| | |
|---|---|
| Course Title: | **Game Programming** |
| Course Code: | **COMP710** |
| Descriptor Start Date: | **01/01/2026** |
| POINTS: | **15.00** |
| LEVEL: | **7** |
| PREREQUISITE/S: | **COMP612** |
| COREQUISITE/S: | **None** |
| RESTRICTION/S: | **None** |

## LEARNING HOURS

Hours may include lectures, tutorials, online forums, laboratories. Refer to your timetable and course information in Canvas for detailed information.

**Total learning hours: 150**

## PRESCRIPTOR

A practical foundation in game programming, using a variety of game development tools and programming libraries, and deploying games on a variety of platforms.

## LEARNING OUTCOMES

1. Analyse, design, document and develop technical solutions for game scenarios.
2. Examine a variety of game-related simulation techniques, data structures and algorithms for implementation in game scenarios.
3. Develop and evaluate memory efficiency, debug, and modify real-time video game software in an industry appropriate language, with professional development tools.
4. Construct clear and efficient game source code, which adheres to a programming standard.
5. Evaluate and utilise third-party APIs and middleware in the development of game scenarios.
6. Plan, schedule, manage, communicate and evaluate on the game development process utilising game project management techniques.

## CONTENT

- Real-time interactive game programming simulation techniques.
- Game programming tools, middleware and libraries.
- Testing and debugging real-time game software.
- Graphical effects programming for games.
- Artificial intelligence programming for games.
- Physics simulation programming for games.
- Network communication programming for games.
- Tools development and scripting for games.
- Game production and project management techniques.
- Game industry culture, documentation and multidisciplinary collaboration.
- History of computer games and computer game platforms.

## LEARNING & TEACHING STRATEGIES

- On Campus Studio Sessions: Each week there will be two lecturer-directed sessions that will include both theory lectures and laboratory-based studio practice exercises. All lecture and exercise materials presented are provided on the Learning Management System. Students will continuously develop their game development skills and knowledge through the completion of short formative programming exercises, and summative assignments.
- Student-Directed Learning: Self-directed research and learning is undertaken in game programming related topics, beyond what is presented in theory lectures. Assessments are completed during self-directed learning hours.
- Personal Portfolio: Throughout the semester students will curate a personal portfolio of game programming artefacts which reflect their learning and development.
- Individual Game Project: Students individually design and develop a game, utilising techniques and technology presented in class. The game showcases game programming aspects learnt during lecture sessions, with scope for additional unfamiliar and sometimes complex game programming solutions.
- Team Game Project: Teams are formed, from which a collaborative game is designed, planned, scheduled and then produced. Individuals share accountability for overall game delivery while taking responsibility for programming various aspects of the overall game. The production process is transparently monitored and reported upon. The game showcases game programming aspects learnt during lecture sessions, with scope for additional unfamiliar and sometimes complex game programming solutions. Team work is undertaken during some of the On Campus Studio Sessions, as well as during student-directed learning hours.

## ASSESSMENT PLAN

| Assessment Event | Weighting % | Learning Outcomes |
|---|---|---|
| Personal Portfolio | 15.00 | 1, 2, 3, 5 |
| Individual Game Project | 35.00 | 1 - 6 |
| Team Game Project | 50.00 | 1 - 6 |

| Grade Map | MAP1 |
|---|---|
| | A+ A A- Pass with Distinction<br>B+ B B- Pass with Merit<br>C+ C C- Pass<br>D Fail |

**Disclaimer: Course descriptors may be amended between teaching periods/semesters**

**Overall requirement/s to pass the course:**

To pass this course, students must achieve a minimum overall grade of C-.

## LEARNING RESOURCES

Prescribed Text: Rabin, S. (2010). Introduction to Game Development (2nd ed.). Portland, OR: Cengage Learning Gregory, J. (2009). Game Engine Architecture. Natick, MA: A. K. Peters/CRC Press Schwab B. (2008). AI Game Engine Programming (2nd ed.). Boston, MA: Cengage Learning Sanchez-Crespo, D. (2003). Core Techniques and Algorithms in Game Programming. Indianapolis, IN: New Riders Games Millington, I. (2007). Game Physics Engine Development. Boston, MA: Morgan Kaufmann Finney, K. (2007). 3D Game Programming – All in One (2nd ed.). Boston, MA: Thomson/Course Technology Sherrod, A. (2008). Game Graphics Programming. Boston, MA: Course Technology/Charles River Media/Cengage Learning Required Resources Classroom and PCs with dedicated graphics cards and graphics rendering libraries

**For further information, contact:** Te Ara Auaha - Faculty of Design & Creative Technologies

| | |
|---|---|
| Principal Programme: | **AK3697, Bachelor of Computer and Information Sciences** |
| Related Programme/s: | **AK3698**<br>**AK1041**<br>**AK3001**<br>**AK3003**<br>**AK3756**<br>**AK3706** |