ALAGAPPA UNIVERSITY

(Accredited with A+ Grade by NAAC (CGPA: 3.64) in the Third Cycle, Graded as Category-I University and granted autonomy by MHRD-UGC)

DIRECTORATE OF COLLABORATIVE PROGRAMMES

Bachelor of Science in Game Programming

Regulations and Syllabus

[For those who join the Course in July 2023 and after]

CHOICE BASED CREDIT SYSTEM

Regulations and Syllabus

GENERAL INSTRUCTIONS AND REGULATIONS

B.Sc. Game Programming conducted by Alagappa University, Karaikudi, Tamil Nadu through its Collaborative Institution.

Applicable to all the candidates admitted from the academic year **2023**onwards.

1. **Eligibility:**

   A pass in the Higher Secondary Examination (HSC) conducted by the Government of Tamil Nadu, or an examination accepted as equivalent thereto by the Syndicate for admission to this programme.

2. **For the Degree:**
   The candidates shall have subsequently undergone the prescribed program of study in an institute for not less than three academic years, passed the examinations prescribed and fulfill such conditions as have been prescribed thereof.

3. **Admission:**
   Admission is based on the marks in the qualifying examination.

4. **Duration of the course:**
   The course shall extend over a period of **Three years** under Semester pattern.

5. **Standard of Passing and Award of Division:**
   a. Students shall have a minimum of 40% of total marks of the University examinations in each subject. The overall passing minimum is 40% both in aggregate of Continuous Internal Assessment and external in each subject.
   b. The minimum marks for passing in each theory / Lab course shall be 40% of the marks prescribed for the paper / lab.
   c. A candidate who secures 40% or more marks but less than 50% of the aggregate marks prescribed for three years taken together, shall be awarded **THIRD CLASS.**
   d. A candidate who secures 50% or more marks but less than 60% of the aggregate marks prescribed for three years taken together, shall be awarded **SECOND CLASS.**
   e. A candidate who secures 60% or more of the aggregate marks prescribed for three years taken together, shall be awarded **FIRST CLASS.**
   f. Only Part-III subjects will be considered for the University academic ranking purpose.
   g. The Practical / Project shall be assessed by the two examiners, by an internal examiner and an external examiner.

   Continuous internal Assessment:

   a. Continuous Internal Assessment for each paper shall be by means of Written Tests, Assignments, Class tests and Seminars
   b. **25 marks** allotted for the Continuous Internal assessment is distributed for Written Test, Assignment, Class test and Seminars.
   c. Internal Assessment - Break-Up of Marks, suggested pattern (Faculty may change the pattern, according to the subject and need)
      a. Two Internal Tests (choose one best out of two) – 50%
      b. Model Test (One model test) – Nil – Should be conducted prior to the University examination. It is a mandate.
      c. Assignments – 25%
      d. Seminar / Case Study – 25%
   d. Conduct of the continuous internal assessment shall be the responsibility of the concerned faculty.
   e. The continuous internal assessment marks should be submitted to the University at the end of every semester, before the commencement of Semester Exams.
   f. The valued answer papers/assignments should be given to the students after the valuation is over and they should be asked to check up and satisfy themselves about the marks they have scored.
   g. All mark lists and other records connected with the continuous internal assessments should be in the safe custody of the institution for at least one year after the assessment.

6. **Attendance:**

Students must have earned 75% of attendance in each course for appearing for the examination.
Students who have earned 74% to 70% of attendance have to apply for condonation in the prescribed form with the prescribed fee.
Students who have earned 69% to 60% of attendance have to apply for condonation on Medical grounds in the prescribed form with the prescribed fee along with the medical certificate / relevant documents.
Students who have below 60% of attendance are not eligible to appear for the examination. They shall re-do the semester(s) after completion of the programme.

7. **Examination:**

Candidate must complete course duration to appear for the university examination. Examination will be conducted with concurrence of Controller of Examinations as per the Alagappa University regulations. **University may send the representatives as the observer during examinations.** University Examination will be held at the end of the each semester for duration of 3 hours for each subject. Certificate will be issued as per the AU regulations. **Hall ticket will be issued to the students at the end of every semester after submitting "No Dues" certificate to the exam cell, under the aegis of Controller of Examinations of the AU.**

8. **Question Paper pattern:**

Maximum: 75 Marks Duration: 3Hours

Part A - Short answer questions with no choice	: 10 x 02=20
Part B –Brief answer with either or type	: 05 x 05=25
Part C- Essay – type questions of either / or type	: 03 x 10=30

9. **Miscellaneous**

a. Every student should possess the prescribed text book for all the subjects, through-out the semester for their theory/lab classes.
b. Every student would be issued an Identity card by the institute/university to identify his/her admission to the course.
c. Every student shall access the library and internet (wi-fi) facilities provided for the self-development and career-development.
d. Every student who successfully completes the course within the stipulated time period would be awarded the degree by the University.

10. **Fee structure**

Course fee shall be as prescribed by the University and 50% of the course fee should be disbursed to University. Special fees and other fees shall be as prescribed by the Institution and the fees structure must be intimated to the University. Course fees should be only by Demand draft / NEFT and AU has right to revise the fees accordingly.

Semester Pattern

Course Fee payment deadline
Fee must be paid before 10th September of the academic year

11. **Other Regulations:**

Besides the above, the common regulation of the University shall also be applicable to this programme.

826 - B. Sc Game Programming - Programme structure

S.No.	Part	Course Code	Courses	Title of the Paper	T/P	Cr.	Hrs./ Week	Max. Marks		
								Int.	Ext.	Total
Semester-I										
1	I	82611T/11H/11F	T/OL	Tamil /Other Languages - I	T	3	4	25	75	100
2	II	82612	E	General English-I	T	3	4	25	75	100
3	III	82613	Core 1	Fundamentals of Programming	T	4	4	25	75	100
4		82614	Core 2	Fundamentals of Programming - Practical	P	4	8	25	75	100
5		82615	Allied 1	Game Analysis and Design	T	3	3	25	75	100
6		82616	Allied 2	Game Analysis and Design - Practical	P	2	4	25	75	100
7	IV	82617	SEC -I	Value Education	T	2	2	25	75	100
8				Library			1			
				Total		**21**	**30**	**175**	**525**	**700**
Semester-II										
1	I	82621T	T/OL	Tamil/Other Languages-II	T	3	4	25	75	100
2	II	82622	E	General English-II	T	3	4	25	75	100
3	III	82623	Core 3	Graphics Programming	T	4	4	25	75	100
4		82624	Core 4	Graphics Programming - Practical	P	4	8	25	75	100
5		82625	Allied 3	Algorithms and Data Structures	T	3	3	25	75	100
6		82626	Allied 4	Algorithms and Data Structures - Practical	P	2	4	25	75	100
7	IV	82627	SEC -II	Environmental Studies	T	2	2	25	75	100
8				Library			1			
9		82628A 82628B		Internship/ Mini Project	I/ PR	2	--	25	75	100
				Total		**23**	**30**	**200**	**600**	**800**
Semester-III										
1	I	82631T	T/OL	Tamil/Other Languages-II	L	3	4	25	75	100
2	II	82632	E	General English-III	L	3	4	25	75	100
3	III	82633	Core 5	Game Engine-I	T	3	3	25	75	100
4		82634	Core 6	Advanced Game Math	T	3	3	25	75	100

				and Physics						
5		82635	Core 7	Game Engine-I - Practical	P	3	5	25	75	100
6		82636	Allied 5	Game Networking Techniques	T	3	3	25	75	100
7		82637	Allied 6	Multiplayer Game Development- Practical	P	2	4	25	75	100
8	IV	82638	SEC-III	Entrepreneurship	T	2	2	25	75	100
9	IV	82639A 82639B 82639C	NME- I	1.Adipadai Tamil	P	2	2	25	75	100
				2.Advance Tamil	T					
				3.IT Skills for Employment	T					
				4. MOOC'S	T					
				Total		**24**	**30**	**225**	**675**	**900**
				Semester-IV						
1	I	82641T	T/OL	Tamil /Other Languages - IV	T	3	4	25	75	100
2	II	82642	E	General English-IV	T	3	4	25	75	100
3	III	82643	Core 8	Game Engine-II	T	4	4	25	75	100
4		82644	Core 9	Web Game Development	T	4	4	25	75	100
5		82645	Core 10	Game Engine-II - Practical	P	3	5	25	75	100
6		82646	Allied 7	Mobile Game Development	T	3	3	25	75	100
7		82647	Allied 8	Mobile and Web Game Development-Practical	P	2	4	25	75	100
8	IV	82648A 82648B 82648C	NME- II	1.Adipadai Tamil	P	2	2	25	75	100
				2.Advance Tamil	T					
				3. Small Business Management	T					
				4. MOOC'S	T					
9		82649		Internship	I	2	-	25	75	100
				Total		**26**	**30**	**225**	**675**	**900**
				Semester-V						
1	III	82651	Core 11	Artificial Intelligence for Games	T	4	4	25	75	100
2		82652	Core 12	Game Programming Patterns	T	4	4	25	75	100
3		82653A 82653B	DSE 1	A. Sound Design for Games	T	4	4	25	75	100

		82653C		B. Shader Programming C. Game Engine Customization						
4		82654A 82654B 82654C	DSE 2	A. Game Market Analysis and Monetization B. Game Engine Architecture C. Emerging Trends in Game Development	T	4	4	25	75	100
5		82655A 82655B 82655C	DSE 3	A. Cinematics in Games- Practical B. Level Design and Environmental Creation- Practical C.Game Testing and Profiling- Practical	P	4	4	25	75	100
6		82656	Core 13	Portfolio & Presentation	P	4	8	25	75	100
7				Career development/employability skills			2			
				Total		**24**	**30**	**150**	**450**	**600**
Semester-VI										
1	III	82661	Core 14	Game Writing Essentials	T	4	4	25	75	100
2		82662	Core 15	Advanced Game Mechanics	T	4	4	25	75	100
3		82663	Core 16	Game Mechanics Development- Practical	P	4	6	25	75	100
4		82664A 82664B 82664C	DSE 4	A. Animation for Games- Practical B. Storyboarding for Games- Practical C. Game User Interface Design- Practical	P	4	4	25	75	100
5		82665A/ 82665B	Core 17	Project/ Dissertation	PR/ D	6	12	25	75	100
				Total		**22**	**30**	**125**	**375**	**500**
	Grand Total					**140**	**180**	**1100**	**3300**	**4400**

DSE – Student Choice and it may be conducted by parallel sections.

