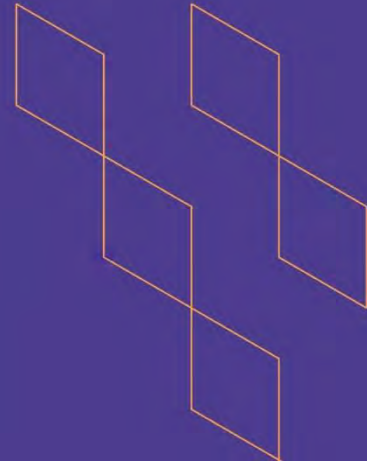T-104
2022

Course Specification

Course Title: **Computer Games**

Course Code: IT1768

Program: **Information Technology**

Department: **Information Technology**

College: **Faculty of Computer Science and Information Technology**

Institution: **Albaha University**

Version: **version1**

Last Revision Date: **29 - 03 – 2023**

Table of Contents:

A. General information about the course:

Course Identification				
1. Credit hours:	3 H			

2. Course type

a.	University ☐	College ☐	Department☐	Track☐	Others☐
b.	Required ☐	Elective☒			

3. Level/year at which this course is offered:	Elective

4. Course general Description

This subject introduces game development through the use of a commercial-grade game engine. It aims to build a generalizable understanding of the theory of virtual interactive simulations, while also more specifically developing intermediate competence in programmatic problem solving within the Unity Game Engine. The subject includes exposure to common terminology used in the industry, an introduction to the core interface tools of an engine, experience in applying vector and matrix mathematics to transform virtual objects in 2D and 3D coordinate space, and a strong focus on programming in games through the use of an engine API. This establishes the skills required to successfully contribute to the development of significant group projects found in many of the subsequent games related subjects.

5. Pre-requirements for this course (if any):

Data Structures   **CS1256**

6. Co- requirements for this course (if any):

7. Course Main Objective(s)

Upon successful completion of this subject students should be able to:

1. Understand foundational language and theory of game development technology.

2. Apply mathematical and game programming knowledge to solve development tasks.

3. Navigate the interface and codebase of an industry standard game development engine.

4. Proactively identify their own knowledge gaps and establish skills in life-long learning through sourcing new knowledge to solve previously unseen intermediate problems.

   5. Design and implement minor variations to existing game designs.

   6. Design Oriented: graduates apply problem solving, design and decision-making methodologies to develop components, systems and processes to meet specified requirements.
5.

7. Technically Proficient: graduates apply abstraction, mathematics and discipline fundamentals, software, tools and techniques to evaluate, implement and operate systems.

8. Reflective: graduates critically self-review their performance to improve themselves, their teams, and the broader community and society.

1. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1.	Traditional classroom	33	100%
2.	E-learning		
3.	Hybrid • Traditional classroom • E-learning		
4.	Distance learning		

2. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	22
2.	Laboratory/Studio	22
3.	Field	
4.	Tutorial	
5.	Others (specify)	
	Total	44

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Understand foundational language and theory of game development technology.	K2	Lectures · Assignments · Lab Exercises	Quizzes · Midterm exams · Final Exam
1.2	Apply mathematical and game programming knowledge to solve development tasks.	K1	Lectures · Assignments · Lab Exercises	Quizzes · Midterm exams · Final Exam
1.3	Navigate the interface and codebase of an industry standard game development engine.	K1	Lectures · Assignments · Lab Exercises	Quizzes · Midterm exams · Lab Exam · Final Exam
2.0				
2.1	Proactively identify their own knowledge gaps and establish skills in life-long learning through sourcing new knowledge to solve previously unseen intermediate problems.	S2	Lectures · Assignments · Lab Exercises	Quizzes · Midterm exams · Lab Exam · Final Exam
2.2	Design and implement minor variations to existing game designs.	S4	Lectures · Assignments · Lab Exercises	Quizzes · Midterm exams · Lab Exam · Final Exam
2.3	Design Oriented: graduates apply problem solving,	S5	Lectures · Assignments	Quizzes

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
	design and decision-making methodologies to develop components, systems and processes to meet specified requirements			· Midterm exams · Lab Exam · Final Exam
3.0	Values, autonomy, and responsibility			
3.1	7. Technically Proficient: graduates apply abstraction, mathematics and discipline fundamentals, software, tools and techniques to evaluate, implement and operate systems.	V1	Assignments Oral Presentations	Reports · Presentations · Class Discussions
3.2	8. Reflective: graduates critically self-review their performance to improve themselves, their teams, and the broader community and society.	V2	Assignments Oral Presentations	Reports · Presentations · Class Discussions

C. Course Content

No	List of Topics	Contact Hours
1	Introduction to Unity – Unity interface, coordinate system, game cycle, components, GameObject, Transform	2
2	Fundamentals – asset instantiation, add/get component, input management, vectors, translation/rotation/scale	2
3	Working with Time – time, delta time, invoke, co-routines	2
4	In-engine Animation – interpolation, programmatic tweening	2
5	Animation and Audio Assets – Playing animations/audio, animation state machines, animation function calls	2
6	UI Programming – UI canvas space, event registration, event listening	2
7	Game State Managers – C# enumerators, switch statements, game managers, scene management	2
8	Collisions - collision detection, triggers, kinematics, layers, the collision matrix	2
9	Physics Programming – rigid body physics, physics materials	2
10	Save Games – PlayerPrefs, resources folder, asset loading/unloading, JSON utility, file writing	2
11	C# Extras – Attributes, ternary operator, statics, delegates, events	2
	Total	22

No	List of Topics (LAB)	Contact Hours
1.	Introduction to Unity – Unity interface, coordinate system, game cycle, components, GameObject, Transform	3
2.	Fundamentals – asset instantiation, add/get component, input management, vectors, translation/rotation/scale	4
3.	Working with Time – time, delta time, invoke, co-routines	4
4.	In-engine Animation – interpolation, programmatic tweening	4
5.	Animation and Audio Assets – Playing animations/audio, animation state machines, animation function calls	4
6.	UI Programming – UI canvas space, event registration, event listening	3
	Total	22

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Weekly Homework and assignments	Weekly	10%
2.	Midterm Exam	7th week	20%
3.	Quiz	9th week	10%
4	Lab Exam	10th week	20%
5	Final Exam	11th week	40%

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.)

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	Multiplatform Game Development in C# (by Joseph Hocking)
Supportive References	C# Cookbook, 2nd Edition
Electronic Materials	Access to the Saudi Digital Library (SDL). • Using the learning management system of the university – Rafid System (https://lms.bu.edu.sa/). • ACM (Association for Computer Machinery) web site - http://www.acm.org/ • ACM SIGCSE (Special Interest Group on Computer Science Education) resource web site - http://www.sigcse.org/SIGresources • IEEE Computer Society web site - http://www.computer.org/portal/web/guest/home • Intel *The Journey Inside* web site (has a collection of interactive, online lessons about technology, computers, and society) - http://educate.intel.com/en/TheJourneyInside/ • *Google Code University* Curriculum Resource web site - http://code.google.com/edu/resources/index.html
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	A classroom or lecture hall with whiteboard. • A laboratory with computers that have installed Windows, MAC OSX, and Linux • An instructor computer station with · High speed Internet connection; · A desktop computer with system administration software installed; · Power outlets for instructor's laptop plug-in; · A digital image projection system with connection and switches to desktop computer, laptop computer and DVD/Blu Ray player.
Technology equipment (projector, smart board, software)	All students shall have · A computer with Microsoft

Items	Resources
	Project © software installed; This software comes with the textbook. · High speed Internet connection; · Power outlets for student's laptop plug-in.
Other equipment (depending on the nature of the specialty)	- A laboratory with multiple computers - A whole IT infrastructure

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students Peer Reviewer Program Leader	Indirect: Survey Direct: Peer Review Direct: Class Visits
Effectiveness of students assessment	Exams Evaluation Committee Students	Direct: Exam Review Indirect: Survey
Quality of learning resources	Faculty	Direct: Exams
The extent to which CLOs have been achieved	Jodah Platform	Electronic
Other		

Assessor (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)
Assessment Methods (Direct, Indirect)

G. Specification Approval Data

COUNCIL /COMMITTEE	
REFERENCE NO.	
DATE	30 – 3 – 2023