IT6034 Course Design

IT IT6034 Game Development

# Reason for New Course / Change to Course

The reason for the new course is NZQA’s reviewed ICT qualifications, in particular:

[2604 New Zealand Diploma in Software Development (Level 6)](http://www.nzqa.govt.nz/nzqf/search/viewQualification.do?selectedItemKey=2604)

# Course Requirements

## NZQA Requirements

This course is designed to meet (parts of) the following Graduate Profile Outcomes:

GPO1. Analyse requirements, design and document software solutions for a range of problems in an organisational context. (LO4, 5)

* User Expertience(UX) design including user interface, HCI principles, and universal accessability
* Creating accurate and clear technical and user documentation
* Application of the core interaction design concepts and practice
* Application of the core information systems skills and knowledge

GPO2. Write and maintain programs using design patterns, data structures and algorithms to meet specifications. (LO1, 2, 3)

* Coding – object oriented, procedural
* Facility in multiple common programming languages and integrated development environments (IDEs), which fosters the ability to migrate to new languages, tools and systems;
* Construct software with complex, multi-element architectures and abstract data types.
* Writing code following design patterns and software development standards;
* Source and version control
* Optimisation concepts and techniques
* Program maintenance techniques
* Application of the core software development concepts and practice

GPO3. Apply a range of software quality assurance techniques to verify correctness of systems. (LO5)

* Design a variety of tests including unit and system tests, usability testing, user acceptance tests; incorporating a range of testing techniques e.g. white box, black box, boundary-value testing;
* Testing on a range of platforms
* Debugging, which includes debugging utilities, managing bug reports and issue tracking

GPO4. Apply data management and storage technologies to support the software application and the development process. (LO4)

* Designing and implementing appropriate application data access, management and storage technologies to match the application domain
* Digital asset management and storage technologies appropriate to match the application domain

GPO5. Establish application security by integrating security principles throughout software development to ensure system integrity. (LO4, 5)

* Protecting data integrity, data validation techniques, data access permissions;
* Authentication and authorisation

GPO6. Choose, justify and apply architecture, technologies, and tools, to implement the software solution (LO6)

* Tool selection and architecture

## CPP Course Descriptor

*Copy the aim, LO and Assessments from course descriptor*

(Version of 31 August 2017, submitted to programme committee)

### Aim

To give students knowledge and skills of using another programming language and framework to enable them to build a game application, including game design, scripting, development and testing. – NEW PROPOSED AIM

To give students an understanding of the software design patterns, and knowledge and skills in another programming language and/or framework to enable them to develop a game application – OLD AIM

### Learning outcomes

LO1. Write code using object oriented and functional approaches

LO2. Design and write programs considering appropriate design patterns and following applicable software development standards

LO3. Analyse and apply software optimisation concepts and techniques

LO4. Investigate, design and implement application data access, management, and storage technologies appropriate to game development

LO5. Research and implement core software development practices as they apply to game development

ADD an OUTCOME

### Assessment schedule

|  |  |  |
| --- | --- | --- |
| **Assessment Method** | **Weighting** | **Learning Outcome/s** |
| Practical Tasks | 20% | LO 1-3 |
| Project Design | 25% | LO 2,5 |
| Project Execution | 40% | LO 1-7 |
| Project Presentation | 15% | LO 3, 5 |

# Assessment

*High-level outline of each Assessment item. Focus on how the assessment items will enable the students to demonstrate that they have met the learning outcomes.*

## Summative Assessment: Practical Tasks (20%)

Suggested Practical Tasks:

* Small application using OO principles
* Using Singleton
* Using Factory
* Using … Pattern (Coconut example)
* Optimise the given code
* WORKSHOP: VR

## Summative Assessment: Project Design (25%)

Create a game design document based on one of the game proposals.