** NME –Students have to select courses offered by other (Faculty) departments.

*** SLC – Voluntary basis

T – Theory P – Practical

<table>
<tr><th colspan="6">I – Semester</th></tr>
<tr><td rowspan="2">Core</td><td rowspan="2">Course code: 82613</td><td rowspan="2">FUNDAMENTALS OF PROGRAMMING</td><td>T</td><td>Credits: 4</td><td>Hours: 4</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Objectives</td><td colspan="5">1. Learn programming basics and control structures, including functions and recursion.
2. To educate a strong command of array manipulation and pointer usage, covering dynamic arrays, function pointers, and array-to-function passage
3. To acquire an in-depth comprehension of Object-Oriented Programming (OOP) principles
4. Acquire advanced proficiency in applying file handling principles within C++ programming
5. Develop a comprehensive understanding of utilizing the Standard Template Library (STL)</td></tr>
<tr><td>Unit I</td><td colspan="5">Programming Basics: Programming Hello world - Data types - Variables - Constants - Operators-Conditional Statements – Looping - Functions - Understanding Functions - Pass values to functions – Inline function - Recursive functions</td></tr>
<tr><td>Unit II</td><td colspan="5">Arrays: One Dimensional - Two Dimensional - Multi Dimensional - Dynamic arrays - Pointers- Pointers Advantage & disadvantage - Variable pointers - Generating pointer to an array - Function Pointers - Array pointers - Pointers to Pointers - Functions - Passing pointers to functions-Returning pointers - Passing Arrays to functions</td></tr>
<tr><td>Unit III</td><td colspan="5">OOPS Principles: Classes - Objects - Encapsulation - Constructors - Destructors – Polymorphism–Types of polymorphism – Abstraction - Virtual Function - Function Overloading - Overriding- Inheritance</td></tr>
<tr><td>Unit IV</td><td colspan="5">File handling: Read and Write operations - Designing and Structuring a Project - Hierarchy- Namespaces - Exception Handling - Templates - Delay and Timer functions - Enumerations - Data Handling using Files</td></tr>
<tr><td>Unit V</td><td colspan="5">Standard Template Library: Containers – Sequences – Vector – List – deque - ContainerAdaptors – Stack – Queue - Algorithms - Mutating Algorithms – Swap – Replace - Remove - Sorting- Binary Search – Merge - Function Object - Random Number Generator - Iterators - Forward- Random Access - Data Structures Types - Linear Data</td></tr>
<tr><td colspan="6">Reference and Text Books
Object-Oriented Programming with C++" by E. Balagurusamy
Bjarene Stroustrup, “Programming: Principles and practices using C++”, Addison-Wesley Professional, 2008.
E. Balagurusamy, “Computing Fundamentals & C Programming, Tata McGraw-Hill, 2ndEdition, 2008.
Herbert Scheldt, “The Complete Reference C++”, Tata McGraw Hill, 2002. · Scott Meyers, Effective STL”, Strangecat Publication, 2001.
Yashwant Kanetkar, “Let Us C++ Solutions”, BPB, 2010.</td></tr>
<tr><td colspan="6">Online Resources
https://onlinecourses.swayam2.ac.in/aic20_sp06/preview
https://onlinecourses.swayam2.ac.in/arp19_ap79/preview</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	Apply programming basics, use operators, conditionals, and loops effectively, understand and utilize functions, and apply recursion for problem-solving.	**K3**
CO-2	Demonstrate array handling techniques and utilize pointers and proficiency in passing pointers and arrays to functions.	**K3**
CO-3	Learners gain understanding of Object-Oriented Programming (OOP) principles	**K2&K3**
CO-4	Demonstrate adeptness in performing file operations, including reading, writing, and managing data, effectively employing file handling techniques.	**K6**
CO-5	Apply STL components effectively, demonstrating the ability to utilize pre-built data structures and algorithms, leading to efficient code development	**K4**

Course Outcome VS Programme Outcomes

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	S(3)	M(2)	L(1)	M(2)	M(2)	M(2)	L(1)	M(2)	L(1)	M(2)
CO2	S(3)	S(3)	L(1)	L(1)	M(2)	L(1)	L(1)	M(2)	L(1)	M(2)
CO3	M(2)	M(2)	M(2)	M(3)	L(1)	M(2)	M(2)	M(2)	M(2)	M(2)
CO4	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)	L(1)
CO5	S(3)	S(3)	L(1)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
W.AV	**2.6**	**2.4**	**1.4**	**2.2**	**2**	**2**	**2**	**2.2**	**2**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	M(2)	L(1)	S(3)	L(1)	L(1)
CO2	M(2)	L(1)	S(3)	L(1)	L(1)
CO3	M(2)	L(1)	M(2)	M(2)	L(1)
CO4	M(2)	L(1)	S(3)	L(1)	M(2)
CO5	M(2)	L(1)	S(3)	L(1)	M(2)
W.AV	**2**	**1**	**2.8**	**1.2**	**1.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="5">I-Semester</th></tr>
<tr><td>Course code</td><td rowspan="2">FUNDAMENTALS OF PROGRAMMING - PRACTICAL</td><td rowspan="2">P</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>82614</td><td>4</td><td>8</td></tr>
<tr><td>Objectives</td><td colspan="4">➢ Design programs with user input, calculations, and interactive responses.
➢ Employ conditional statements and branching logic for interactive game creation.
➢ Utilize loop structures proficiently to manage repetition and control program flow.
➢ Develop programs to read, process, and write data for specific outcomes.
➢ Design and implement class hierarchies and inheritance for modeling complex systems.</td></tr>
<tr><td colspan="5">1. Program to calculate the area and perimeter of different shapes based on user input.
2. Write a program to rock-paper-scissors game: Implement a game where the player chooses rock, paper, or scissors and plays against the computer.
3. Create a program to guess the number game: a program where the computer generates a random number and the player has to guess it, with hints if the guess is too high or too low.
4. RPG character stats: define functions to calculate and display stats for a role-playing game character.
5. Hangman game: Implement a simple hangman game where the player guesses letters to complete a word from an array of words.
6. High Scores Tracker: Create a program to track and display the top scores of players using an array.
7. Write a program for player class: design a class that represents a player in a game, encapsulating attributes like name, score, and health.
8. Create a program for Zoo simulation: model a zoo using classes with inheritance, like base Animal class and derived classes for specific animal types.
9. Write a program that reads data from a file, processes it, and writes the results back to another file.
10. Student Database: Design a program to manage a student database with features like adding, deleting, and displaying student records.</td></tr>
<tr><td>Outcomes</td><td colspan="4">Craft user-friendly interfaces, incorporate input effectively, perform accurate calculations, and present results coherently.
Cultivate dynamic decision-making skills, implement effective conditional logic, and construct engaging gameplay experiences.
Attain deep comprehension of loop mechanisms, create optimized algorithms for repetitive tasks, and confidently manage loop behavior.
Excel in data handling from files, implement processing algorithms, and derive insightful conclusions through data manipulation.
Internalize object-oriented principles, construct modular class structures, and adeptly employ abstraction for real-world modeling.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	M(2)	S(3)	L(1)	L(1)	M(2)	M(2)	L(1)
CO2	S(3)	S(3)	M(2)	M(2)	S(3)	L(1)	L(1)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	M(2)	M(2)	S(3)	L(1)	L(1)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**2.2**	**2**	**3**	**1**	**1.4**	**2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	S(3)	S(3)	S(3)	M(2)	M(2)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	M(2)
W.AV	**3**	**3**	**2.6**	**2.2**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

I – Semester					
Allied	**Course code: 82615**	**Game Analysis and Design**	**T**	**Credits: 3**	**Hours:3**
Objectives	1. To learn communication principles and basic game mechanics, demonstrating comprehension through simple game design and interactive elements 2. To comprehend the social function of games and their elements 3. Grasp game world dynamics, including transmedia elements, character attributes, spatial design, aesthetics, and art-technology integration 4. To understand player experiences and game design concepts. 5. To acquire knowledge about player classification, interactions, ethics, and communities in game design.				
Unit I	**Introduction to communication:** Interactive and New Media - Human Computer Interaction Fundamentals - Ethics of New Media **Introduction to Games:** Evolution of Games - Basic terminologies - Types of Games- Game Genres - Three Practical Approaches - Core Dynamics - MDA - Mechanics, Dynamics- Aesthetics - MDA at work - Tuning - Flow - Types of Fun -Types of Players - Skill vs Difficulty- Affordability - Orthogonality - Tension maps in Game Design - Circumspection.				
Unit II	**Social function of Games**: Dramatic Elements of Game - Structuring a Game - Linear Plot - Braided Plot - Branching Tree - Networks - Open Worlds - The Loop of Interaction - Channels of Information Gameplay - Chance - Probability - Alea - Strategy - Skill - Adding and Subtracting Mechanics- Emergence and Progression in Games - Integrating Emergence and Progression				
Unit III	**The Game World:** Transmedia World - Properties - Common Elements of Successful Worlds- Nature of Game Characters - Spaces - Architecture - Organizing Game Space - Real vs. Virtual Architecture - Level Design - World Aesthetics - Value of Aesthetics - Audio of Environment - Letting Aesthetics Guide the Design - Balancing Art and Technology				
Unit IV	**Games and Experience:** Player's Experience - Modeling - Focusing - - Empathizing - Imagination- Motivating - Judgement - Game Mechanics - Space - Objects, Attributes and States - Actions - Rules- Skill - Chance - Interest Curves - Patterns inside Patterns - Factors of Interest - Game Balancing Methodologies - Balancing Game Economics - Dynamic Game Balancing.				
Unit V	**Know your Players:** Taxonomy of Players - Changing the Player Type Balance - Player Interactions-Flow of Influence - Dynamics of Player Taxonomy - Demographics - Psychographics - Ethics in Game Design - Ergodic, Code and Other Laws of Computer Game Design - Ethical Instances - Player Communities - Strong Communities				

Reference and Text Books

"Game Design: Principles and Practice" Michael Salmond, Jeannie Novak, and Ananda Shankar Chakrabarty, Course Technology

"Fundamentals of Game Design" Ernest Adams and Andrew Rollings, New Riders

Anderson EF, Engel S, McLoughlin L, Comninos P. The case for research in game engine Architecture.

Andrew Rollings, Dave Morris, Game Architecture and Design - A New Edition, Newriders, 1st edition, 2003

Castillo, T., & Novak, J, "Game development essentials: game level design", Delmar Learning, 2008.

Heather Maxwell Chandler, The Game Production Handbook, Jones & Bartlett Publishers, 3rd edition, 2013.

Johannes Fromme, Alexander Unger, Computer Games and New Media Cultures: AHandbook

of Digital Games Studies, Springer Science & Business Media, 2012.	
Online Resources https://onlinecourses.swayam2.ac.in/aic20_ed01/preview https://books.google.co.in/books/about/It_s_All_a_Game.html?id=3shyDQAAQBAJ&redir_esc=y	
Course Outcomes	**Knowledge level**
CO-1	Gain a foundational understanding of communication and game concepts.	K2 & K3
CO-2	Apply dramatic elements, plot structures, and emergence/progression concepts to craft compelling gameplay	K3
CO-3	Apply this knowledge to craft immersive and harmoniously balanced game environments	K4
CO-4	Create engaging games using mechanics, spatial elements, and balancing techniques	K4
CO-5	Create engaging games that cater to diverse players and ethical considerations	K5

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	L(1)	L(1)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	S(3)
CO2	L(1)	L(1)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)
CO3	L(1)	L(1)	S(3)	S(3)	L(1)	M(2)	M(2)	M(2)	M(2)	S(3)
CO4	L(1)	L(1)	S(3)	S(3)	L(1)	S(3)	S(3)	M(2)	M(2)	S(3)
CO5	L(1)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**1**	**1.1**	**3**	**3**	**1.4**	**2.8**	**2.6**	**2.2**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	S(3)	M(2)	L(1)	S(3)
CO2	L(1)	S(3)	M(2)	L(1)	S(3)
CO3	L(1)	S(3)	M(2)	L(1)	S(3)
CO4	L(1)	S(3)	M(2)	L(1)	S(3)
CO5	L(1)	S(3)	M(2)	L(1)	S(3)
W.AV	**1**	**3**	**2**	**1**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

I-Semester					
Allied	**Course Code: 82616**	**GAMES ANALYSIS AND DESIGN - PRACTICAL**	**P**	**Credits:**	**Hours:**
				2	**4**
Objectives	1. To introduce various art forms and styles, enabling students to appreciate the wide range of creative expressions. 2. To learn about human anatomy's significance in art, enhancing their ability to depict realistic and proportionate figures. 3. To teach students to break down complex body parts into simple 2D shapes, aiding them in structured figure drawing. 4. Through practical exercises, students apply anatomical knowledge to their artwork, honing their skills in portraying the human body. 5. Students gain insights into how different cultures and time periods have influenced artistic representations of the human form, enriching their artistic perspective.				
1. Create a face using images of fruits and vegetables. 2. Use a close up photo of you and enhance one half of your face. 3. Create a poster for the Movie / Game title specified by the tutor. 4. Redesign a popular logo. 5. Download photographs of two animals and create a new animal using features from the downloaded animals. 6. Create a Manga character using your photographs for reference.					
Outcomes	1. To develop an understanding and enjoyment of art and design. 2. Study formal aspects of diverse art movements. 3. To learn how to use texturing and coloring effectively. 4. To understand how texture and color relate to the subject. 5. To enhance critical observation of artworks.				

