INTERACTIVE DEVELOPMENT 102

BA DEGREE IN INTERACTIVE ARTS

STUDY GUIDE

Subject Code: ID_102
Status of subject: Core (Leading to MAJOR)
Credits: 15
Contact hours: 64 (4 modules *8 weeks* 2 hours per week)
Pre-requisites: Basic computer literacy, Basic math literacy, Good command of the English language
Studio: B3

Subject manager: Morne Venter
Email: morne@openwindow.co.za

Lecturer: Rouan van der Ende
Email: rouan@openwindow.co.za

Course overview

The Interactive Development 101 course is a series of lectures introducing students to programming and interactive electronics in academic and practical assignments. The focus is on building the student's ability to solve problems logically and in a detailed enough way for machines to execute.

We start off with how to draw simple shapes to screen such as lines, circles and triangles. Then we progressively increase complexity developing a deep understanding of looped algorithms and data manipulation. The student will learn how to program complex ideas and generate beautiful patterns at the same time.

We progress from the first module into interactivity keyboard, mouse and touch. We'll build a simple game or interactive application. We'll learn about storing and retrieving data from storage.

We build simple electronic circuits and program microcontrollers to control robots that can sense and react to the environment.

We learn how to design and build interfaces to ease interaction and control of electronics and robotics.

Course Outcomes

By the end of the course students should be able to:

On completion of the module students should be able to

- Demonstrate an understanding of programming conventions and design patterns in code.
- Understand fractal and procedural algorithms.
- Understand how programming fits into society and utilizes the elements of complex systems.
- Show a working knowledge of javascript, html, css.
- Demonstrate handling input data to drive actions inside a program.

- Show a working knowledge of javascript events in the DOM and be able to use mongodb.
- Theorise about human machine interface design and future ways of interaction.
- Demonstrate understanding of microcontrollers and building simple circuits for interaction.
- Have ideas for implementing interaction design to practical applications in various industries.
- Combine interactive design for screen with communication to electronic circuits.
- Plan and implement a device with interface combination.

Components of course

Theoretical components:

- Study the history of computing as well as look at various futurists work.
- How the world can be improved by technology.
- This can take the form of an illustration or written piece.
- The learners study human machine interaction design by looking at various machines or devices we use now and problems or inefficiencies currently faced.
- Explain in a written piece or illustrated artwork what could be improved or made to improve the interaction between humans and computers.
- Setting up drivers to program microcontroller
- Software to program microcontroller.
- Components and breadboards
- Example code and circuits
- Making the circuit do what you want
- Publishing and documenting project online
- Github version control
- Markdown syntax

Practical components:

- Learners create a programmatic looping digital artwork for web distribution. Heavy use of fractals and l-system like algorithms should be used and direct image use avoided.
- They should show an understanding of random number generation, utilizing 3D distance and angles, nearest neighbours, and visualising hierarchical data structures.
- Learners plan, design and implement an interactive web application or game that utilizes keyboard, mouse, touch or other inputs. The system must make use of a method to retain information through a power cut.
- Electricity and the growth of civilization, looking at the future.
- Learners will build a device that works together with a remote interface of their own design. This needs to be presented and demonstrated in week 8. The build log will form the theoretical part of this module and has to be online.
- JSON communication between circuits and computers
- Device API design
- HTML socket.io interface binding

Modular overview

- Module 1: Introduction to Programming*
- Module 2: Interactive Programming
- Module 3: Introduction to Electronics
- Module 4: Networked Interface Design for Interactive Robotics

Assessed components and mark structure

Assessed components
20% Theoretical
80% Practical

Modular marks structure

25% Module 1
25% Module 2
25% Module 3
25% Module 4

Relevance to industry

As mobile devices increase we are become a more digital society. Part of this progression is our understanding and command of all things digital. This means being able to not only craft digital data, but also use digital as a means to affect and measure the physical world. By learning how to combine electronics, art and code we unlock new means of solving problems.

Relation to other subjects

We have a strong focus on the technical aspects in this course that will have to be supported by more design or philosophical based classes to be of maximal use. As with all things, the more you know about everything, the better you understand how something fits into the whole.

Relation to previous and following years

The previous year we introduced much of this content as two electives in the Film Arts department as an augmentation of the gamedevelopment, allowing students to build their own game controllers and games.

Module 1: Introduction to Programming

Overview

We start off with how to draw simple shapes to screen such as lines, circles and triangles. Then we progressively increase complexity developing a deep understanding of looped algorithms and data manipulation. The student will learn how to program complex ideas and generate beautiful patterns at the same time.

Outcomes

On completion of the module students should be able to demonstrate an understanding of programming conventions and design patterns in code. The learner should also understand fractal and procedural algorithms. The learner should understand how programming fits into society and utilizes the elements of complex systems. Show a working knowledge of javascript, html, css.

Assessment Activities

Theoretical assignments - 20%

The learners study the history of computing as well as look at various futurists work. They then need to convey their point of view on how the world can be improved by technology. This can take the form of an illustration or written piece.

Practical assignments - 80%

Learners create a programmatic looping digital artwork for web distribution. Heavy use of fractals and l-system like algorithms should be used and direct image use avoided. They should show an understanding of random number generation, utilizing 3D distance and angles, nearest neighbours, and visualising hierarchical data structures.

Module 1 Content

The following content will be covered during this module:

- Elements of Programming
    - Variable
    - Array
    - Object
    - Function
    - Iteration
    - Events
- Elements of Graphical Programming
    - Bit, Pixel, Line, Ellipse
    - Colour
    - Frame rate
    - Location
    - Relation
    - Events
- Role of programming
    - Calculation
    - Organisation
    - Visualisation
    - Simulation
    - Automation
    - Transformation

Module 1 Schedule

Week 1

- Module brief.
- Introduction to programming and computer theory.
- HTML

Week 2

- CSS
- id, class

Week 3

- JAVASCRIPT
- Conditionals, if else.
- Variables, Integers, Floating point, Strings
- Arrays, pop, splice.
- For, while.
- Line, Rect, Triangle, Ellipse.
- Colour, Fill, Stroke.

Week 4

- Objects.
- JavaScript fundamentals
- Distance and Random.
- Emergent behaviour.

Week 5

- Formats and keeping time.
- Scale and relativity
- Conway game of life
- Popular coding patterns

Week 6

- Revision session on programmed artworks.

Week 7

- Final class critique on programmed artworks.

Week 8

- Learner presentations.

Module 1 Learning resources

Douglas Crockford - Crockford on JavaScript - Volume 1: The Early Years (https://www.youtube.com/watch?v=JxAXlJEmNMg)
http://www.w3schools.com/html/ (http://www.w3schools.com/html/)
http://www.w3schools.com/css/ (http://www.w3schools.com/css/)
http://www.w3schools.com/js/ (http://www.w3schools.com/js/)

Module 2: Introduction to Interactive Programming

Overview

We progress from the first module into interactivity with keyboard, mouse and touch inputs. We'll build a simple game or interactive application. We'll also learn about storing and retrieving data from storage.

Outcomes

On completion of the module students should be able to:

- Demonstrate handling input data to drive actions inside a program.
- Show a working knowledge of javascript events in the DOM and be able to use mongodb.
- Theorise about human machine interface design and future ways of interaction.

Assessment activities

Theoretical assignments - 20%

The learners study human machine interaction design by looking at various machines or devices we use now and problems or inefficiencies currently faced. Explain in a written piece or illustrated artwork what could be improved or made to improve the interaction between humans and computers.

Practical assignments - 80%

Learners plan, design and implement an interactive web application or game that utilizes keyboard, mouse, touch or other inputs. The system must make use of a method to retain information through a power cut.

Module 2 Content

The following content will be covered during this module:

- The Elements of Interactive program implementation I
    - Device display and inputs
    - Connectivity
    - User identity
    - Accessibility
- The Human Machine interface
    - Intuitive UX design
    - Robot interfaces
    - Data accessibility
    - Data input

Module 2 Schedule

Week 1

- Module brief.
- Introduction to human machine interfaces

Week 2

- Mouse, Keyboard.
- Events

Week 3

- Input boxes and click events.
- JQUERY

Week 4

- socket.io
- Xbox Kinect. Leapmotion.
- Facial recognition. Processing.

Week 5

- IMU (inertial measurement – gyro/accel/magnetometer).
- Force feedback. Joysticks/steeringwheels. Arcade.
- Oculus rift. VR

Week 6

- Revision session on programmed interactive games/programmes/artworks.

Week 7

- Final class critique on programmed interactive games/programmes/artworks.

Week 8

- Learner oral theory presentations.
- Learner practical evaluations.

Module 2 Learning Resources

Kevin Kelly - The Technium (https://www.youtube.com/watch?v=j0fkrJgmQxs)
http://jquery.com/ (http://jquery.com/)

http://threejs.org/ (http://threejs.org/)
http://www.mongodb.org/ (http://www.mongodb.org/)
http://socket.io/ (http://socket.io/)
http://nodejs.org/ (http://nodejs.org/)
http://expressjs.com/ (http://expressjs.com/)

Module 3: Interactive Matter

Overview

We build simple electronic circuits and program microcontrollers to control robots that can sense and react to the environment. We look at how robotics and automation affect the means of production in society.

Outcomes

On completion of the module students should be able to:

- Demonstrate understanding of microcontrollers and building simple circuits for interaction.
- Have ideas for implementing interaction design to practical applications in various industries.

Assessment activities

Theoretical assignments**

The learners study human machine interaction design by looking at various machines or devices we use now and problems or inefficiencies currently faced. Explain in a written piece or illustrated artwork what could be improved or made to improve the interaction between mankind and devices/machines/robots in large-scale industry such as farming, automotive, aerospace, military etc.

Practical assignments**

Learners will get an arduino UNO microcontroller, usb-cable, breadboard, LEDs, LDRs (and hopefully some more components. http://arduino.cc/en/Main/ArduinoStarterKit (http://arduino.cc/en/Main/ArduinoStarterKit)) We will learn how to blink LEDs, sense light and program custom interactions. They have to build an arduino device.

Module 3 Content

The following content will be covered during this module:

Practical

- Setting up drivers
- Software
- Components and breadboards
- Example code and circuits
- Making the circuit do what you want

Theory

- Electricity recap
- Resistance
- Capacitance
- Inductance
- Electricity and the growth of civilization

Module 3 Schedule

Week 1

- Module brief.
- Introduction to arduino and electricity
- Blinking LEDs
- Breadboards
- Drivers and software

Week 2

- Light sensors
- Conditionals. if/else

Week 3

- buttons, potentiometers

Week 4

- Stepper motors, temperature sensors.

Week 5

- IMU (gyro/accel/magnetometer)

Week 6

- Revision session on programmed interactive games/programmes/artworks.

Week 7

- Final class critique on programmed interactive games/programmes/artworks.

Week 8

- Learner oral theory presentations.
- Learner practical evaluations.

Module 3 Learning resources

http://arduino.cc/ (http://arduino.cc/)
https://www.robotics.org.za/ (https://www.robotics.org.za/)

Module 4: Remote Interfaces for Interactive Matter

Overview

Continuation of previous modules hardware. We learn how to design and build interfaces to ease interaction and control of electronics and robotics.

Outcomes

On completion of the module students should be able to:

- Combine interactive design for screen with communication to electronic circuits.
- Plan and implement a device with interface combination.

Assessment activities

Theoretical assignments

The theory component for this module is focussed on the practical device and interface software the learner needs to hand in. This will be documenting the planning, design, prototyping, implementation up to completion. The documenting of the project needs to be published online. Marking will happen based on a URL submitted.

Practical assignments

Learners will build a device that works together with a remote interface of their own design. This needs to be presented and demonstrated in week 8. The build log will form the theoretical part of this module and has to be online.

Module 4 Content

The following content will be covered during this module:

- Publishing and documenting online
- Github
- Markdown syntax
- JSON communication between circuits and computers
- Device API design
- HTML socket.io interface binding

Module 4 Schedule

Week 1

- Module brief
- Introduction to module

Week 2

- aJSON
- APIs for devices
- Github and markdown

Week 3

- jquery, socket.io and nodejs

Week 4

- Interface design study
- Implementation of sensor feed graph

Week 5

- IMU (inertial measurement – gyro/accel/magnetometer).
- Force feedback. Joysticks/steeringwheels. Arcade.
- Oculus rift. VR

Week 6

- Support on learner projects.

Week 7

- Class critique on learner projects.

Week 8

- Learner Final Presentations.

Module 4 Learning Resources

https://github.com/interactive-matter/aJson (https://github.com/interactive-matter/aJson)
http://socket.io/ (http://socket.io/)
http://nodejs.org/ (http://nodejs.org/)