ARCH A163: 3-D MODELING: RHINO 2

Item	Value
Curriculum Committee Approval Date	12/04/2024
Top Code	020100 - Architecture and Architectural Technology
Units	2 Total Units
Hours	72 Total Hours (Lecture Hours 18; Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Open Entry/Open Exit	No
Grading Policy	Standard Letter (S)

Course Description

This course introduces fundamental skills of coding and 3-D computational design using Rhino software with additional plug-ins. Hands-on instruction will focus on parametrically modeling and testing design variations with rectilinear and non-rectilinear geometry, including preparing files for fabrication and presentation. Students should have basic knowledge of 3-D modeling. ADVISORY: ARCH A162. Transfer Credit: CSU.

Course Level Student Learning Outcome(s)

1. Students will be able to computationally design and visually present design projects with complex geometry using 3-D modeling software and scripting plug-ins at an entry-level professional quality, as evaluated by the instructor.

Course Objectives

- 1. Set up a new file, install plug-ins, and get started
- 2. Computationally manage and manipulate forms, space, and surfaces
- 3. Demonstrate iterative, code-based workflows across multiple software platforms, including: 3-D modeling, parametrics, and analysis
- 4. Prepare, export, and compose documentation into a useful format for design fabrication and presentation
- 5. Share code and results with online community

Lecture Content

Coding Fundamentals (The essentials of code-based design will be presented to facilitate a beginner-level of understanding using languages and workflows associated with the subject.) Variables Databases Syntax Control Flow Object Oriented Programming Grasshopper (Node-based coding will be presented to allow for an easy entry-point into code-based design. Comparisons will be made to text-based design coding.) Interface Comparisons Syntax Data Trees Vectors Plug-ins Python (Text-based coding will be presented to allow for entry into code sampling and customization.) Interface Comparisons Syntax Data Storage Vectors Code Sampling Presentation (Critical research projects that align with code-based design movements will be presented and discussed, including specific tecniques.) Historic Trajectories Design Projects Exporting Formatting Sharing

Lab Content

In support of lectures, lab content will include assignments and projects geared toward reinforcing topics covered. Coding Fundamentals (Assignments: warm-ups to gain confidence in code formatting and workflows) Logic Debugging Code Sampling Grasshopper (Patterning Project: an iterative approach to pattern making) Concept Development Command Plan Proof of Concept Iteration Python (Generative Design Project: a code-sampling exercise in working with multiple design variables to generate unexpected forms) Concept Development Pseudocode Proof of Concept Iteration Presentation (Portfolio: project-based research using coding techniques aligned with a historic design trajectory) Form Exploration Hybrid Drawings Renderings Code Documentation Composition Sharing

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)
- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Instructional Techniques

Lecture and in-class computational modeling assignments, quizzes, individual, and small group activities and instruction

Reading Assignments

Students will spend 2 - 3 hours a week reading assigned chapters from the text book, instructor handouts, and articles from online sources. Students will be expected to follow along with assigned exercises in the reading material and online assignments. Reading of Grasshopper and Python manuals to support project specifics, research on historic code-based design trajectories, and review of blog posts from designers with written content about their projects, e.g. Marc Fornes, Nervous Systems, Cmmnwlth.

Writing Assignments

Students will spend 1 - 2 hours a week completing written assignments, code writing, and examinations. Writing for this course includes minor notations and short professional descriptors. Critical thinking is reinforced in the act of developing a graphic, 3D concept and presenting it. A written survey and comparison of historic trajectories in code-based design will be required to help develop individual student interests.

Out-of-class Assignments

Students will spend 3 - 4 hours per week completing weekly 3D modeling and programming assignments or projects. Development of a visual and 3D-based personal projects to align individual student interests with project-appropriate design techniques covered in class and/or online. Review of Grasshopper and Python code shared openly on the Internet. Development of a visual and code-based narrative to align individual student interests with a historical trajectory in design, e.g. Greg Lynn's "Animate Form," Frank Gehry's "Digital Project," or Marcos Novak's "Transarchitecture.

Demonstration of Critical Thinking

Critical thinking is reinforced in the act of developing a design, coding and debugging 3-D solutions, and in the documentation and presentation of qualified results in each project.

Required Writing, Problem Solving, Skills Demonstration

Writing for this course only includes minor notations and short professional descriptors, as evidenced in the design and computational methods employed and presented.

Eligible Disciplines

Architecture: Any bachelor's degree and two years of professional experience, or any associate degree and six years of professional experience.

Textbooks Resources

1. Required Parsons, R. and Akos, G.. The Grasshopper Primer, 3.3 ed. ModeLab, 2014 Rationale: Primary source of information for Grasshopper 3D programming techniques

Manuals Resources

1. Fugier, M., Golay, P., Hambly, J., and Steeg, V.. Rhinoceros v6.0, Level 1 Traing Manual, Robert McNeel Associates , 02-21-2019 2. Tibbits, S., van der Harten, A., and Baer, S.. Rhino.Python Primer 101, Robert McNeel Associates , 01-01-2011

Other Resources

1. Instructor handouts and reference materials as needed for project 2. Digital Fabrication Equipment: CNC, laser cutters/engravers 3. Cleaning Supplies Equipment: Vacuum, broom, dustpan, trash cans