Course Components

Game Programming

Bachelor of Software Engineering

www.mediadesign.com

Year One

GD1P01: Introduction To Software Engineering for Games

Year one begins with an introduction to the C++ programming language. Once basic programming skills have been mastered students get the opportunity to construct simple games. They begin by solving easy problem-based tasks with C++ programming and progress on to learn how to construct, test, and debug simple computer games.

The lecturers will provide modern theoretical perspectives and demonstrate approaches to the tasks with examples. To demonstrate their understanding, students are presented with a brief to create a text-based game using the skills and knowledge that they have gained in this component.

GD1M01: Fundamental Mathematical and Engineering Principles

Mathematics is a fundamental building block of game development. This component begins with basic mathematics before progressing to the core mathematical skills required for solving games problems (and building on them throughout this component). When game problems require complex mathematical solutions the concepts are taught using gaming analogies and in game examples to explain the mathematical functions.

Teaching generally consists of theoretical elements, a demonstration, and then the lecturers allow students to put these skills into practice. The students collaborate and share mathematical problem-solving approaches during frequent in-class discussions and are expected to provide their solutions for class reviews.

GD1P02: Algorithms and Data Structures

This component teaches the fundamental data structures and algorithms that are needed to solve common gaming problems. Wherever possible, lecturers will show examples and use analogies to explain and make the principles easier to understand. Students improve their learning throughout this component by working on a large number of projects; they solve common gaming problems by designing, developing, implementing, testing, and enhancing a collection of data structures and algorithms.

GD1S01A: Introduction to Games Mechanics

Students are introduced to the interdisciplinary academic study of videogames through examining their cultural, educational, technical and social functions in contemporary settings. By playing, analysing, reading, discussing and writing about them, they will examine how games function from a technical perspective.

This component will help students gain an understanding of in-game mechanisms, economies and narratives by investigating and analysing existing games.

GD1M02: Mathematics for Graphical Games

Building on the foundations provided in previous components, students learn to construct mathematical solutions to common gaming problems. They design, develop, test, and enhance a game that requires a significant degree of mathematics. Trigonometry is used to solve problems for 2D games and, as students progress these 2D physics problems will become more complicated.

Software engineering models and notations are used to represent mathematical questions and students learn to write these for all mathematical code. This component then throws in an additional dimension introducing principals used in 3D games (vectors and matrices). Groups work together to solve the more challenging mathematical problems and in-class discussions are held and encouraged to assist students in their understanding of the concepts.

GD2S01: Software Engineering Principles and Practices

This component focuses on the skills required to produce a game both on time and on budget. Students are introduced to project management, design patterns and produce game management approaches for simple projects using a software engineering framework. Lecturers present a range of software project management methodologies and contemporary methods, as well as discussing effective/ineffective planning and effective/ineffective management examples (using case studies).

Students will learn to separate project management considerations from the wider context of game development. They are given game design and technical design documents to review in terms of project management, task allocations and stakeholders' roles and responsibilities.

GD1P03: 2D Game Programming

More advanced programming concepts are introduced including a basic introduction to user-interface design and software engineering management methods. Students will follow a predetermined plan and track their progress throughout this component with the experience that they gain assisting in the development of future projects.

Teaching approaches incorporate theoretical lectures and practical project-based learning. Lecturers provide game frameworks for students to read and understand which are then followed to solve progressively more complex problems. By the end of this component students will develop simple 2D games with effective user-interface design strategies.

GD1J01: Game Design Principles

Introducing ethics and social issues to gaming, students will learn to analyse games from a non-technical viewpoint. They learn about various target platforms that support games and to identify real-world game design problems.

Lecturers teach this component with in-class demonstrations and lectures on design and game design principles. The class collaborates to produce a game design document that justifies decisions made across a broad range of design and game design elements.

Year Two

GD2S02: Software Engineering for Games

Building on the project management principles from earlier components, students learn a theoretical modelling system for formal analysis of correctness and quality in the production of a game. They will also gain knowledge of software product assurance, experimenting with a variety of product assurance strategies. For each defect, students ascertain the cause and attempt to prevent similar scenarios in future game development projects.

Once they have enough experience, students create defect prevention strategies for a sizeable game development project. The economics of software development is also considered with students solving problems in multiple ways to ascertain the value and cost implications of various strategies.

GD1P04: 3D Graphics Programming

3D Graphics Programming introduces students to the fundamental topics of core computer graphics, 3D graphics programming and the stages of the rendering process. Topics included are the transformation pipeline, device states, primitive rendering, basic camera systems, lighting, texturing and alpha techniques as well as software engineering design principles and testing strategies. By the end of the component, students will create a game utilizing 3D graphics concepts to demonstrate their understanding and put the theory in to practice.

GD2S03: Advanced Software Engineering and Programming for Games

This component teaches students how to create a collection of game asset tools for development teams. Lecturers provide theoretical and practical examples of hardware and software technologies for the rapid development of games. Contemporary technologies are used and student learning is facilitated with in-class debates regarding the usefulness of each technology.

Students will design, construct, test, assess, and enhance an integrated game asset export-tool. They then evaluate their own software development and implementation processes against widely accepted software engineering principles and practices.

GD2P01: Artificial Intelligence

Artificial intelligence systems are ever evolving. In order to pinpoint the correct tool for each job, students identify the strength and weaknesses of various software engineering strategies and evaluate their effectiveness. Lecturers provide case studies and theoretical foundations of various contemporary artificial intelligence engineering practices. They also facilitate in-class discussions, debates and critiques of these practices. Learning is achieved through debating how real world problems should be approached.

