a set of points. Create points on a surface.

Week 04 Class 1 - Tuesday 2012.09.11

Processing

Lecture	Lab
Processing: Interface Processing: Sketch Structure Java: Variables Java: Syntax Processing: Canvas Processing: Draw Lines Processing: Draw Rectangles Processing: Draw Circles. Java: Random Processing: Color Processing: Stroke Processing: Fill Java: For-loops	Draw random points. Draw random lines. Draw random rectangles. Draw random circles. Draw grid of rectangles.

Week 04 Class 2 - Thursday 2012.09.13

Processing

Lecture	Lab
Simple Cellular Automata Processing: Pixels Java: If-then Java: Case-Switch Java: Modulus 1 Dimensional Tiling Processing: Key Press Processing: Save Image	Create a random Simple CA start row. Calculate 100 generations of a Simple CA in pixels. Export .png file on keypress.

Week 05 Class 1 - Tuesday 2012.09.18

Processing

Lecture	Lab
2D Cellular Automata Java: Arrays Processing: Create and Save Video	Draw a 2D array of square cells with random initial conditions. Run basic 2D Cellular Automata for 100 iterations. Run custom 2D Cellular Automata for 100 iterations. Save video of 2D Cellular Automata for 100 iterations.

Week 05 Class 2 - Thursday 2012.09.20

Rhinoscript

Lecture	Lab
3D Cellular Automata VBScript: Arrays VBScript: Multidimensional Arrays Rhinoscript: Create Box Rhinoscript: Color	Create 3D array of boxes. Create 3D array of on/off values with random initialization. Run 100 iterations of a 3D Cellular Automata

Week 06 Class 1 - Tuesday 2012.09.25

Rhinoscript

Lecture	Lab
VBScript: Recursion Menger Sponge VBScript: If-Then-Else VBScript: Subroutine VBScript: Function	Create Sierpinski Carpet. Create Menger Sponge with Iteration. Create Menger Sponge with Recursion.

Week 06 Class 2 - Thursday 2012.09.27

Rhinoscript

Lecture	Lab
L-Systems VBScript: String Manipulation Moore Curve Sierpinski Triangle Dragon Curve	Create Moore Curve of at least 4 generations. Create Sierpinski Triangle of at least 4 generations. Create Dragon Curve of at least 4 generations. Use color to identify generations.

Week 07 Class 1 - Tuesday 2012.10.02

Rhinoscript

Lecture	Lab
Subdivision Triangle Centroid Rhinoscript: Geometry Input Selection	Subdivide a rectangle until a certain area size. Subdivide a triangle until a certain area size.

Week 07 Class 2 - Thursday 2012.10.04

Rhinoscript

Lecture	Lab
Shape Grammars Rhinoscript: Case-Switch Iceray Shape Grammar	Optional:(Create an Iceray from an initial rectangle.)

Week 08 Class 1 - Tuesday 2012.10.09

Midterm

Lecture	Lab
	Midterm Test on VBScript

Week 08 Class 2 - Thursday 2012.10.11

Processing

Lecture	Lab
Processing: ArrayList Processing: Objects Kinematics	Add circles to the canvas and store them in an arraylist.

Week 09 Class 1 - Tuesday 2012.10.16

No Class

Lecture	Lab
None	None

Week 09 Class 2 - Thursday 2012.10.18

Processing

Lecture	Lab
Collision Dectection Circle Packing Diffusion Limited Aggregation	Pack the circles using iterative collision detection. Create DLA Application with 2D Circles.

Week 10 Class 1 - Tuesday 2012.10.23

Grasshopper

Lecture	Lab
Grasshopper: Interface Grasshopper: Points Grasshopper: Series Grasshopper: Curves Grasshopper: Function Grasshopper: Sliders	Create Rose Curve. Create Lissiduous Curve.

Week 10 Class 2 - Thursday 2012.10.25

Grasshopper

Lecture	Lab
Attractors Vector Fields Grasshopper: Vector Grasshopper: Circles Grasshopper: Min/Max	Create a 2D array of points. Create a 2D array of circle. Create an attractor point to affect circle size. Create and display vector field.

Week 11 Class 1 - Tuesday 2012.10.30

Grasshopper

Lecture	Lab
Diagrid Grasshopper: Loft Surface Grasshopper: Subdivision Surface Grasshopper: Pipe	Create curves from points. Create surface from curves. Create Diagrid from surface.

Week 11 Class 2 - Thursday 2012.11.01

Grasshopper

Lecture	Lab
Grasshopper: Morph Box	Create Morph Box on façade surface

Week 12 Class 1 - Tuesday 2012.11.06

Grasshopper Script Module

Lecture	Lab
VB.NET Grasshopper: Script Modules VB.NET: Create Points VB.NET: Create Lines VB.NET: Create Curves	Create Rose Curve. Create Lissiduous Curve.

Week 12 Class 2 - Thursday 2012.11.08

Grasshopper Script Module

Lecture	Lab
Catenary Curve Hooke's Law of Elasticity for Springs Grasshopper: Timer	Create module to iteratively solve a catenary curve between two 3D points given a curve length and elasticity.

Week 13 Class 1 - Tuesday 2012.11.13

Grasshopper Script Module

Lecture	Lab
Ecotect: Solar Incidence Analysis Grasshopper: Meshing Meshes Rhino: Export Excel Manipulation	Create a façade with holes that adjust based on analysis output.

Week 13 Class 2 - Thursday 2012.11.15

Grasshopper Script Module

Lecture	Lab
Vector Fields Line Integral Convolution Grasshopper: Bitmaps	Convert an image to a vector field with R and G channels. Create integrated curves from vector fields at each point in the vector field. Try a 3D Vector Field

Week 14 Class 1 - Tuesday 2012.11.20

Grasshopper Galapagos

Lecture	Lab
Galapagos: Interface Genetic Algorithms Simulated Annealing	Find the highest peak on a 2D surface using simulated annealing. Find the point on a surface with the highest peak and the highest slope or curvature.

Week 14 Class 2 - Thursday 2012.11.22

Holiday

Week 15 Class 1 - Tuesday 2012.11.27

Final Help

Week 15 Class 2 - Thursday 2012.11.29

Final Help

Week 16 Class 1 - Tuesday 2012.12.04

No Class

Week 16 Class 2 - Thursday 2012.12.06

No Class

Week 17 Final Tuesday 2012.12.11

Final Presentations 11:30am-5:40pm (Exact time to be determined)

Notes

Students with disabilities requiring special accommodations must obtain an accommodations letter from the ADAPTS Office [www.adapts.gatech.edu] to ensure appropriate arrangements.

Georgia Tech aims to cultivate a community based on trust, academic integrity and honor. Students are expected to act according to the highest ethical standards. For policy information on Georgia Tech's Academic Honor Code, please see [http://www.catalog.gatech.edu/rules_regulations/#18].

All cell phones should be turned off during class and when entering the classroom. If a cell phone rings with an incoming call or text message, the instructor will answer it.

In case of emergency (i.e. fire, accident, criminal act), please call the Georgia Tech Police at 894-2500. Please note that Perry Minyard, IT Support Administrator is also a firefighter and an Emergency Medical Technician (EMT) certified in performing CPR.