Possible tasks:

* Task 1 – Requirements Analysis & Planning
* Task 2 – Solution Design, incl. Security Design and UI Design
* User Cases

## Summative Assessment: Project Execution (40%)

*To assess theory points of each module and anything not covered by the project.*

*Code and build game solution based on the approved project design.*

*Provide proof of testing conducted.*

*Create appropriate documentation.*

*[Alternative approach: to assess the theoretical part would be to include a critical discussion of project process and outcomes within the project.]* -> See last Task above

* Task 3 – Code Game based on the game design
* Implement different types of testing – regression testing

## Summative Assessment: Project Presentation (15%)

*To assess theory points of each module and anything not covered by the project.*

*[Alternative approach: to assess the theoretical part would be to include a critical discussion of project process and outcomes within the project.]* -> See last Task above

* VR or Augmented reality research ?
* Present the game, talk about design decisions and critical discussion on the optimisation challenges, design patterns and end product, UX challenges
* User testing during presentation
* What challenges faced, what would have done differently if started from scratch, what design patterns used, what initial idea, theme of the game.

## Formative Assessment (0%)

*1-2 formative tasks prepare the student for the summative assessment. Scaffolding.*

Some of the Practical tasks listed above would be formative rather than summative, especially if they are going to be part of the project. We don’t want to overassess.

Go through game design phases and check each stage.

VR Workshop

# Course Structure

*Initial outline of course modules and activities. Learning activities and course content to support and guide student towards skills and knowledge needed for the summative assessment.*

*FROM UNITY CERTIFICATION*

*Introduction to Game Design, Prototyping and Development book*

1. The Basics of Game Design (UX, documentation, process, see links in the design document)
2. Unity Environment (Installing, setting up, Unity interface, creating new project, inspector tab, transform component, collider component, scene tab, game tab, project tab, layout customization)
3. Game Development in Unity (Game Objects, Prefabs, Models, Rigidbodies, Colliders, Materials)
4. Scripting with C# (Visual Studio environment, variables, conditional statements, loops, functions, arrays and lists, OO class, inheritance, interface)
5. Scripting in Unity – Giving objects some (physics, scripting)
6. Design Patterns (Singleton, Factory, Object …)
7. Optimisation techniques
8. Game Testing (Test driven development(TDD), Unity Test tools, Bug reporting cycle, Red>Green> Refactor, Try catch for Error Handling)
9. Game deployment (PC, mobile)
10. Write code using both object oriented and functional approaches

Objectives:

* Creating C# scripts
* Using variables and components
* Working with Boolean operations and conditionals
* Understand and use C# loop structures
* Functions and parameters
* Classes
* The object-oriented thinking

1. Design and write programs following design patterns and software development standards

Objectives:

* Design patterns overview: Singleton. Factory, Object …
* Recognise and apply Singleton pattern to …
* Using Factory design pattern to …
* Using Object … pattern
* Optimise code to use Design Patterns
* Design pattern in gaming

1. Explain and apply optimisation concepts and techniques

Objectives:

* Explain raycast parameters
* Describe the function of raycast
* Assess raycast trajectories
* Utilise colliders

1. Design and implement application data access, management, and storage technologies appropriate to game development

Objectives:

1. Explain and implement core software development practices as they apply to game development

Objectives:

* Implementing and utilising various methods for game playtesting
* Game Objects
* Identify and prepare models, materials and textures
* Terrain
* Light and Cameras
* Creating scripts
* Collision (rigidbodies, collision, triggers, raycasting)
* Working with prefabs
* Basic UI principles
* Use state machine behaviors and animation curves to drive game logic.
* Create your own missions that chain together different series of game events, unlock additional items and uncover new areas of the world.
* Build an in-game inventory system for players to manage their weapons, ammo, health boosters, mission items, and more.
* Learn how to use scripted events to drive your story line forward.
* Understand fundamental concepts in the field of game AI, like navigation and path-finding, state machines, animation state machines, and more.
* Create custom interfaces for health bars, ammo, inventory items, and more.

Useful reference for defining learning objectives:

[*https://www.udemy.com/build-your-own-first-person-shooter-survival-game-in-unity/*](https://www.udemy.com/build-your-own-first-person-shooter-survival-game-in-unity/)

<http://www.mediadesignschool.com/media/documents/unity-certified-developer_courseware_learning-outcomes.pdf>

<https://www.coursera.org/specializations/game-development>

## AUT: Introduction to Game and Play Design (15.00)

Introduces the foundations of game and play design. Students investigate "Why We Play" (Ludology) and "The Rules of Play". This is followed by Game Design, Playability, Types of Games and Play as a Design Process. Explorations and conceptualisations of play and game design will be developed through practical experimentation, critical reflection and theory contextualisation.

