1. **What is this course about?**

1.1 **Course description**

In this course, you will learn the basics of programming, while developing your own playable text-based adventure game. By the end of the course, you will have a knowledge of the Python programming language, and a firm foundation in programming concepts that you can apply to Unity, or other game development environments.

1.2 **Course content**

Programming concepts
- Variables and Types
- Conditions and Loops
- Functions
- Arrays, Lists, Tuples and Dictionaries
- Object Oriented Programming (OOP) - Objects and Classes
- Properties, Methods and Variable Scope
- Listeners and Broadcasters

Programming skills
- Basics of Programming
- Work from concept, through pseudo-code, to working program code
- Perform mathematical operations including random number generation
- Parse and interpret user-entered text
2. **Unit value**

12 units

3. **How does this course contribute to my learning?**

<table>
<thead>
<tr>
<th>The specific learning outcomes that you will achieve by successful completion of this course:</th>
<th>You will be assessed on the learning outcome in task/s:</th>
<th>Completing these tasks successfully will contribute to you becoming:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply fundamental programming concepts, which apply to all programming languages.</td>
<td>1,2,3</td>
<td>Knowledgeable. Creative and critical thinkers.</td>
</tr>
<tr>
<td>Develop rapid digital game prototypes.</td>
<td>1, 3</td>
<td>Empowered. Creative and critical thinkers.</td>
</tr>
<tr>
<td>Plan how to implement a program, as a step between concept and execution.</td>
<td>2</td>
<td>Empowered. Creative and critical thinkers.</td>
</tr>
<tr>
<td>Develop working programs.</td>
<td>3</td>
<td>Knowledgeable. Creative and critical thinkers.</td>
</tr>
</tbody>
</table>

4. **Am I eligible to enrol in this course?**

Refer to the Undergraduate Coursework Programs and Awards - Academic Policy for definitions of “pre-requisites, co-requisites and anti-requisites”

4.1 **Enrolment restrictions**

Nil

4.2 **Pre-requisites**

Nil

4.3 **Co-requisites**

Nil

4.4 **Anti-requisites**

ICT221

4.5 **Specific assumed prior knowledge and skills (optional)**

Nil
5. How am I going to be assessed?

5.1 Grading scale
Standard – High Distinction (HD), Distinction (DN), Credit (CR), Pass (PS), Fail (FL)

5.2 Assessment tasks

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Assessment Tasks</th>
<th>Individual or Group</th>
<th>Weighting %</th>
<th>What is the duration / length?</th>
<th>When should I submit?</th>
<th>Where should I submit it?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Finish a given game prototype</td>
<td>Individual</td>
<td>25%</td>
<td>N/A</td>
<td>Friday 5:00pm Week 5</td>
<td>Blackboard</td>
</tr>
<tr>
<td>2</td>
<td>Plan a game</td>
<td>Individual</td>
<td>25%</td>
<td>N/A</td>
<td>Friday 5:00pm Week 10</td>
<td>SafeAssign</td>
</tr>
<tr>
<td>3</td>
<td>Program a small game prototype</td>
<td>Individual</td>
<td>50%</td>
<td>N/A</td>
<td>Friday 5:00pm 21st November (end of the Centrally Scheduled Exam Period)</td>
<td>Blackboard</td>
</tr>
</tbody>
</table>

Assessment Task 1: Finish a given game prototype

**Goal:** Rapidly develop and finalise an incomplete game prototype via an application of fundamental programming concepts.

**Product:** You will complete the development of a game prototype, using only the given resources.

**Format:** Your prototype will be finished in the given game framework using the existing code as a base.

**Criteria**

- *You will NOT be assessed on the user experience of the prototype* – it doesn’t matter if it is fun to play, or even if the prototype usefully answers a development issue about the game. You are being assessed solely on your program.
- Application of programming fundamentals including variables, control flow, and functions.
- Proper usage of the existing codebase.

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applying technologies</td>
</tr>
<tr>
<td>Problem solving</td>
</tr>
</tbody>
</table>

Assessment Task 2: Plan a game

**Goal:** Leaping straight into writing code before you know what you’re trying to achieve is bound to result in messy programs. In this task, you will plan out your game, in preparation for writing the code in the next assessment.
Course Outline: SGD203. Introduction to Game Programming

**Product:** You will produce a design document for a small game prototype, which clearly conveys all of the information a programmer would need to implement the game. Use tables and diagrams as appropriate. The game should include a simple game loop and scoring method, which will be discussed in class.

**Format:** The document should be a PDF. This is an individual task.

**Criteria** The design document will be assessed according to the clarity of information it would provide to a programmer who needs to code the game:
- Does it clearly show how input is handled?
- Does it clearly show how the game world updates?
- Does it clearly show how output is displayed?
- Does it clearly show all of the states of the game, including the initial state and end state(s)?
- Is it informative and succinct?

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Generic skill assessed</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organisation</td>
<td>Introductory</td>
</tr>
<tr>
<td>Communication</td>
<td>Introductory</td>
</tr>
</tbody>
</table>

**Assessment Task 3: Program a small game prototype**

**Goal:** For this task, you will bring all of the skills you have learned in this course together, and build a small game prototype, based on the design document from Assessment 2.

**Product:** A game prototype

**Format:** All of your own code should be submitted in a single zip file, including a README.txt which describes the correct method of running and playing the game.