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	L(1)	L(1)	S(3)	L(1)	S(3)	L(1)	M(2)	M(2)	M(2)	M(2)
CO2	L(1)	L(1)	S(3)	L(1)	S(3)	L(1)	M(2)	M(2)	M(2)	M(2)
CO3	L(1)	L(1)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)	L(1)	M(2)
CO4	L(1)	M(2)	S(3)	M(2)	S(3)	M(2)	S(3)	M(2)	L(1)	M(2)
CO5	L(1)	M(2)	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)
W.AV	**1**	**1.4**	**3**	**1.8**	**3**	**1.6**	**2.4**	**2**	**1.6**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	S(3)	L(1)	M(2)	L(1)
CO2	L(1)	M(2)	L(1)	L(1)	L(1)
CO3	L(1)	S(3)	M(2)	M(2)	L(1)
CO4	L(1)	M(2)	L(1)	L(1)	L(1)
CO5	L(1)	S(3)	M(2)	M(2)	M(2)
W.AV	**1**	**2.6**	**1.4**	**1.6**	**1.2**

S–Strong (3), M-Medium (2), L-Low (1)

II – Semester - Core					
Core	**Course Code:82623**	**GRAPHICS PROGRAMMING**	**T**	**Credits:**	**Hours:**
				4	**4**
Course Objectives	**1. To understand graphics programming essentials, including libraries, terminology, and program structure.** **2. To learn the fundamentals of vertex and color manipulation in graphics programming, covering shapes, rendering pipelines, buffers, and user interaction.** **3. To attain proficiency in transformation matrices for graphics manipulation, covering translation, rotation, scaling, and camera operations..** **4. To understand 3D models, covering loading, rendering optimization, collision detection, and occlusion culling techniques.** **5. Excel in shader programming fundamentals and advanced rendering methods.**				
UNIT-I	**Introduction to Graphics Programming**: Graphics Libraries - Need for a Graphics Library- Computer Graphics Terminologies - Coordinate spaces - Point, Vector, Vertex - Structure Of a Graphics Program - Various APIs for Graphics Programming - Creating Window - Game Loop- Input Manipulation				
UNIT-II	**Vertex and Color:** Hello Triangle - Drawing a Quad - Draw Primitives - Rendering Pipeline - Fixed Vs Programmable pipeline - Introduction to Buffers - Vertex buffer - Index Buffer - Viewport - Projection - Managing Aspect ratio and FOV for various effects -User Interaction				
UNIT-III	**Transformation matrix:** Translation matrix - Rotate Matrix - Scale Matrix - Skew Matrix - Basic Virtual Camera - Eye, LookAt and Up Vector - MVP Matrix - Various camera operations - Local Coordinate, World Coordinate and Screen Coordinate - Converting Screen to World Coordinate				
UNIT-IV	**Understanding a 3D model:** Loading 3D Models - Back face Culling- Player movement - Terrain - Simple terrain Creation - Height Maps - Using Height Maps - Collision With terrain - Multi-Textured Terrain - Blend Map Creation - Shaders for using Blend Map data - AABB - Bounding box for objects - Bounding Tree - BSP Trees for Collision Detection - Optimized Render Cycle - BSP for Rendering Objects - Occlusion Detection - Occlusion Culling				
UNIT-V	**Shader Basics:** Using Multiple Shaders - Multi texturing for Terrain - Blend Maps - Phong and Gouraud shading - Per-pixel lighting - Specular lighting - Multiple Lights - Point Lights - Sky-Box- Day/Night - Cell Shading - Transparency - Fog - Mipmapping - Rendering to Texture - Shadow Mapping - Anti Aliasing and Anisotropic Filtering - Bump mapping- Normal Mapping - Bloom effect - Specular Mapping - 3D Particle Effects				

Text Book:

- "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V", John Kessenich, Graham Sellers, and Dave Shreiner, Addison-Wesley Professional

References:

- Angel, E., "OpenGL". Pearson Education, 2011.
- Fabio Ganovelli, "Introduction to Computer Graphics: A Practical Learning Approach(Chapman & Hall/CRC Computer Graphics, Geometric Modeling, and Animation)", Chapmanand Hall/CRC, 2014.
- Ginsburg, D, "OpenGL ES 3.0 Programming Guide", Addison-Wesley Professional, 2014. Graham Sellers, "OpenGL Superbible: Comprehensive Tutorial and Reference (7th Edition)", 7 Edition Addison-Wesley Professional, 2015.
- James M. Van Verth, "Essential Mathematics for Games and Interactive Applications", ThirdEdition. 3 Edition. A K Peters/CRC Press, 2015.
- Eric Lengyel, "Mathematics for 3D Game Programming and Computer Graphics", ThirdEdition. Cengage Learning PTR, 2011.
- Fletcher Dunn, "3D Math Primer for Graphics and Game Development", 2nd Edition, AKPeters/CRC Press, 2011.

Online Resources

- https://onlinecourses.nptel.ac.in/noc22_cs111/preview
- OpenGL

Course Outcomes		**Knowledge level**
CO-1	**To apply knowledge to develop graphics programs, coordinate systems, and implement input manipulation.**	**K3**
CO-2	**Implement concepts to craft shapes, manage rendering, utilize buffers, and incorporate user interaction**	**K3**
CO-3	**To utilize matrix concepts to manipulate objects, perform camera actions, and convert coordinate systems effectively**	**K4**
CO-4	**Able to utilize their acquired knowledge to effectively load, render 3D models, optimize rendering cycles, implement collision detection, and use occlusion culling in graphics programming**	**K3**
CO-5	**Implement shaders to achieve lifelike rendering, including lighting, textures, effects, and enhancements**	**K5**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	M(2)	S(3)	M(2)	S(3)	L(1)	L(1)	M(2)	M(2)	S(3)
CO2	S(3)	M(2)	S(3)	M(2)	S(3)	L(1)	M(2)	S(3)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	S(3)	L(1)	M(2)	M(2)	M(2)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	L(1)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	L(1)	S(3)	S(3)	S(3)
W.AV	**3**	**2.6**	**3**	**2.2**	**3**	**1.4**	**1.4**	**2.6**	**2.4**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	L(1)	S(3)	M(2)	M(2)
CO2	S(3)	M(2)	S(3)	L(1)	L(1)
CO3	S(3)	L(1)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	S(3)	L(1)	M(2)
CO5	S(3)	L(1)	S(3)	M(2)	L(1)
W.AV	**3**	**1.4**	**3**	**1.6**	**1.6**

S–Strong (3), M-Medium (2), L-Low (1)

II – Semester - Core					
Core	**Course Code: 82624**	**GRAPHICS PROGRAMMING - PRACTICAL**	**P**	**Credits: 4**	**Hours: 8**

Objectives

- **Demonstrate understanding of collision detection, sprite animation, parallax scrolling, and projectile motion principles.**
- **Implement player movement mechanics, dodge, and crouch actions.**
- **Showcase proficiency in loading and navigating 3D environments.**
- **Apply raycasting for object picking and interactive text display.**
- **Utilize directional and spot lights to enhance visual realism in open-world creation.**

1. Demonstrate Collision Detection with an example
2. Demonstrate sprite animation with an example.
3. Demonstrate Parallax Scrolling.
4. Demonstrate Projectile motion.
5. Create a Player. Add 8-directional player movement.
6. Load a 3D world Blockout and Create a First Person Walkthrough.
7. Demonstrate Object Picking with Raycast.
8. Display Text in windows and add some effects to it.
9. Create a Player and Demonstrate Dodging and Crouch Movement.
10. Create an open world environment and add directional and spot light.

Outcomes

- **Develop interactive scenarios that exemplify collision detection, sprite animation, parallax scrolling, and projectile motion.**
- **Design a player character with dynamic movement, dodge, and crouch functionalities.**
- **Construct a first-person walkthrough of a 3D world using loaded blockout environments.**
- **Create interactive experiences by selecting objects through raycasting and implementing visually enhanced text displays.**
- **Craft an open-world environment with directional and spot lighting for heightened realism and user engagement.**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	L(1)	M(2)	S(3)	M(2)	L(1)	M(2)	M(2)	M(2)
CO2	S(3)	S(3)	L(1)	M(2)	S(3)	L(1)	L(1)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)
CO4	S(3)	S(3)	L(1)	M(2)	S(3)	L(1)	L(1)	S(3)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
W.AV	**3**	**3**	**1.4**	**2.2**	**3**	**1.6**	**1.4**	**2.6**	**2.6**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	M(2)
CO2	M(2)	S(3)	M(2)	L(1)	L(1)
CO3	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	M(2)	L(1)
CO5	M(2)	S(3)	M(2)	L(1)	M(2)
W.AV	**2.4**	**3**	**2**	**1.4**	**1.6**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">II – Semester - Allied</th></tr>
<tr><td rowspan="2">Allied</td><td rowspan="2">Course Code: 82625</td><td rowspan="2">ALGORITHMS AND DATA STRUCTURES</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>Course Objectives</td><td colspan="5">1. To build a solid understanding of algorithmic problem-solving principles.
2. Develop proficiency in various algorithmic techniques for problem-solving.
3. To attain proficiency in diverse algorithmic approaches.
4. To gain expertise in Dynamic Programming and fundamental algorithms such as Binomial Coefficients and graph algorithms.
5. Acquire in-depth knowledge of Backtracking, Branch and Bound, and Decision Trees algorithms, addressing complex problems including N-Queens, Hamiltonian circuits, Subset Sum, Assignment, Knapsack, and Traveling Salesman.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Notion of Algorithm: Fundamentals of Algorithmic Problem Solving – Important Problem Types–Analysis Framework – Asymptotic Notations and Basic Efficiency Classes – Mathematical Analysis of Recursive and Non-Recursive Algorithms – Algorithm Visualization</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Brute Force: Selection Sort and Bubble Sort – Sequential Search and Brute Force String Matching–Closest Pair – Exhaustive Search - Divide and Conquer – Merge sort – Quick sort – Binary search–Binary Tree Traversal</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Decrease and Conquer: Insertion Sort – Depth First Search and Breadth First Search - Transform And Conquer – Presorting – AVL Trees – Heaps and Heap sort</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Dynamic Programming: Computing a Binomial Coefficient – Warshall’s and Floyd’s algorithm–Optimal Binary Search Trees – Greedy Technique - Prim’s algorithm – Kruskal’s algorithm–Dijkstra’s algorithm</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Backtracking – N-Queens problem – Hamiltonian circuit problem – Subset Sum Problem– Branch And Bound – Assignment problem – Knapsack problem – Traveling salesman problem- Decision Trees - P & NP Problems – NP Complete problems – Approximation algorithms for NP-hard problems – Traveling salesman problem – Knapsack problem</td></tr>
<tr><td colspan="6">Textbooks:
● Ellis Horowitz, Satraj Sahni and Sanguthevar Rajasekaran, Fundamentals of Computer Algorithms. Galgotia Publications.
● Langsam, Y., Augenstein, M. J., & Tenenbaum, A. M. (1996). Data Structures using C and C++. Prentice Hall Press.
References:
● AhoAlfred,V.,Hopcroft John,E.,Ullman Jeffrey. Data Structures and algorithms.USA:Addison-Wesley.
● Goodman,S.E.,& Hedetniemi,S.T. Introduction to the Design and Analysis of Algorithms. McGraw-Hill,Inc.
● Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems (Vol. 5, pp. 79-104). New York: Springer.</td></tr>
<tr><td colspan="6">Online Resources
● https://nptel.ac.in/courses/106102064
● https://books.google.co.in/books/about/Data_Structures_and_Algorithms_in_C++.html?id=q_NxAjqWDGoC&redir_esc=y</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	Demonstrate proficiency in utilizing essential algorithms, evaluating efficiency, employing asymptotic notations, and visualizing algorithmic processes	**K2**
CO-2	To implement and compare sorting algorithms, string matching techniques, and demonstrate understanding of Divide and Conquer strategies, and Binary Tree traversal	**K3**
CO-3	Excel in Insertion Sort, graph traversal, Transform And Conquer, AVL Trees, Heaps, and Heap Sort.	**K2**
CO-4	Demonstrate proficiency in employing dynamic programming for problem-solving, implementing graph algorithms for tree structures and shortest paths, and assessing their practical implications	**K5**
CO-5	Exhibit proficiency in implementing advanced algorithms, analyzing their computational efficiency, and understanding the intricacies of P & NP problems and NP-Complete problems	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	L(1)	L(1)	S(3)	M(2)	M(2)
CO2	S(3)	S(3)	M(2)	L(1)	M(2)	L(1)	L(1)	S(3)	M(2)	M(2)
CO3	S(3)	S(3)	L(1)	M(2)	M(2)	L(1)	L(1)	S(3)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	L(1)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	L(1)	L(1)	M(2)	M(2)	M(2)	S(3)	S(3)	S(3)
W.AV	**3**	**3**	**1.6**	**1.4**	**2**	**1.4**	**1.4**	**3**	**2.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	L(1)	L(1)	L(1)	L(1)
CO2	L(1)	L(1)	M(2)	L(1)	L(1)
CO3	M(2)	L(1)	M(2)	L(1)	L(1)
CO4	L(1)	M(2)	M(2)	L(1)	L(1)
CO5	M(2)	M(2)	M(2)	M(2)	M(2)
W.AV	**1.4**	**1.4**	**1.8**	**1.2**	**1.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">II – Semester - Allied</th></tr>
<tr><td rowspan="2">Allied</td><td rowspan="2">Course Code: 82626</td><td rowspan="2">ALGORITHMS AND DATA STRUCTURES-PRACTICAL</td><td rowspan="2">P</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>2</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Develop proficiency in implementing and manipulating fundamental data structures such as stacks, queues, linked lists, binary search trees, and graphs.
➢ Acquire a deep understanding of essential algorithmic problem-solving techniques including graph traversals, knapsack problems, and sorting algorithms.
➢ Master the creation, insertion, deletion, and searching operations in various data structures.
➢ Gain expertise in applying data structures and algorithms to solve real-world programming challenges.
➢ Explore advanced concepts like Huffman coding, Prim's algorithm, and Hamiltonian cycles to enhance problem-solving skills.</td></tr>
<tr><td colspan="6">Data Structure:
1. Stack: Creation, Push and Pop, Conversion and evaluation of Prefix and Postfix expression
2. Queues: Creation, Insertion, Deletion
3. Linked list: Creation, Insertion and Deletion using Singly Linked List, Circular List and Doubly Linked list.
4. Binary Search Tree: Creation, Searching and Deleting an element
5. Graphs: Depth-First Search (DFS) and Breadth-First Search (BFS).
Algorithms:
1. Knapsack problem
2. Prim’s algorithm
3. All pairs shortest paths
4. 8 Queens problem
5. Huffman Coding
6. Hamiltonian Cycle
7. Sorting – Heap, Merge, Selection, Quick</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Demonstrate practical skills in designing and implementing data structures and algorithms for efficient data manipulation and storage.
➢ Apply acquired knowledge to develop optimized solutions for various algorithmic challenges and programming tasks.
➢ Implement effective strategies for searching, sorting, and managing data using the learned data structures.
➢ Analyze and evaluate algorithmic complexities to make informed decisions on selecting appropriate data structures and algorithms for different scenarios.
➢ Showcase an improved ability to approach complex problems systematically and develop efficient and elegant solutions using the acquired knowledge of data structures and algorithms.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	L(1)	L(1)	L(1)	S(3)	M(2)	L(1)
CO2	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	S(3)	M(2)	S(3)
CO3	S(3)	S(3)	L(1)	M(2)	M(2)	L(1)	L(1)	S(3)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	M(2)	L(1)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**1.6**	**2.2**	**1.6**	**1.4**	**1.4**	**3**	**2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	M(2)	M(2)	L(1)	M(2)
CO2	M(2)	M(2)	M(2)	L(1)	L(1)
CO3	M(2)	L(1)	M(2)	M(2)	L(1)
CO4	S(3)	M(2)	M(2)	L(1)	M(2)
CO5	M(2)	L(1)	M(2)	L(1)	M(2)
W.AV	**2.4**	**1.6**	**2**	**1.2**	**1.6**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">III – Semester-Core</th></tr>
<tr><td rowspan="2">Core</td><td rowspan="2">Course Code: 82633</td><td rowspan="2">GAME ENGINE-I</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>Course Objectives</td><td colspan="5">➢ Acquire foundational knowledge of 3D game development, including distinctions between 2D and 3D games, screen-to-world positioning, and working with 3D models.
➢ Develop proficiency in terrain design, game environment setup, scripting, and optimization.
➢ Master game development concepts like namespaces, coroutines, raycasting, animation control, and 3D physics.
➢ Attain proficiency in camera manipulation, GUI integration, cinematic techniques, and various rendering processes.
➢ Learn game UI design, HUD communication, sound integration, networking basics, cross-platform development, and code organization.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to 3D Game Development: Concepts of 2D vs 3D Game.
Understanding the 3DGameWorld: screen dimensions - Convert screen positions to world positions - Working in 3DScene- 3DGame Objects.
Importing Models: Model, Rig, and Animations - Understanding Meshes: Polygonal meshes, mesh models, mesh import - Components.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Terrain Design: Designing Level Maps - Setting up the Game Environment.
Profiler Window: Input Settings, Console - Prefabs and Tags.
Scripting: Basic 3D Methods - Collision Detection - Triggers.
Controlling Game Objects Behavior: Rendering Mesh, Mesh filter.
Event Handling: Mouse, Keyboard, Touch - Handling Frame Rate and performance.</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Namespaces, List Collections: Generic Functions - Coroutines and Exceptions - Raycasting- Navigation and Pathfinding - Working with Animation - Controlling Animation - 3D Physics - Joints-Types of Joints - Exploring different Colliders</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Camera: Camera Properties, Lens Flare - GUI - Cinematics: Rendering to Texture - Particle Effects- Global Illumination - Rendering sky - Implementing render passes - Lighting, Shading - Occlusion Culling - Optimize event management - Check for memory leaks - Memory Optimization</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Designing Game UI: Basic UI Layout - Designing Game UI - Information sharing to HUD- Sound and Music - Networking Concepts: server, host, spawn, Instantiate - Building for Different Platforms-Clean up code.</td></tr>
<tr><td colspan="6">Reference and Text Books:
● Alan Thorn, “UDK Game Development”, Course technology, 2012.
● Aung Sithu Kyaw, Clifford Peters, Thet Naing Sw, Unity 4.x, 2013.
● Deborah Todd, “Game Design: From Blue Sky to Green Light”, 2007.
● Lee Zhi Eng, “Building a Game with Unity and Blender”, 2015.
● Michelle Menard, “Game Development with Unity”, Course technology, 2012.</td></tr>
<tr><td colspan="6">Online Resources
● https://learn.unity.co
● https://www.ebooks.com/en-in/book/209638243/unity-game-development-cookbook/paris-buttfield-addison/</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Demonstrate comprehension in discerning 2D and 3D game aspects, translating screen positions to world positions, managing 3D scenes, and handling 3D models effectively**	**K3**
CO-2	**Design level maps, create game environments, script 3D interactions, handle collision detection, triggers, and optimize performance**	**K6**
CO-3	**Apply knowledge to create dynamic game environments, manage animations, implement physics, and navigate with pathfinding techniques**	**K4**
CO-4	**Demonstrate the ability to create immersive scenes, implement cinematic effects, optimize rendering, manage lighting and shading, and optimize memory usage.**	**K4**
CO-5	**Design UI layouts, share info via HUD, add audio, grasp networking, adapt to platforms, maintain clean code.**	**K4**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	M(2)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
W.AV	**3**	**2.8**	**3**	**2.6**	**3**	**2.6**	**2**	**2.6**	**2.6**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	S(3)	S(3)
CO2	S(3)	S(3)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	M(2)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**2**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

III – Semester-Core					
Core	**Course Code: 82634**	**ADVANCED GAME MATH AND PHYSICS**	**T**	**Credits: 3**	**Hours: 3**
Objectives	● **Understand Linear Algebra and Affine Algebra, covering number systems, matrices, vectors, coordinate systems, and transformations.** ● **Acquire expertise in vector operations, including advanced properties and quaternions, as well as mastering rotation matrices for 3D transformations.** ● **Analyze and interpret the dynamics of rigid bodies, showcasing comprehension of fundamental physics concepts.** ● **Utilize vector calculus and fluid mechanics concepts to analyze and model fluid flow phenomena**.				
UNIT-I	**Linear Algebra:** A Review of Number Systems - Systems of Linear Equations - Matrices - Vector Spaces - Advanced Topics. **Affine Algebra**: Introduction - Coordinate Systems - Cartesian Coordinates - Subspaces - Transformations - Barycentric Coordinates.				
UNIT-II	**Vectors**: Basic operations and properties – Advanced operations and properties - Approximation- Quaternions - Rotation Matrices - The Classical Approach - A Linear Algebraic - Approach- Interpolation of Quaternions - Derivatives of Time-Varying Quaternions				
UNIT-III	**Basic Concepts from Physics**: Rigid Body Classification - Rigid Body Kinematics - Newton's Laws- Forces - Momenta - Energy - Rigid Body Motion - Newtonian Dynamics - Lagrangian Dynamics- Euler's Equations of Motion				
UNIT-IV	**Deformable Bodies:** Introduction - Elasticity, Stress, and Strain - Mass–Spring Systems - Control Point Deformation - Free-Form Deformation - Implicit Surface Deformation				
UNIT-V	**Fluids and Gases:** Vector Calculus - Strain and Stress - Conservation Laws - A Simplified Model forFluid Flow - Implementing the Simplified 2D Model - Implementing the Simplified 3DModel - Variations of the Simplified Model				

Reference and Text Books:

- Hartle JB, "Gravity: An introduction to Einstein's general relativity", 2003.
- O'Donnell LJ, Westin CF, "An introduction to diffusion tensor image analysis", NeurosurgeryClinics. 2011.
- Schouten JA. "Ricci-calculus: an introduction to tensor analysis and its geometrical applications", Springer Science & Business Media, 2013.
- Halliday, D., Resnick, R., & Walker, J. "Fundamentals of physics extended". John Wiley&Sons, 2010.
- Spiegel, M. "Schaum's outline of theory and problems of vector analysis and an introduction to tensor analysis", 1974.

Online Resources

- https://www.oreilly.com/library/view/beginning-math-and/0735713901/
- https://docs.unity3d.com/Manual/PhysicsSection.html

Course Outcomes		Knowledge level
CO-1	**Apply these concepts to solve equations, work with matrices and vectors, and grasp geometric transformations in various contexts.**	**K2**
CO-2	**Apply advanced vector manipulation techniques and quaternion operations to effectively model and simulate complex spatial transformations in computer graphics and animation.**	**K3**
CO-3	**Employ Newtonian and Lagrangian approaches to predict the motion of interconnected rigid bodies, illustrating the utility of Euler's equations in accurate motion description.**	**K3**
CO-4	**Apply elasticity, stress, and strain concepts to assess mass-spring systems and deformation methods like control points, free-form, and implicit surfaces.**	**K3**
CO-5	**Use strain, stress, and conservation laws to analyze fluid behavior, implement simplified 2D and 3D fluid flow models, and predict fluid flow patterns in various scenarios.**	**K4**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M(2)	S(3)	L(1)	L(1)	L(1)	L(1)	M(2)	M(2)	L(1)	S(3)
CO2	M(2)	S(3)	M(2)	M(2)	M(2)	L(1)	M(2)	M(2)	L(1)	S(3)
CO3	M(2)	S(3)	M(2)	M(2)	M(2)	L(1)	M(2)	M(2)	M(2)	S(3)
CO4	M(2)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	S(3)
CO5	M(2)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)
W.AV	**2**	**3**	**2**	**2.2**	**2.2**	**1.4**	**2.4**	**2.2**	**1.6**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	M(2)	M(2)	L(1)	L(1)	M(2)
CO2	M(2)	M(2)	L(1)	L(1)	M(2)
CO3	M(2)	S(3)	L(1)	L(1)	M(2)
CO4	M(2)	S(3)	M(2)	M(2)	M(2)
CO5	M(2)	S(3)	M(2)	M(2)	M(2)
W.AV	**2**	**2.6**	**1.4**	**1.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

III-Semester -Core					
Core	**Course Code: 82635**	**GAME ENGINE-I -PRACTICAL**	**P**	**Credits: 3**	**Hours: 5**
Objectives	➢ **Develop terrain creation skills within a game engine.** ➢ **Design a First Person Shooter level for immersive gameplay.** ➢ **Integrate custom models from a design tool into a game engine.** ➢ **Incorporate animated characters seamlessly into your level.** ➢ **Craft a new GUI and HUD for enhanced user interaction in a game engine.**				

1. Create a terrain using game engine
2. Create a First Person Shooter level
3. Import custom models from a design tool to game engine
4. Import animated character and use it in your level
5. Create a new GUI and HUD for your game and import it in game engine
6. Create a 2D character for a 2D casual game
7. Import 2D character to use it inside your game
8. Make a side scrolling game

Outcomes	➢ **Create diverse terrains with realistic elements using game engine tools.** ➢ **Construct engaging First Person Shooter levels with strategic design.** ➢ **Import custom-designed models, enriching the game environment.** ➢ **Utilize animated characters effectively to enhance storytelling.** ➢ **Design intuitive GUIs and HUDs for a better player experience**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	L(1)	M(2)	S(3)	S(3)
CO2	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	L(1)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	L(1)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**3**	**1.4**	**3**	**2.6**	**1.6**	**2**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	S(3)	M(2)
CO2	S(3)	S(3)	M(2)	S(3)	M(2)
CO3	S(3)	S(3)	M(2)	S(3)	M(2)
CO4	S(3)	S(3)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)
W.AV	**3**	**3**	**2.4**	**3**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">III – Semester-Allied</th></tr>
<tr><td rowspan="2">Allied</td><td rowspan="2">Course Code: 82636</td><td rowspan="2">GAME NETWORKING TECHNIQUES</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>Objectives</td><td colspan="5">● Familiarize with essential computer network components and principles for effective communication.
● Acquire knowledge of OSI layers, protocols, and network protection.
● Gain insight into network multiplayer game structures and concepts for effective game design and development.
● Learn to establish effective multiplayer project setups by understanding network behavior and implementing essential components.
● Understand and apply network communication principles for multiplayer game development, encompassing callbacks, scene synchronization, lobby setup, and host migration.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Computer networks: Network Topology - IEEE Standards - Hub - Switch - Router- Modem - Network Card - Bridges - Routing Algorithms - Protocols - Encoding and Decoding- Multiplexing/De-Multiplexing -Data Security - Encryption/Decryption - Authentication</td></tr>
<tr><td>UNIT-II</td><td colspan="5">OSI Layers: Bluetooth Network - Wireless Network - Mobile Network - TCP - UDP - Bit Stream- Error Detection and Correction - Network security and firewalls - WEP - WPA - WPA2 - PublicandPrivate key encryption</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Types of Network Multiplayer Games: Popular Network Multiplayer Games - Network System Concepts - Client Server - Hosting - Local Client and Remote Client - Player Object - CommandandAuthority - Non Player Characters/Objects and Authority - Network Context</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Multiplayer Project setup: Network Behavior - Setting up a Network Player - Game State Management - Spawning - Scene Management - Matchmaking - Customizing - SpawningwithAuthority - Remote Actions - Commands - Client RPC [Remote Procedure Call] - Arguments of RPC</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Network Communication: Network Manager Callbacks - Network Behavior Callbacks - NetworkMessages - Discovering Local Players - Scene Object - Multiplayer Lobby - Network Clients andServers - Host migration - Migration Manager Callbacks</td></tr>
<tr><td colspan="6">Reference and Text Books:
● Andrew S. Tanenbaum, “Computer Networks”, Prentice Hall, 4th Edition, 2002. · Behrus A. Forouzan et al, “Data Communication and Networking”, 2nd Edition, TataMcGraw-Hill, 2000.
● Brian Schwab, “Fundamentals of Network Game Development”, Cengage Learning, 2008. · Doug Lowe, “Networking All-in-One For Dummies”, For Dummies, 5th Edition, 2012.
● Rabin S, editor, “Introduction to game development”, Boston: Charles River Media, 2005</td></tr>
<tr><td colspan="6">Online Resources
● https://docs-multiplayer.unity3d.com/</td></tr>
<tr><td colspan="5">Course Outcomes</td><td>Knowledge level</td></tr>
<tr><td>CO-1</td><td colspan="4">Understand network components, security measures, and device functions, including protocols, encryption, and authentication.</td><td>K2</td></tr>
<tr><td>CO-2</td><td colspan="4">Identify Bluetooth, wireless, and mobile networks across OSI layers, explain TCP, UDP, error handling, and discuss security tools like WPA2, firewalls, and encryption methods.</td><td>K3</td></tr>
<tr><td>CO-3</td><td colspan="4">Differentiate game types, understand client-server, player objects, and non-player characters.</td><td>K4</td></tr>
</table>

CO-4	**Build network players, manage states, handle spawning and scenes, employ matchmaking, and execute remote actions, including spawning with authority, commands, and client RPCs with arguments.**	**K6**
CO-5	**Proficiently design and implement multiplayer features including network behavior callbacks, local player discovery, lobby creation, and migration management, showcasing practical multiplayer game development skills.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	L(1)	L(1)	S(3)	L(1)	M(2)	L(1)	M(2)	L(1)	M(2)
CO2	S(3)	L(1)	L(1)	S(3)	L(1)	M(2)	L(1)	M(2)	L(1)	M(2)
CO3	S(3)	L(1)	L(1)	S(3)	L(1)	M(2)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	M(2)	S(3)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	M(2)
CO5	S(3)	M(2)	S(3)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	M(2)
W.AV	**3**	**1.4**	**1.8**	**2.6**	**1.8**	**1.6**	**1.4**	**2**	**1.8**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	L(1)	M(2)	L(1)	L(1)
CO2	L(1)	L(1)	M(2)	L(1)	L(1)
CO3	L(1)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**1.8**	**1.8**	**2**	**1.8**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">III-Semester - Allied</th></tr>
<tr><td>Allied</td><td>Course Code: 82637</td><td>MULTIPLAYER GAME DEVELOPMENT- PRACTICAL</td><td>P</td><td>Credits: 2</td><td>Hours:4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Implement a functional real-time chat system enabling players to communicate in-game.
➢ Create an engaging multiplayer player versus player (PVP) shooter game with seamless multiplayer mechanics.
➢ Develop a cooperative multiplayer dungeon crawler that encourages teamwork and exploration.
➢ Design an immersive multiplayer racing game offering diverse vehicles and tracks for competitive play.
➢ Build a team-based strategy game where players collaborate strategically to achieve victory.</td></tr>
<tr><td colspan="6">1. Real-time Chat System: Implement a real-time chat system where players can communicate with each other using text messages within the game.
2. Multiplayer PVP Shooter: Create a multiplayer player versus player (PVP) shooter game with multiple players battling each other in a virtual environment.
3. Cooperative Dungeon Crawler: Develop a cooperative multiplayer game where players work together to explore dungeons, defeat enemies, and complete quests.
4. Racing Game: Design a multiplayer racing game where players can compete against each other in various types of vehicles and tracks.
5. Team-Based Strategy Game: Build a strategy game where players are divided into teams, each working together to conquer territories and achieve objectives.
6. Card Game: Develop a multiplayer card game where players can play against each other in real time, utilizing different strategies to win.</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Establish a functional and interactive communication channel for players to interact (Create, Apply).
➢ Develop a dynamic multiplayer environment that fosters competitive engagement (Create, Analyze).
➢ Foster teamwork and coordination among players in a collaborative dungeon exploration setting (Create, Understand).
➢ Provide an exhilarating multiplayer racing experience with varied challenges and competition (Create, Apply).
➢ Promote strategic thinking and cooperation among players in a team-based strategy game (Create, Analyze).</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	M(2)	S(3)	L(1)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	L(1)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	L(1)	M(2)	S(3)	S(3)
W.AV	**3**	**2.8**	**3**	**1.6**	**2.8**	**2.6**	**1.2**	**2**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	M(2)	M(2)
CO2	S(3)	S(3)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	M(2)	S(3)	S(3)
CO4	S(3)	S(3)	L(1)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**2**	**2.8**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

IV – Semester-Core					
Core	**Course Code: 82643**	**GAME ENGINE-II**	**T**	**Credits:**	**Hours:**
				4	**4**
Objectives	● **To understand the essential concepts and tools of game engine usage, including installation, asset creation, and basic scene manipulation.** ● **To learn advanced game development techniques: terrain creation, visual effects, cinematic production, audio integration, and optimization.** ● **Gain expertise in blueprint scripting for game mechanics, AI, and UI design, along with packaging and exporting games.** ● **Attain proficiency in VFX, mechanics, abilities, UI, and level design within game development.** ● **Attain proficiency in game development through the creation of enemy AI, level design, interactive elements, UI, and lighting.**				
UNIT-I	**Introduction to Game Engine:** Installation Process - Project Creation - User Interface Overview-Transform tools - Primitive Geometry - Geometry Editing - Introduction to content browser - BSPSurface - Static Mesh. **Introduction to lighting:** Importing custom static mesh - Creating Material - Diffuse Texture - Landscape Editing Basics.				
UNIT-II	**Importing and Using Height maps:** Terrain Material, Using The Foliage Editor - Normal Maps - emissive Maps - Decals and Opacity masks - Vertex painting, Using Video Texture. **Introduction to sound:** Destruction Meshes - Matinee - Introduction-Creating Cinematic and cut scene - Using Particle Systems - Matinee soundtracks - Matinee Skeletal Mesh Animation - Fade Director Tracks - Audio Master Tracks - Volume Introduction - Post Processing - Level Streaming Quick Start - Creating Prefab-Creating Water with Swimming Feature.				
UNIT-III	**Introduction to blueprint:** Blueprint classes - Blueprint input key binding - Blueprint VariableTypes and Math Functions - How To Create AI And Enemy Basics - Setting Up AI Roaming and Destinations- Health System. **Introduction To UI Widgets:** Creating A HUD - Creating HUD Bindings - Basic UM GUI Animation - Floating UI Widget Component - Loading Screens - Main Menu - Styling MainMenu- Adding Main Menu Functionality - Gamepad Inputs - Showing Game Mouse Cursor - PauseMenuFunctionality - Styling Pause Menu - Packaging and Export - Settings				
UNIT-IV	**Cascade VFX:** Spark Emitter - Cascade GPU Sprites - Cascade Mesh Emitters - Save/Load Game - SaveGame Data - Check Point System - Teleporting Players - Side Scroller Game - Basic Mechanics and Health - Working on The Fuel System - The Health bar - The Fuel Bar - Pickup Items. **Game Countdown Timer**: Speed Boost Ability - Gravity Boost Ability - Slow Motion Ability - Level CompleteScreen - Time Up Screen -Death Animation and Function - Exploding Obstacle - Damaging Player WithFire - Low Health Vignette Effect - Opening Door With Key - Coin Pickup and Counter - MainMenu - Level Selection - Ability Cool Down System - Animated Cool Down Timer				
UNIT-V	**Creating Basic Enemy Bot AI:** Regenerating Health System - Blocking Out The Level - Creating a moving Platform - Crushing Pillar - Using Structural Meshes - Decorating Our Level - Ability Popup Messages - Animated Popup Messages - Death / Game Over Screen - Lighting Our Level - Creating the Flashlight - Adding The Battery - Cleaning Up Our Blueprints.				

Reference and Text Books:

- Alan Thorn, "UDK Game Development", Course technology, 2012.
- Lee, J, "Learning Unreal Engine Game Development", Packt Publishing Ltd, 2016.
- Plowman, J, "3D game design with Unreal Engine 4 and Blender", PacktPub, 2016.
- Satheesh, P. V, "Unreal Engine 4 Game Development Essentials", Packt Publishing Ltd, 2016.
- Thomas Mooney, "Unreal Development Kit Game Design Cookbook", Packt PublishingLtd, 2012

Online Resources

- https://www.unrealengine.com/en-US/learn

Course Outcomes		**Knowledge level**
CO-1	**Able to navigate the game engine interface, create and modify basic game assets, and explain the significance of different components within a game development environment.**	**K3**
CO-2	**Showcase proficiency in height maps, material creation, visual enhancements, cinematic sequencing, audio integration, ParticleSystems, level optimization, and water mechanics for game development.**	**K3 to K5**
CO-3	**Students will proficiently create blueprints, design AI behaviors, craft UI elements, and package/export functional game projects using blueprint scripting.**	**K5**
CO-4	**To design VFX using Cascade, implement game mechanics like abilities and pickups, create engaging UI elements including timers and counters, and construct well-structured levels with interactive features.**	**K4**
CO-5	**Implement basic enemy AI, design interactive levels with moving platforms and hazards, integrate UI elements like ability and popup messages, apply dynamic lighting, and manage blueprint organization.**	**K5**

Course Outcome VS Programme Outcomes

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	S(3)	M(2)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO2	S(3)	M(2)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	S(3)	M(2)	S(3)	M(2)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)
CO4	S(3)	M(2)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
W.AV	**3**	**2.2**	**3**	**2.4**	**3**	**3**	**1.8**	**2**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	S(3)	S(3)	M(2)	S(3)	S(3)
CO2	S(3)	S(3)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	L(1)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**1.8**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">IV – Semester-Core</th></tr>
<tr><td rowspan="2">Core</td><td rowspan="2">Course Code: 82644</td><td rowspan="2">WEB GAME DEVELOPMENT</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand HTML5 enhancements and Canvas fundamentals.
● Achieve expertise in advanced JavaScript and web development concepts.
● Develop interactive web pages with diverse features and functionalities.
● To learn gameplay programming techniques for creating dynamic games.
● To educate physics programming with Box2D for dynamic web applications.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">HTML 5 Introduction: Difference between HTML 4 & HTML 5 - Semantic Tags - Header and Footer- Nav tag - Section - Article - Content - Aside - Media Tags - Audio tag - Properties - VideoTag- Properties.
Canvas: Introduction - SVG Vs Canvas - Application of Canvas - Canvas DOM- HelloWorld in Canvas.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Advanced Java Script: Document Object Model - Introduction - Arrays - One Dimensional Array- Two Dimensional Array - Callback Functions - Form Handling - Get/Post Method - Form Validation- HTML Events - OOPS with JavaScript - Web Development Frameworks - Java script Frameworks</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Building Interactivity in web pages: Scrolling effects - Image Sliders and Image Manipulation- File Handling - XML Parsing - JSON Parsing - Canvas Game Development - Drawing Basic Shapes- Drawing Text , Sprites - Sprite Sheets - Sprite Animations - Keyboard Event Handling</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Gameplay programming: Player Movement - Background Scrolling - Implementing Jump - Collision Detection - Circle Collision Detection - Square Collision Detection - Designing GameUI - Implementing Interactions - Keyboard Event - Mouse Event - Listeners - Implement System Controlled Game Elements - Implementing Timer - Managing Lives and Health - Maintaining Score Information</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Physics Programming: Box2D for Web - Basic Setup and World Definitions - Pre defined functions- Debug Draw - World Render - Collision Detection - Asynchronous web page updates - Introduction- Application - Request and Response.</td></tr>
<tr><td colspan="6">Reference and Text Books:
● Alexis Goldstein- Louis Lazaris-Estelle Weyl,“HTML5 & CSS3 For The Real World”, SitePoint, 2015.
● David Sawyer McFarland, “JavaScript &JQuery:The Missing Manual”, Pogue Press, 2ndEdition, 2011.
● Douglas Crockford, “JavaScript: The Good Parts”, O'Reilly Media, 2008.
● Joe Burns, ”Web site design goodies”, Que Corp., 2001.
● Makzan, “HTML5 Game Development by Example”, Packt Publishing, 2011.</td></tr>
<tr><td colspan="6">Online Resources
● Web game Development</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Differentiate HTML4 and HTML5, apply semantic tags, discuss media tags, compare SVG and Canvas.**	**K3&K5**
CO-2	**Demonstrate competence in DOM manipulation, arrays, callback functions, form handling with Get/Post method and validation, HTML events, OOP using JavaScript, and explore JavaScript and web development frameworks.**	**K3**
CO-3	**Implement scrolling effects, image manipulation, file handling, XML and JSON parsing, Canvas game development, sprite animations, and keyboard event interactivity.**	**K4**
CO-4	**Develop player movement, background scrolling, jumping, collision detection, UI design, interactions, system-controlled game elements, timers, life/health management, and score tracking in games for an enriched player experience.**	**K6**
CO-5	**Implement Box2D setup, predefined functions, collision detection, and asynchronous updates for dynamic and interactive web-based physics simulations.**	**K5**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	L(1)	M(2)	M(2)	M(2)	L(1)	L(1)	L(1)	M(2)	M(2)
CO2	S(3)	L(1)	M(2)	M(2)	M(2)	L(1)	L(1)	M(2)	M(2)	M(2)
CO3	S(3)	S(3)	M(2)	M(2)	M(2)	L(1)	L(1)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**2.2**	**2.4**	**2.2**	**2.4**	**1.4**	**1.4**	**1.8**	**2.4**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	L(1)	L(1)	L(1)	L(1)
CO2	L(1)	L(1)	L(1)	L(1)	L(1)
CO3	M(2)	M(2)	M(2)	M(2)	M(2)
CO4	M(2)	S(3)	M(2)	M(2)	M(2)
CO5	M(2)	S(3)	M(2)	S(3)	S(3)
W.AV	**1.6**	**2**	**1.6**	**1.8**	**1.8**

S–Strong (3), M-Medium (2), L-Low (1)

IV-Semester - Core					
Core	**Course Code: 82645**	**GAME ENGINE-II -PRACTICAL**	**P**	**Credits: 3**	**Hours: 5**

Objectives

- **Create intricate environments demonstrating advanced level design principles.**
- **Experiment with lighting configurations to evoke varying emotional responses in the game environment.**
- **Build comprehensive character blueprints that include movement, interactions, animations, and sound integration.**
- **Design interactive objects, employing Blueprints for seamless character interaction and providing visual and audio feedback.**
- **Construct functional HUD/UI elements, such as health and ammo indicators, utilizing for player convenience.**

1. **Level Design and Lighting in Unreal Engine:**
   a. Create a small environment with detailed level design.
   b. Experiment with different lighting setups to evoke different moods.
2. **Character Blueprint in Unreal Engine:**
   a. Develop a character blueprint with basic movement and interactions.
   b. Implement animations and sounds for character actions.
3. **Interactive Objects in Unreal Engine:**
   a. Design objects that the character can pick up or interact with.
   b. Use Blueprints to handle object interaction and feedback.
4. **User Interface (UI) Design in Unreal Engine:**
   a. Design and implement a HUD/UI with health, ammo, and other essential indicators.
   b. Use UMG to create functional UI elements.
5. **AI Enemy Behavior in Unreal Engine:**
   a. Create AI enemies with simple behaviors like patrolling or following.
   b. Integrate AI perception to detect the player and react accordingly.
6. **Physics and Destruction in Unreal Engine:**
   Set up physics-based interactions, like breakable objects or moving platforms.
7. **Multiplayer Gameplay in Unreal Engine:**
   a. Establish a multiplayer session with synchronized character movement.
   b. Explore replication techniques for networked gameplay.
8. **Particle Effects in Unreal Engine:**
   Add dynamic particle effects for events like explosions or environmental effects.
9. **Blueprint Scripting Challenges in Unreal Engine:**
   Choose a specific gameplay mechanic (e.g., grappling hook, stealth) and implement it using Blueprints.
10. **Optimization and Packaging in Unreal Engine:**
    a. Optimize a scene for better performance using techniques like culling and LODs.
    b. Package your project for a specific platform and ensure it runs smoothly.

Outcomes

- **Generate a well-detailed environment exhibiting a profound understanding of level design techniques.**
- **Display expertise in employing diverse lighting setups to manipulate ambiance and emotion within the game world.**
- **Develop character blueprints, incorporating movement, interaction, animation, and sound elements for immersive gameplay.**
- **Create interactive objects within the game, utilizing Blueprints for smooth interaction mechanics and delivering player feedback.**
- **Implement a functional HUD/UI with essential indicators, skillfully utilizing UMG to enhance the player's experience.**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	M(2)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
CO5	S(3)	M(2)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
W.AV	**3**	**2.6**	**3**	**2.4**	**3**	**3**	**2.4**	**3**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	S(3)	M(2)
CO2	S(3)	S(3)	M(2)	S(3)	M(2)
CO3	S(3)	S(3)	M(2)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**2**	**3**	**2.6**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">IV – Semester-Allied</th></tr>
<tr><td rowspan="2">Core</td><td rowspan="2">Course Code: 82646</td><td rowspan="2">MOBILE GAME DEVELOPMENT</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>3</td><td>3</td></tr>
<tr><td>Objectives</td><td colspan="5">To study the Java programming concepts and language components
To develop proficiency in inheritance and multithreading concepts
Gain proficiency in mobile platform concepts and app development fundamentals.
Acquire foundational knowledge of game development and graphics libraries.
Develop proficiency in screen transitions, sensor handling, and game physics.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Java: OOPS Concept - Data Abstraction and Encapsulation - Inheritance, Polymorphism, Dynamic binding.
Tokens of Java: Identifiers, Operators, Data Types, Primitives- Control statements - Conditional statements - Arrays - Introduction and Implementation, Types of Arrays - Working with Arrays - Wrapper Class and Type Casting - Math and String Class- Constructors-Static Members, this keyword.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Inheritance: Examples, Types of Inheritance with example- Method Overloading and Overriding- Abstract and Final Classes-Collections and Generic classes-Array List, Vectors-Enumeration.
Threading and MultiThreading: Thread class and Runnable Interface - Multi threadingusingThread class - Multithreading using Runnable Interface-Synchronization-Exception Handling</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Introduction to Mobile Platforms: Role and Benefits of Mobile Platforms - Elements of a Mobile OS-Activity, Service-UI - Views - Introduction to Development Environment - Understanding the IDE Interface - Understanding Build System - Introduction to build tools - Emulator - Running Application with emulators - Working with Views - Working with Layouts - Activity, Service Input- Implementation - Parsing of external files.</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Introduction to Game Development: Basics of Graphics Libraries - Introduction to Game Development Framework - Creating a Project - Importing into IDE - Importing Assets - Game Class- Game Life Cycle - Spritebatch - Sprite - Rendering Text - Camera - Setting up the Camera - Screen Interface - Implementation - Viewports - Texture Atlas - Texture Region - Sprite Animation - Handling Input - Touch Input - Input Processor - Gesture Listener</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Screen Transition and Handling Sensors: Particle Effects - Implementation - Parallax Scrolling- Designing Levels - Event Handling - Programming GamePlay - Basic Interactions – Integrating Physics Engine - Adding Gravity and other Physics Elements - Working with Physics Bodies- Developing a Complete Game.</td></tr>
<tr><td colspan="6">Reference and Text Books:
Andrew Davison, “Killer Game Programming in Java: Java Gaming & Graphics Programming”, O'Reilly Media Inc, 2005.
David Brackeen, Bret Barker, Laurence Vanhelsuwé, “Developing Games in Java”, NewRiders, 2004.
Davison A, “Vision-based User Interface Programming in Java”, Amazon Digital Services, Inc. 2013.
Patrick Hoey, “Mastering LibGDX Game Development”, Packt Publishing Ltd, 2015. 5. Posch M, “Mastering And Engine Game Development”, Packt Publishing Ltd, 2015.</td></tr>
<tr><td colspan="6">Online Resources
https://developer.android.com/games/guides/basics</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Students will acquire the ability to differentiate between 2D and 3D game concepts, design 2D levels and transition to 3D environments, while also becoming skilled in tools like the Profiler and prefabs for proficient 3D game development.**	**K1&K2**
CO-2	**Master scripting techniques for 3D game development, including collision detection, event handling, raycasting, animation control, and 3D physics. Apply optimized frame rates, handle exceptions, utilize list collections, and navigate complex game environments using pathfinding and joint types.**	**K2**
CO-3	**Utilizing camera properties, GUI, cinematic rendering, and global illumination, enhancing their ability to create visually compelling scenes. Implementing advanced rendering techniques, optimizing memory usage, and effectively managing events, resulting in improved performance and immersive 3D game experiences.**	**K4**
CO-4	**Designing functional game UI, implementing HUD for information sharing, managing sound, and comprehending networking concepts for interactive and platform-ready game development.**	**K5**
CO-5	**Proficiency in advanced gameplay programming, including event-driven systems, 2D game mechanics, basic AI mechanics, and pathfinding.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	M(2)	L(1)	M(2)	S(3)	L(1)	L(1)	M(2)	L(1)	S(3)
CO2	S(3)	M(2)	L(1)	M(2)	S(3)	L(1)	L(1)	M(2)	L(1)	S(3)
CO3	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	L(1)	M(2)	M(2)	S(3)
CO4	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
W.AV	**3**	**2.4**	**2**	**2.4**	**3**	**2**	**1.4**	**2.4**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	L(1)	L(1)	M(2)	L(1)	M(2)
CO2	L(1)	L(1)	M(2)	L(1)	M(2)
CO3	M(2)	M(2)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	S(3)	M(2)
W.AV	**2**	**2**	**2**	**2**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

IV-Semester -Allied					
Allied	**Course Code: 82647**	**MOBILE AND WEB GAME DEVELOPMENT - PRACTICAL**	**P**	**Credits:2**	**Hours: 4**

Objectives

- **Acquire hands-on experience in mobile game development through practical projects.**
- **Enhance web game development skills by creating interactive and engaging games.**
- **Develop a strong understanding of fundamental game mechanics and their implementation.**
- **Explore various game design principles and techniques to create enjoyable gaming experiences.**
- **Build a solid foundation in programming and problem-solving by creating diverse types of games.**

Mobile Game Development:

1. Develop a clone of the popular Flappy Bird game where the player controls a character by tapping the screen to make it jump and navigate through obstacles.
2. Build a memory matching game where the player flips over cards to find matching pairs within a grid.
3. Create a sliding puzzle game where the player rearranges pieces of an image to complete it.
4. Develop a classic brick-breaking game where the player controls a paddle to bounce a ball and break bricks.
5. Design an endless runner game where the player's character automatically moves forward, and the player must swipe to avoid obstacles and collect items.

Web Game Development:

1. Develop a simple quiz application
2. Create a canvas and demonstrate parallax scrolling
3. Develop a simple game and demonstrate player movement and collision detection.
4. Define different types of collision detection methods and demonstrate them using html5 canvas.
5. Create a Simple Click to Shoot game.

Outcomes

- **Attain proficiency in developing mobile and web games, showcasing practical skills in game design and programming.**
- **Exhibit creativity by designing diverse game concepts, fostering imaginative game mechanics and experiences.**
- **Strengthen problem-solving abilities through tackling challenges in game development, fostering critical thinking and analytical skills.**
- **Create engaging and interactive game environments, demonstrating an understanding of user experience and interface design.**
- **Generate a comprehensive portfolio of varied game projects, illustrating competence and versatility in game development to potential employers or educational pursuits.**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)
CO2	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)
CO4	M(2)	M(2)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	S(3)
CO5	M(2)	M(2)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.6**	**2.6**	**2.6**	**1.6**	**3**	**1.6**	**2**	**2**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	M(2)	S(3)
CO2	S(3)	S(3)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	M(2)	M(2)	S(3)
CO4	M(2)	M(2)	M(2)	S(3)	S(3)
CO5	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.6**	**2.6**	**2**	**2.4**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

V – Semester-Core					
Core	**Course Code: 82651**	**ARTIFICIAL INTELLIGENCE FOR GAMES**	**T**	**Credits: 4**	**Hours: 4**
Objectives	● **To learn the fundamental concepts of problem-solving in artificial intelligence, including problem spaces, search techniques, and production system characteristics.** ● **To educate the intricacies of implementing diverse AI strategies in game development, encompassing roaming, patterned behavior, chasing, evading, backtracking, and strategic decision-making.** ● **To acquire a solid grasp of various advanced AI methods used in games, spanning pathfinding, rule-based systems, fuzzy logic, genetic algorithms, and neural networks.** ● **Gain proficiency in diverse knowledge representation methods, including production and frame-based systems, fuzzy reasoning, Bayesian networks, and advanced plan generation techniques.** ● **Comprehend expert systems' architecture, knowledge acquisition, meta knowledge, and the integration of AI techniques for intelligent agents in games.**				
UNIT-I	**Introduction to Artificial Intelligence:** The AI Problems - AI Technique - The Level of the Model - Criteria for success - Problems, Problem Spaces and Search : Defining the problem as a StateSpaceSearch - Production System Characteristics - Issues in the Design of Search Programs.				
UNIT-II	**Game Artificial Intelligence:** Types of AI - Roaming AI - Patterned Roaming , Chasing Evading- Backtracking - Creating Grid Based Canvas - Behavioral AI - State change - Strategically AI - HowtoCreate Strategically AI in Games - The importance of good Game AI. The differences between Game AI and AI and their relative advantages and disadvantages				
UNIT-III	**Deterministic and Non deterministic:** consideration for Game AI & AI systems Pathfinding - A* and its derivatives - Flocking and Steering AI - Rule Based Systems - Finite State Machines - Patterning and Way point - Chasing and Evading - Fuzzy Logic and Fuzzy State Machines - Genetic Algorithms- Artificial Neural Networks - Rule based AI				
UNIT-IV	**Knowledge representation:** Production based system - Frame based system - Inference – Backward chaining - Forward chaining - Rule value approach - Fuzzy reasoning – Certainty factors - Bayesian Theory - Bayesian Network-Dempster – Shafer theory - Basic plan generation systems – Strips- Advanced plan generation systems – K strips				
UNIT-V	**Expert systems:** Architecture of expert systems - Roles of expert systems – Knowledge Acquisition – Meta knowledge - Heuristics. - Applied AI : Combining AI techniques to produce Intelligent Agents - Strategic AI : The Future for AI in games				

Reference and Text Books:

- Copeland J, “Artificial intelligence: A philosophical introduction”, John Wiley & Sons, 2015.
- David L. Poole, Alan K. Mackworth, “Artificial Intelligence: Foundations of Computational Agents”, Cambridge University Press, 2010.
- Elaine Rich, Kevin Knight, Shivashankar B Nair, “Artificial Intelligence”, Tata McGraw-Hill publishing, 2009.
- Rich, “Artificial Intelligence 3E (Sie)”, Tata McGraw-Hill Education, 2004.
- Russell SJ, Norvig P, “Artificial intelligence: a modern approach”, Pearson EducationLimited, 2016.

Online Resources

- Artificial Intelligence

Course Outcomes		Knowledge level
CO-1	**Deconstruct problems into state space models, employ diverse search methods, and construct rudimentary production systems, demonstrating an awareness of search program design challenges.**	**K3**
CO-2	**To apply AI techniques in games, create behavioral patterns, and recognize the significance of effective Game AI, while understanding differences and trade-offs between Game AI and general AI.**	**K4**
CO-3	**To implement a range of advanced AI strategies, enhancing games through efficient pathfinding, complex behaviors, adaptive decision-making, evolutionary optimization, and learning-based actions.**	**K5**
CO-4	**To apply these techniques to represent knowledge, utilize reasoning mechanisms, and design effective plans in AI systems.**	**K5**
CO-5	**To create expert systems, gather knowledge, use meta knowledge, combine AI techniques for intelligent agents, and recognize the significance of strategic AI for the gaming future.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	L(1)	M(2)	M(2)	M(2)	M(2)	M(2)
CO2	M(2)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	M(2)	M(2)	M(2)	L(1)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**2.6**	**2.6**	**2.4**	**1.8**	**2**	**2**	**2**	**2.6**	**2.4**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	L(1)	M(2)	M(2)	M(2)
CO3	S(3)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**2.2**	**2**	**2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

V – Semester-Core					
Core	**Course Code: 82652**	**GAME PROGRAMMING PATTERNS**	**T**	**Credits: 4**	**Hours: 4**
Objectives	● **To understand the history, types, selection, and application of design patterns, and revisit core OOP concepts.** ● **To learn various creational, structural, and behavioral design patterns, and comprehend their uses and implementations.** ● **Apply design patterns to game development, specifically focusing on builder, factory method, prototype, singleton, and various other patterns.** ● **To educate sequencing and decoupling patterns, including double buffer, game loop, component-based design, and various optimization techniques.** ● **Apply design patterns to specific game components like brick systems, power-ups, paddle mechanics, enemy behaviors, and collision control.**				
UNIT-I	**Introduction to Design Patterns:** Design Pattern History - Types of Design Patterns - Problem Solving using Design Patterns - Selecting Design Pattern - Using Design Pattern - Revisiting OOPS-Abstraction - Inheritance - Polymorphism - Encapsulation				
UNIT-II	**Creational Design Patterns:** Abstract Factory - Builder - Factory Method - Object Pool - Prototype-Singleton - Structural Design Pattern: Adapter - Bridge - Composite - Decorator - Facade - Flyweight- Private Class Data - Proxy Behavioral Design Pattern: Chain of Responsibility - Command- Interpreter - Iterator - Mediator - Memento - Null Object - Observer - State - Strategy - Template method - Visitor				
UNIT-III	**Design Patterns in Games with Examples:** Builder - Factory Method - Prototype – Singleton- Adapter - Composite - Facade - Flyweight - Proxy Chain of Responsibility - Command - Mediator- Observer - State - Strategy - Template Method				
UNIT-IV	**Sequencing Patterns:** Double Buffer - Game Loop - Update Method - Behavioural Patterns- Bytecode - Subclass Sandbox - Type Object - Decoupling Patterns - Component – Event Queue - Service Locator - Optimization Process - Data Locality - Dirty Flag - Object Pool - Spatial Partition- Entity Component System				
UNIT-V	**Design Patterns in Breakout**: Bricks System - Power Up Management - Simple Paddle - Paddlewith Special Power - Managing Game Mechanics - Collision Control – Space Invaders: - EnemySystem - Upgrade system - Weapon system - Power Up Management - Enemy Movement Pattern- Identifying the Common Factors in Breakout and Space Invaders				

Text Book:

- "Game Programming Patterns", Robert Nystrom, Genever Benning, 2014

References:

- Ahnert, K., & Mulansky, M "Odeint–solving ordinary differential equations in C++", InAIPConference Proceedings, AIP, 2011.
- Andrei Alexandrescu, "Modern C++ Design: Generic Programming and DesignPatternsApplied" illustrated, reprint, Addison-Wesley Professional, 2011.
- Bangerth, W, "Using Modern Features of C++ for Adaptive Finite Element Methods", Dimension Independent Programming in dealwII, 2000.
- Gamma, E, "Design patterns: elements of reusable object-oriented software", PearsonEducatio India, 1995.
- M. S. Joshi, "C++ Design Patterns and Derivatives Pricing", Cambridge University Press, 2011.