GAME DESIGN DOCUMENT:

<https://www.gamasutra.com/view/feature/131791/the_anatomy_of_a_design_document_.php?page=2> – Game Design document

<https://www.gamasutra.com/view/feature/131632/creating_a_great_design_document.php>

<https://en.wikipedia.org/wiki/Game_design_document>

## Learning Activities

*Outline which learning activities will help students to develop skills, knowledge and attitudes required for the formative and summative assessments.*

Scaffold learning towards the practical tasks listed

* Install and use server-side framework
* Write server-side scripts to connect to database
* Write sever-side scripts to implement security features, one by one
* Study/modify code-snippets that implement the above
* Test websites for security flaws – using online learning resources
* Correct security flaws in websites
* Discuss security threats and mitigation in a group
* Investigate security threats and report to a group/ contribute to course forum

<https://www.codehunt.com/>

## Content

Suggested content (Luisa, 29/8/2017 and 08/09/2017, and Benedicte 08/09/2017), order and number of modules to be refined

* Server-side scripting with django, incl. connecting to database – follow tutorial (see refs) LO2, LO5
* Security principles, Security design and design patterns LO1, LO6
* Key security features of applications, eg authentication, authorisation, data validation, data integrity, etc – Practical tasks to implement each, e.g. logon, form processing, using admin console for permissions, … LO2, LO3, LO4
* Main security threats – OWASP to ten – and their mitigation – Practical tasks to implement them and then mprevent them, code warrior LO1
* Penetration Testing – testing apps and websites for security faults
* Project: implementing a complete website, with security features and protections LO1-LO6

<https://www.khanacademy.org/partner-content/pixar>

<https://www.coursera.org/learn/gamification/lecture/4h5k1/1-1-introduction>

Self-assessment: a Game with 3 doors where each door has a random question. You need to go through 5 doors each time to get a chest (or something). Every time you answer the question wrong you get -50, every time you answer right you get +100. Every time you answer wrong the question goes back in the pool of questions.

Use unfinished games to showcase specific topics. At the end give the source code to the students.

<https://taiga.io/> - project management Agile free tool

This course includes the following modules:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Module** | **Weight**  **(credits)** | **Content hours\*** | **LO** | **Content, Notes** | **Who** | **Status**  **(dd/mm/yy)** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

*\* Based on standard course workload table, total 60h content study = 4h content per credit point.*

## Learning Objectives of Modules

*Initial outline of learniong pbjectives for each module. Can be removed/ignored once they have been entered in iQualify.*

### Module: Server-side scripting

* ….

### Module: Security principles

* …

# Design Decisions

*Note key decisions that have been made, e.g. on technology or content, with rationale, especially for contentious points…*

Technology:

Database: SQLite

Server-side Framework: django framework (alternative: .net)

Server-side scripting language: Python

Justification

SQLite very popular

Use open source technologies where possible.

Students have already learnt Python in IT5014 Programming Principles

# IT Resources

*This section outlines the IT environment and software needed for students to take the course. These requirtements need to be sent to IT early on, so that these resources can be made available.*

## IT Environment

*what students need for development tasks, e.g. web server, sql server, mobile devices, etc.*

* Students wil need the ability to download and install software on their workstation.
* They will also need the ability to deploy their website on a server that is accessible to CPP staff and students via an url
* Ideally a range of mobile devices to perform website tests (tablets and phones, iOS, Android and Windows)

## Software resources

*List any software needed, and whether we need to purchase licenses.*

* SQLite
* Django framework
* Python
* Collaboration software
* …

## Learning Apps

* Code Warriors: <http://www.codewarriorsgame.com/kai-web-kai/login>

Does this need a license or account? What tech does it use?

* Capture the Flag competitions <https://ctf365.com/>

<https://www.youtube.com/watch?v=8ev9ZX9J45A>

<https://blogs.cisco.com/perspectives/cyber-security-capture-the-flag-ctf-what-is-it>

<http://captf.com/practice-ctf/>

# References & Teaching Resources

## Video Tutorials

*List lynda.com tutorials used, and any others.*

Unity 5: 3D Essential Training by Adam Crespi – from Lynda.com

Unity tutorials

There’s also a whole range of Lynda.com videos on ethical hacking.