GD2P02: Physics Programming

The application of physics to in-game objects has a massive impact on the user experience. This component introduces a variety of technologies and software development strategies for game development with students applying their software process skills, knowledge and modelling techniques to create a simple physics system for a game.

Fundamental techniques include, among others, how to apply Newtonian physics for game development; rigid body dynamics; evaluating and applying various collision detection techniques; and using modelling principles for deterministic physics modelling.

GD2P03: Technology Leverage for Games

A broad range of game development technologies are explored and examined in-depth. Using an existing game framework, students will practice solving simple and complex gaming problems that include looking at a multithreaded game engine, databases and network-based games.

Students gain insights into using assembly for debugging and optimizing purposes along with the ability to write multi-threaded code that can enhance the performance of a game on multi-core machines. They will also reflect on the implications of developing software with legacy systems and prebuilt assets, and experiment with their integration.

GD2P04: Advanced Graphical Games Programming

Advanced Graphics Games Programming takes an in-depth look at the more complex graphical programming and introduces students to tool construction. The required analysis techniques for tools are discussed to streamline design and increase the likelihood of a positive result. Students expand on already existing libraries and create plug-ins for pre-existing technologies. Additionally, students will design, construct, test, and evaluate a 3D scene - drawing on a collection of human-computer interaction, visual design, and game design elements to enhance it. Both visual and non-visual elements that enable the creation of the 3D scene are evaluated.

GD2J01: People And Games

People and Games examines the non-technical, human issues (societal and cultural) of the game industry. A history of the game development industry is taught, which students critically analyse from a historic and contemporary perspective. Lecturers provide theoretical and conceptual frameworks for game design and investigation. In-class discussions are held around game design and technical limitations, with students debating their ideas and defending their decisions. Students' abilities to identify key design features will improve, which is used in later components when students propose design features for games.

GD2J03: Rapid Game Prototype

Students' skills and knowledge are used to plan, produce, test, enhance and manage a group game that can be exhibited in a public forum. Working in teams, students collaborate to self-manage this project utilising an effective software engineering strategy.

Undertaking this project will enhance students' design and management skills, by utilising a variety of software engineering processes (including software design, project management, software processes, bug tracking, etc.).

Year Three

GD3P01: Game Engine Development

In this component students study the fundamentals of real time software systems in the form of a game engine with its enclosed components. Building on the foundations created so far in the programme, students design and develop a game engine collaboratively to facilitate their production of the final year project.

The components of a game engine - optimization issues, systems flow and the "how's" and "what's" of the development are extensively discussed. Student learning is now advanced significantly to the point where they can start specializing in areas of their choice and expressing more individuality in their work.

GD3S01: Software Engineering Capstone Project

The Software Engineering Capstone Project enables students to investigate specific areas of interest towards the application of software engineering principles, practices and expertise in their major production.

Students are encouraged to study the interaction between software engineering and its associated disciplines (e.g. computer science, management, mathematics, and systems engineering). This knowledge can then be applied to enhance and integrate with information from scholarly sources, forming academic research and creating the path towards higher educational opportunities.

GD3S02: Software Engineering Game Development Capstone Project

The Software Engineering Game Development Capstone Project allows a student to select a technical area of game development to specialize in by researching a specific topic. Students select a specialization in an area of technical game development (e.g. game design, game play, mathematical programming, 3D programming and artificial intelligence programming).

The research topic will enhance a student's understanding of complex technical solutions and will allow them to evaluate, interpret, and appraise the game development theories and concepts within their chosen field of study.

GD3J01: Professional Practice

Professional Practice teaches the knowledge, skills and strategies to undertake a collaborative production that is based on sound management theories and advice.

Students use creative and critical thinking methodologies to form a production investigation. Using project management methods, they are taught to plan and manage a project through to completion and gain commercial acumen and understanding of business realities in the process. There is an emphasis on personal responsibility during production because the knowledge, skills and attitudes that are developed are aimed at assisting postgraduate study.

GD3J02: Preproduction

In this component students develop comprehensive design documents to be used in the production of a game, enhancing their skills in the areas of industry procedures and game design principles. Learning of previous materials is drawn on and students debate and justify the contents of their design documents. Lecturers act as producers and assist in the solution of team dynamic problems when needed.

Students should now possess all the skills, knowledge and abilities to undertake this component alone and assistance is primarily provided to ensure the game design is produced within the set timeframe. During this preproduction period, the environment is studio based, helping students prepare for industry realities.

GD3J03: Game Development Team Production Alpha

Students collaborate on a major production in this component, with individuals specialising in their chosen areas of game development and software engineering. They manage, produce, test and enhance a game that matches an agreed specification and design; the game must demonstrate major in-game functionality. Individual students will document their investigatory findings for major problems.

This learning gives provides the analytical skills and knowledge necessary for higher study. Again, whilst support is provided, students are expected to demonstrate the necessary skills to undertake this component on their own. During the production period, students will be based in a studio environment to simulate the realities of a real world situation.

GD3J05: Game Development Team Production Gold

In their final component students produce a release-candidate game that has already gone through several test cycles. In preparing the game for open testing a wide variety of problems will need to be solved both individually and as a team.

Identifying these problems, investigating, discovering solutions, and testing possible solutions throughout the testing phase will enhance the technical skills acquired throughout the course. The final game will be feature complete and have no show-stopper bugs. As with the previous two components, the environment is studio based, again helping prepare students for the realities of game development and production.