**Criteria** Delivery of the functionality detailed in Assessment 2.
Application of programming fundamentals including variables, control flow, and functions.
Robustness of code
Clarity of code

**Generic skill assessed**

<table>
<thead>
<tr>
<th>Generic skill assessed</th>
<th>Skill assessment level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applying technologies</td>
<td>Developing</td>
</tr>
<tr>
<td>Problem solving</td>
<td>Developing</td>
</tr>
</tbody>
</table>

5.3 Additional assessment requirements

**SafeAssign**
In order to minimise incidents of plagiarism and collusion, this course may require that some of its assessment tasks are submitted electronically via SafeAssign. This software allows for text comparisons to be made between your submitted assessment item and all other work that SafeAssign has access to. If required, details of how to submit via SafeAssign will be provided on the Blackboard site of the course.

**Eligibility for Supplementary Assessment**
Your eligibility for supplementary assessment in a course is dependent of the following conditions applying:
a) The final mark is in the percentage range 47% to 49.4%
Course Outline: SGD203. Introduction to Game Programming

b) The course is graded using the Standard Grading scale
c) You have not failed an assessment task in the course due to academic misconduct

5.4 Submission penalties
Late submission of assessment tasks will be penalised at the following maximum rate:
- 5% (of the assessment task’s identified value) per day for the first two days from the date identified as the due date for the assessment task.
- 10% (of the assessment task’s identified value) for the third day
- 20% (of the assessment task’s identified value) for the fourth day and subsequent days up to and including seven days from the date identified as the due date for the assessment task.
- A result of zero is awarded for an assessment task submitted after seven days from the date identified as the due date for the assessment task.

Weekdays and weekends are included in the calculation of days late. To request an extension you must contact your course coordinator to negotiate an outcome.

6. How is the course offered?
6.1 Directed study hours
On campus computer lab: 3 hours per week

6.2 Teaching semester/session(s) offered
Semester 2

6.3 Course activities

<table>
<thead>
<tr>
<th>Teaching Week / Module</th>
<th>What key concepts/content will I learn?</th>
<th>What activities will I engage in to learn the concepts/content?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Directed Study Activities</td>
</tr>
<tr>
<td>1</td>
<td>Course overview</td>
<td>Write a “Hello World” program.</td>
</tr>
<tr>
<td></td>
<td>Text-based adventure games</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tech overview</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Variables and Types Operators</td>
<td>Create and manipulate variables.</td>
</tr>
<tr>
<td></td>
<td>Operators</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Conditions</td>
<td>Use code blocks.</td>
</tr>
<tr>
<td></td>
<td>Loops</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functions</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Variable Scope</td>
<td>Scope experiments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Revise and apply concepts from weeks 1-4.</td>
<td>Prototyping workshop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Compound variables: Lists, tuples and dictionaries</td>
<td>Work with compound variables.</td>
</tr>
</tbody>
</table>
## Course Outline: SGD203. Introduction to Game Programming

<table>
<thead>
<tr>
<th>Week</th>
<th>Activity</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>String manipulation</td>
<td>Manipulate strings.</td>
<td>Do the exercise again, on your own.</td>
</tr>
<tr>
<td>8</td>
<td>Objects</td>
<td>Examine game engine objects</td>
<td>Do the exercise again, on your own.</td>
</tr>
<tr>
<td>9</td>
<td>Flowcharts and State diagrams</td>
<td>Begin designing and documenting your game.</td>
<td>Continue designing and documenting your game.</td>
</tr>
</tbody>
</table>

### Mid Semester Break

- **10**
  - *Monday, 3rd October Public Holiday*
  - Documentation workshop
  - Documentation workshop
  - Complete and submit your design document.

- **11**
  - Development workshop
  - Development workshop
  - Independent development

- **12**
  - Development workshop
  - Development workshop
  - Independent development

- **13**
  - Development workshop
  - Development workshop
  - Independent development

*Please note that the course activities may be subject to variation.*

### 7. What resources do I need to undertake this course?

#### 7.1 Prescribed text(s)
There are no prescribed texts for this course. All resources needed are freely available online.

#### 7.2 Required and recommended readings
Lists of required and recommended readings may be found for this course on its Blackboard site. These materials/readings will assist you in preparing for tutorials and assignments, and will provide further information regarding particular aspects of your course.

#### 7.3 Specific requirements
Nil

#### 7.4 Risk management
There is minimal health and safety risk in this course. It is your responsibility to familiarise yourself with the Health and Safety policies and procedures applicable within campus areas. Whenever using computers for prolonged periods, take regular screen breaks and set up your work environment according to ergonomic principles.
8. **How can I obtain help with my studies?**
In the first instance you should contact your tutor, then the Course Coordinator. Additional assistance is provided to all students through Peer Advisors and Academic Skills Advisors. You can drop in or book an appointment. To book: Tel: +61 7 5430 2890 or Email: student central@usc.edu.au

9. **Links to relevant University policies and procedures**
For more information on Academic Learning & Teaching categories including:
- Assessment: Courses and Coursework Programs
- Review of Assessment and Final Grades
- Supplementary Assessment
- Administration of Central Examinations
- Deferred Examinations
- Student Academic Misconduct
- Students with a Disability


10. **General enquiries**
In person:
- **Sippy Downs** - Student Central, Ground Floor, Building C
- **USC SouthBank** - Student Central, Building B, Ground floor (level 1)
- **USC Gympie** - Student Central, 71 Cartwright Road, Gympie
- **USC Fraser Coast** - Student Central, Building A

Tel: +61 7 5430 2890
Email: studentcentral@usc.edu.au