Django videos

## Django resources

<https://www.lynda.com/Django-tutorials/Up-Running-Python-Django/386287-2.html>

<https://www.tutorialspoint.com/django/>

<https://www.lynda.com/Django-tutorials/Using-Django-authentication-framework/594454/600923-4.html>

ORM video: <https://www.lynda.com/Django-tutorials/Querying-data-Django-ORM/386287/435077-4.html>

and resources for later courses:

APIs [https://www.lynda.com/Django-tutorials/Using-Tastypie-create-REST-API/521202/529207-4.html](https://www.lynda.com/Django-tutorials/Using-Tastypie-create-REST-API/521202/529207-4.html?srchtrk=index%3a5%0alinktypeid%3a2%0aq%3adjango%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2)

Maintenance

<https://www.lynda.com/Django-tutorials/Model-managers/521202/529219-4.html>

## Textbooks

*List books from ProQuest Safari*

Dauzon, S., Bendoraitis, A., & Ravindran, A. (2016). *Django: Web Development with Python*. Birmingham, UK: Packt Publishing <http://proquestcombo.safaribooksonline.com/book/web-development/django/9781787121386> (this was used in the L5 Testing course as intro to django framework)

Developer's Guide to Web Application Security, By: Michael Cross, Pub. Date: April 18, 2011 <http://proquestcombo.safaribooksonline.com/book/networking/security/9781597490610>

Application Security in the ISO 27001:2013 Environment, By: Vinod Vasudevan, Publisher: IT Governance Ltd, Pub. Date: October 15, 2015 <http://proquestcombo.safaribooksonline.com/book/networking/security/9781849287692/chapter-1-introduction-to-the-international-information-security-standards-iso27001/chapter_01_html>

Security in Computing, Fifth Edition, By: Charles P. Pfleeger; Shari Lawrence Pfleeger; Jonathan Margulies, Publisher: Prentice Hall, Pub. Date: January 26, 2015 <http://proquestcombo.safaribooksonline.com/book/networking/security/9780134085074>

## Web Articles & Other References

*Paste links any resources found during your initial research. The ones you will use in the course will end up in the references page of the course.*

Shore, X. (2016). Visiting OWASP [Video file] Retrieved from <https://www.lynda.com/Linux-tutorials/Visiting-OWASP/512727/571178-4.html>

(The new OWASP Top 10 2017 is to be released in late November 2017. Link may be posted here: <https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project> )

OWASP 2013 <https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#OWASP_Top_10_for_2013>

Design Principles for Secure applications <https://multimedia.telos.com/blog/11-design-principles-for-secure-applications/>

ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements. <https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en>

What is the Difference Between Black, White and Grey Hat Hackers? <https://community.norton.com/en/blogs/norton-protection-blog/what-difference-between-black-white-and-grey-hat-hackers>

## Resources from industry events

*Introduction to Web Security*, by Ian Ross, ANZTB Chair, March 2017. This was an **excellent introduction to the main concepts of application security**, easy to understand.

http://www.anztb.org/userfiles/files/Introduction%20to%20web%20security%20ANZTB-March2017.pdf

*Security Testing, by Mark Shaw*, Testing for Tomorrow conference, Wellington, May 2017. No slides available, unfortunately. But great presentation!!!! Some of the references and ideas have been included above. BF has made some notes from this presentation.

Not great slides but listed anyway…Laura Bell, *Secure by Design* <http://www.anztb.org/userfiles/files/Secure_by_Design.pdf>

# Working Group

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Contact** |
| Developers  (lead and co-developers) | Inna Medyannikova  Benedicte Florin | [Inna.Medyannikova@cpp.ac.nz](mailto:Inna.Medyannikova@cpp.ac.nz)  Benedicte.Florin@cpp.ac.nz |
| Industry Advisors | Joe Chang (winner of the …)  David Cummins (programmer for the indie game Phantasmal, Game Engineer/Programmer at Rush Digital ) |  |
| Tutor | Anchal …  John Wang |  |
| Tech Reviewer |  |  |
| Proofreader |  |  |
| Pre Assessment Moderator |  |  |
| Post Assessment Moderator |  |  |

## Industry consultation

*Note any form of consultation with industry or attendance of professional events that inform your course development. This will need to be incorporated into the overall consultation log.*

*Example from testing course.*

| **Date** | **Type** | **Who / Contact details** | **Outcome** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |