

**BA DEGREE IN INTERACTIVE ARTS**

**STUDY GUIDE DRAFT**

**INTERACTIVE DEVELOPMENT 101**

**The Open Window School of Visual Communication**

**School of Design Studies**

Lecturer: Rouan van der Ende

Email: rouan@openwindow.co.za

Lecture Studio: UNKNOWN

**INTERACTIVE DEVELOPMENT 101**

|  |  |
| --- | --- |
| Subject code |  |
| Status of subject | CORE (leading to MAJOR) |
| Credits | 15 |
| Contact hours | 64 |
| Pre-requisites | * Basic computer literacy * Basic math literacy * Good command of the English language |
| Studio | B3 |
| Subject manager | Rouan van der Ende |
| Lecturer | Rouan van der Ende |

**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.

**Modular overview**

***Module 1: Introduction to Programming***

We start off teaching the student 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.

***Module 2: Interactive Programming***

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.

***Module 3: Introduction to Electronics***

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

***Module 4: Networked Interface Design for Interactive Robotics***

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

**Assessed components and mark structure**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Assessed components*** |  | ***Modular marks structure*** |  |  |  |  |  |
| Theoretical | Practical | Module 1 | Module 2 | Mid-Year Evaluation | Module 3 | Module 4 | End of year evaluation |
| **20%** | **80%** | **25%** | **25%** | **-** | **25%** | **25%** | **-** |

**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***

**1.1 Module Overview**

We start off teaching the learner 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.

**1.2 Module outcomes**

*On completion of the module students should be able to:*

* Demonstrate an understanding of programming conventions and design patterns in code. An understanding of fractal and procedural algorithms.
* Understand how programming fits into society and the elements of complex systems.
* Show a working knowledge of javascript, html, css.

**1.3 Assessment activities**

***1.3.1 Theoretical assignments***

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.

***1.3.2 Practical assignments***

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.

**1.4 Module content**

The following content will be covered during this module:

|  |  |
| --- | --- |
| **Component** | **Sub-components** |
| The Elements of Programming | * Elements of Programming   + Variable   + Array   + Object   + Function   + Iteration   + Events * Elements of Graphical Programming   + Pixel, Line, Ellipse etc.   + Colour   + Frame rate   + Location   + Relation   + Events |
| The role of Programming in the world | * Role of programming   + Calculation   + Organisation   + Visualisation   + Simulation   + Automation   + Transformation |

**1.5 Module programme**

|  |  |
| --- | --- |
| **Week** | **Themes / Topics** |
| 1 | * Module brief. * Introduction to programming and computer theory. * HTML |
| 2 | * CSS |
| 3 | * JAVASCRIPT * Conditionals, if else. * Variables, Integers, Floating point, Strings * Arrays, pop, splice. * Iteration. For, while. * Line, Rect, Triangle, Ellipse. * Colour, Fill, Stroke. |
| 4 | * Functions. Objects. * JavaScript fundamentals * Pythagoras. Distance and Random. * Emergent behaviour. |
| 5 | * Time. Formats and keeping time. * Scale and relativity * Conway game of life * Popular coding patterns |
| 6 | * Revision session on programmed artworks. |
| 7 | * Final class critique on programmed artworks. |
| 8 | * Learner oral theory presentations. * Learner practical evaluations. |

**1.7 Learning resources**

***1.7.1 Prescribed material***

# Douglas Crockford. Crockford on JavaScript - Volume 1: The Early Years.

# <https://www.youtube.com/watch?v=JxAXlJEmNMg>

**Kevin Kelly.** The Technium. <https://www.youtube.com/watch?v=j0fkrJgmQxs>

***Module 2: Introduction to Interactive Programming***

**2.1 Module 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.

**2.2 Module 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.

**2.3 Assessment activities**

***2.3.1 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 humans and computers.

***2.3.2 Practical assignments***

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.

**2.4 Module content**

The following content will be covered during this module:

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

**2.5 Module programme**

|  |  |
| --- | --- |
| **Week** | **Themes / Topics** |
| 1 | * Module brief. * Introduction to human machine interfaces |
| 2 | * Mouse, Keyboard. * Events |
| 3 | * Input boxes and click events. JQUERY |
| 4 | * Socket.io * Gesture. Xbox Kinect. Leapmotion. * Facial recognition. Processing. |
| 5 | * IMU (inertial measurement – gyro/accel/magnetometer). * Force feedback. Joysticks/steeringwheels. Arcade. * Immersion. Oculus rift. VR |
| 6 | * Revision session on programmed interactive games/programmes/artworks. |
| 7 | * Final class critique on programmed interactive games/programmes/artworks. |
| 8 | * Learner oral theory presentations. * Learner practical evaluations. |

**2.6 Learning resources**

***2.6.1 Prescribed material***

# TBD

***Module 3: Introduction to Programmable Electronics***

**3.1 Module 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.

**3.2 Module 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.

**3.3 Assessment activities**

***3.3.1 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.

***3.3.2 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> ) We will learn how to blink LEDs, sense light and program custom interactions. They have to build an arduino device.

**3.4 Module content**

The following content will be covered during this module:

|  |  |
| --- | --- |
| **Component** | **Sub-components** |
| Practical | * Setting up drivers * Software * Components and breadboards * Example code and circuits * Making the circuit do what you want |
| Theory | * Voltage \* Current = Watts * Resistance * Capacitance * Inductance * Electricity and the growth of civilization |

**3.5 Module programme**

|  |  |
| --- | --- |
| **Week** | **Themes / Topics** |
| 1 | * Module brief. * Introduction to arduino and electricity |
| 2 | * Blinking LEDs * Breadboards * Drivers and software |
| 3 | * Light sensors, buttons, potentiometers |
| 4 | * Stepper motors, temperature sensors. |
| 5 | * IMU (gyro/accel/magnetometer) |
| 6 | * Revision session on programmed interactive games/programmes/artworks. |
| 7 | * Final class critique on programmed interactive games/programmes/artworks. |
| 8 | * Learner oral theory presentations. * Learner practical evaluations. |

**3.6 Learning resources**

***3.6.1 Prescribed material***

# TBD

***Module 4: Networked Interface Design for Interactive Robotics***

**4.1 Module Overview**

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

**4.2 Module 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.

**4.3 Assessment activities**

***4.3.1 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.

***4.3.2 Practical assignments***

Learners will build a device that works together with a remote interface. 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.

**4.4 Module content**

The following content will be covered during this module:

|  |  |
| --- | --- |
| **Component** | **Sub-components** |
| Practical | * JSON communication between circuits and computers * Device API design * HTML socket.io interface binding |
| Theory | * JSON * Documenting a build * Posting projects online |

**4.5 Module programme**

|  |  |
| --- | --- |
| **Week** | **Themes / Topics** |
| 1 | * Module brief. * Introduction to arduino and electricity |
| 2 | * Blinking LEDs * Breadboards * Drivers and software |
| 3 | * Light, Ultrasonic, Sound, * Arduino demonstration (next module). |
| 4 | * Gesture. Xbox Kinect. Leapmotion. * Facial recognition. Processing. |
| 5 | * IMU (inertial measurement – gyro/accel/magnetometer). * Force feedback. Joysticks/steeringwheels. Arcade. * Immersion. Oculus rift. VR |
| 6 | * Revision session on programmed interactive games/programmes/artworks. |
| 7 | * Final class critique on programmed interactive games/programmes/artworks. |
| 8 | * Learner oral theory presentations. * Learner practical evaluations. |

**4.6 Learning resources**

***4.6.1 Prescribed material***

# TBD