Online Resources

- https://gameprogrammingpatterns.com/

Course Outcomes		Knowledge level
CO-1	**Able to identify appropriate design patterns for problem-solving, apply them effectively, and demonstrate a strong grasp of OOP principles including abstraction, inheritance, polymorphism, and encapsulation.**	**K3**
CO-2	**Employ creational and structural design patterns such as abstract factory, builder, adapter, composite, decorator, and more, enabling them to create well-structured and modular software designs.**	**K4**
CO-3	**Integrate design patterns into game development, using examples of builder, factory method, prototype, and singleton patterns to enhance the architecture of games.**	**K3**
CO-4	**Implement sequencing patterns like game loops, apply decoupling techniques to improve code flexibility, and use optimization methods to enhance game performance**.	**K5**
CO-5	**Implement design patterns within game development, specifically focusing on applying patterns to various components and mechanics in breakout-style and space invaders-style games.**	**K5**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	L(1)	M(2)	M(2)	M(2)	M(2)	M(2)
CO2	S(3)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**2.6**	**2.4**	**2.2**	**2**	**2**	**2**	**2.6**	**2.4**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	L(1)	M(2)	M(2)	M(2)
CO3	S(3)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**2.2**	**2**	**2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">V – Semester-Elective I</th></tr>
<tr><td rowspan="2">Elective</td><td rowspan="2">Course Code: 82653A</td><td rowspan="2">DISCIPLINE SPECIFIC ELECTIVE-I
A. SOUND DESIGN FOR GAMES</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand the significance of sound design in gaming and its impact on player experience.
● Develop skills in recording and editing sound to create polished audio assets.
● Learn how to create dynamic and interactive audio experiences using scripting.
● Gain proficiency in integrating spatial audio techniques for realistic in-game soundscapes.
● Understand how sound can convey emotions and contribute to storytelling in games.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Sound Design for Games: Role and importance of sound in game development - Elements: music, effects, ambient sounds, voiceovers - Sound's psychological impact on player immersion - Introduction to audio tools in game development - Hands-on: Setup audio environment, basic integration.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Sound Recording and Editing:Basics of sound recording: microphones, techniques - Introduction to digital audio workstations (DAWs)- Cleaning, editing: noise reduction, equalization - Layering, mixing for depth and richness - Hands-on: Record, edit sounds for a simple game scene.</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Interactive Audio and Implementation:Create adaptive soundscapes based on player actions - Integrate audio events for in-game interactions - Introduction to audio scripting languages for interaction - Dynamic music systems that react to gameplay - Hands-on: Implement interactive audio with scripting.</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Spatial Audio and 3D Sound:Understand spatial audio: binaural, 3D positioning - Simulate distance, direction, environmental effects - Use audio middleware for spatial audio - Design soundscapes to enhance immersion - Hands-on: Integrate spatial audio into game levels.</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Emotional Impact and Storytelling through Sound:Explore sound's emotional impact in games - Convey narrative, atmosphere, emotions through audio - Collaborate with other disciplines for storytelling - Case studies of games with exceptional sound design.</td></tr>
<tr><td colspan="6">Reference and Text Books:
● "The Essential Guide to Game Audio: The Theory and Practice of Sound for Games" by Steve Horowitz and Scott Looney- UNIT-I
● "The Sound Effects Bible: How to Create and Record Hollywood Style Sound Effects" by Ric Viers- UNIT-II
● "Game Audio Programming: Principles and Practices" by James Boer - UNIT-III
● "3D Audio Programming: Theories and Practices" by Ravish Mehra and Jyoti Narang - UNIT-IV
● "Music, Sound and Story in Film and Media" by Kathryn Kalinak - UNIT-V</td></tr>
<tr><td colspan="6">Online Resources
● https://www.gamedesigning.org/learn/video-game-sound/</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Able to articulate the importance of sound in games and describe its role in enhancing player immersion.**	**K2**
CO-2	**Able to record and edit sound using digital audio workstations (DAWs) to produce high-quality audio assets for games.**	**K3**
CO-3	**Implement interactive audio elements in games using scripting languages to enhance gameplay immersion.**	**K3**
CO-4	**Integrate spatial audio into game environments, creating a sense of depth and directionality in sound.**	**K4**
CO-5	**Design soundscapes that evoke emotions and enhance narrative elements, showcasing the storytelling potential of sound.**	**K4**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	L(1)	M(2)	M(2)	M(2)	M(2)	M(2)
CO2	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**2.6**	**2.4**	**2.2**	**2.4**	**2.4**	**2**	**2.6**	**2.4**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	L(1)	M(2)	M(2)	M(2)
CO3	S(3)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**2.2**	**2**	**2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">V – Semester-Elective I</th></tr>
<tr><td rowspan="2">Elective</td><td rowspan="2">Course Code: 82653B</td><td rowspan="2">DISCIPLINE SPECIFIC ELECTIVE-I
B. SHADER PROGRAMMING</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand the role of shaders in graphics programming, shading languages, and different types of shaders.
● Explore uniforms, built-in variables, functions, and the process of creating, compiling, and running shader programs.
● To educate lighting principles, surface normals, different types of lights, and effects like cartoon shading and fog.
● Familiarize texture mapping techniques, different types of textures, and image-based lighting.
● Understand image manipulation operations, filters, and various shader effects.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Shaders: Introduction - Applications - Shading Languages - GLSL - Introduction - Types of Shaders- Vertex Shaders - Geometry Shaders - Fragment Shaders - Tessellation Shaders - Primitive Shaders- Vertex Data - Vertex Attributes - Vertex Arrays - Fragment Data.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Uniforms: Built in variables - Build in Functions - Creating Shader Program - Running the Shader-Shader Compilation & Linking - Algorithmic Drawing - Matrices - Shapes - Colors - Transformations- Translations - Animation - Depth Buffering</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Lighting: Lighting Principles - Surface Normals - Light Normals - Light Material - MultiplePositional Lights - Directional Light - Spot Light - Cartoon Shading Effect - Fog Effects</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Textures: Image Operations - Texture Mapping - Texture Objects - Multiple Textures - Alpha Maps- Normal Maps - Cube Maps - Image based Lighting - Mipmap - Projected Texture</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Image Operations: Filters - Edge Detection Filter - Gaussian Blur Effect - Bloom Effect - GammaCorrections - Anti aliasing - Mesh Shader - Smoothing - Silhouette Effects - Reflection Map- BumpMap</td></tr>
<tr><td colspan="6">Reference and Text Books:
● "OpenGL Shading Language" by Randi J. Rost -UNIT-I
● "OpenGL SuperBible: Comprehensive Tutorial and Reference" by Graham Sellers, Richard S. Wright Jr., and Nicholas Haemel- UNIT-II
● "Real-Time Rendering, Fourth Edition" by Tomas Akenine-Möller, Eric Haines, Naty Hoffman- UNIT-III
● "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5" by Dave Shreiner, Graham Sellers, John M. Kessenich, Bill M. Licea-Kane - UNIT-IV
● "OpenGL Insights" edited by Patrick Cozzi and Christophe Riccio - UNIT-V</td></tr>
<tr><td colspan="6">Online Resources
● Shader Programming</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Able to differentiate between vertex, geometry, fragment, tessellation, and primitive shaders, and grasp the concept of vertex attributes and arrays for rendering graphics.**	**K2**
CO-2	**To apply uniforms, use built-in variables and functions, create and run shader programs, and understand how matrices, shapes, colors, transformations, translations, and animations are applied in shader-based rendering.**	**K3**
CO-3	**To apply lighting concepts, calculate normals, implement multiple lights including directional and spot lights, and create special effects like cartoon shading and fog in graphics scenes.**	**K3**
CO-4	**Able to use textures, implement techniques like texture mapping, alpha maps, normal maps, and cube maps, and understand the concept of image-based lighting and mipmap generation.**	**K3**
CO-5	**To apply filters, create shader effects, and understand advanced graphics techniques.**	**K5**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)
CO2	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**2.2**	**3**	**2.4**	**2**	**2.6**	**2.4**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	S(3)	M(2)	M(2)	M(2)
CO3	S(3)	L(1)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	S(3)
CO5	S(3)	S(3)	M(2)	S(3)	S(3)
W.AV	**3**	**2.6**	**2**	**2.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">V – Semester-Elective I</th></tr>
<tr><td rowspan="2">Elective</td><td rowspan="2">Course Code: 82653C</td><td rowspan="2">DISCIPLINE SPECIFIC ELECTIVE-I
C. GAME ENGINE CUSTOMIZATION</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand the role and components of game engines in game development.
● Develop skills in customizing graphics rendering within game engines.
● Gain proficiency in customizing physics simulations for interactive gameplay.
● Learn how to integrate and customize audio and animations for immersive experiences.
● Acquire scripting skills to implement and enhance gameplay mechanics.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Game Engines and Customization: Game engines' role in development - Components: graphics, physics, audio, scripting - Intro to programming languages (C++, C#) - Explore Unity, Unreal Engine - Hands-on: Set up environment, simple project.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Graphics and Rendering Customization: Rendering pipeline and APIs (OpenGL, DirectX) - Shader basics: vertex, fragment, geometry - Custom rendering: shadows, post-processing, particles - GPU programming and parallel computing - Hands-on: Write custom shaders, visuals.</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Physics and Simulation Integration: Physics engines and integration - Rigid bodies, collision detection, simulations - Customize physics for gameplay - Advanced: soft bodies, cloth, vehicles - Hands-on: Implement custom physics, simulations.</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Audio and Animation Customization: Audio systems: spatial, effects, mixing - Dynamic soundscapes, interactive audio - Animation: keyframing, skeletal, blend trees - Custom animations for characters, objects - Hands-on: Add custom audio, animations.</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Scripting and Gameplay Mechanics:Scripting in engines (C#Lua) for gameplay - Design, implement mechanics using scripts - Customize UI, menus, user input - Modding support for community customization - Hands-on: Develop mechanics, scripted events.
Game Engine Extension: Implement custom rendering, physics, audio, mechanics - Emphasis on creativity, optimization - Regular milestones, documentation, presentation.</td></tr>
<tr><td colspan="6">Reference and Text Books:
● “Game Engine Architecture" by Jason Gregory- UNIT-I
● "Real-Time Rendering, Fourth Edition" by Tomas Akenine-Möller, Eric Haines, Naty Hoffman - UNIT-II
● "Physics for Game Developers" by David M. Bourg - UNIT-III
● "The Game Audio Tutorial: A Practical Guide to Sound and Music for Interactive Games" by Richard Stevens and Dave Raybould - UNIT-IV
● "The Animator's Survival Kit" by Richard Williams - UNIT-IV
● "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking -UNIT-V
● "Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#" by Jeremy Gibson Bond - UNIT-V</td></tr>
<tr><td colspan="6">Online Resources
● GAME ENGINE CUSTOMIZATION</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Able to explain the purpose of game engines and identify their key components.**	**K2**
CO-2	**To create and integrate custom shaders to achieve specific visual effects in games.**	**K5**
CO-3	**To modify physics behaviors to create dynamic interactions and engaging game mechanics.**	**K3**
CO-4	**To integrate interactive audio elements and apply customized animations to enhance game aesthetics.**	**K3**
CO-5	**Develop functional scripts to create dynamic gameplay systems and interactions.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)
CO2	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**2.2**	**3**	**2.4**	**2**	**2.6**	**2.4**	**2.8**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	S(3)	M(2)	L(1)	M(2)
CO3	S(3)	L(1)	M(2)	L(1)	M(2)
CO4	S(3)	S(3)	M(2)	L(1)	S(3)
CO5	S(3)	S(3)	M(2)	M(2)	S(3)
W.AV	**3**	**2.6**	**2**	**1.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">V – Semester-Elective II</th></tr>
<tr><td rowspan="2">Elective</td><td rowspan="2">Course Code: 82654A</td><td rowspan="2">DISCIPLINE SPECIFIC ELECTIVE-II
A. GAME MARKET ANALYSIS AND MONETIZATION</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand the importance of market analysis and various monetization models in the game industry.
● Develop skills in conducting effective market research and analyzing player behavior.
● Gain proficiency in designing and selecting appropriate monetization strategies.
● Understand the mechanics of in-game advertising and in-app purchases.
● Learn how to measure game success and optimize monetization strategies.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Game Market Analysis and Monetization:
Overview of the game industry landscape: platforms, genres, trends - Importance of market analysis and monetization strategies - Understanding target audience, demographics, and player behavior - Introduction to different monetization models: freemium, premium, ads, in-app purchases - Case studies of successful games with different monetization approaches.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Market Research and Player Insights:
Conducting market research: data collection, surveys, analytics - Analyzing player behavior and preferences using player data - Defining player personas and understanding player motivations - Identifying trends, demands, and gaps in the market - Hands-on: Analyzing player data and identifying potential opportunities.</td></tr>
<tr><td>UNIT-III</td><td colspan="5">Monetization Strategies and Business Models:
In-depth exploration of various monetization models -Pros and cons of each model in different game genres -Creating a sustainable revenue stream: pricing strategies and value propositions -Developing a business plan: budgeting, forecasting, and revenue projections -Hands-on: Designing a monetization strategy for a hypothetical game.</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Advertising and In-Game Purchases:
Understanding the mechanics of in-game ads and their impact on player experience - Integrating ads effectively: rewarded videos, interstitials, banners -Designing in-app purchases: virtual goods, cosmetic items, power-ups - Ethical considerations in monetization and player engagement - Hands-on: Implementing ads and in-game purchases in a sample game.</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Metrics, Analytics, and Optimization:
Key performance indicators (KPIs) for measuring success - Using analytics tools to monitor player engagement and revenue - A/B testing and optimizing monetization strategies - Responding to player feedback and adapting monetization approaches - Hands-on: Analyzing metrics and optimizing monetization in a live game.</td></tr>
<tr><td colspan="6">Reference and Text Books:
Text Book:
● "The Business of Game Design: A Guide to Creating & Marketing Games" by Brian Robbins and Larry C. Medsker
References:
● "The Art of Game Design: A Book of Lenses" by Jesse Schell
● "Game Analytics: Maximizing the Value of Player Data" by Magy Seif El-Nasr, Anders Drachen, Alessandro Canossa
● "Monetization in Video Games" by David Wesley
● "Free-to-Play: Making Money From Games You Give Away" by Will Luton
● "Game Data Analysis – Tools and Methods" by Sander Dieleman, Benjamin Schrauwen</td></tr>
</table>

Online Resources • GAME MARKET ANALYSIS AND MONETIZATION		
Course Outcomes		**Knowledge level**
CO-1	**Able to explain the significance of market analysis and describe different monetization approaches used in games.**	**K3**
CO-2	**To gather player data, analyze trends, and define player personas to inform game development decisions.**	**K4**
CO-3	**Develop the ability to create a sustainable revenue stream by choosing suitable monetization models and pricing strategies.**	**K2**
CO-4	**To integrate ads and design in-game purchases while considering player experience and ethical considerations.**	**K5**
CO-5	**Able to interpret key performance indicators (KPIs), use analytics tools, and optimize monetization approaches based on data analysis.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M(2)	M(2)	M(2)	L(1)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)
CO2	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)
CO4	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	M(2)	S(3)	M(2)	S(3)
CO5	M(2)	M(2)	M(2)	S(3)	L(1)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**2**	**2**	**2**	**2.4**	**1.6**	**2**	**2**	**2.6**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	M(2)	M(2)	M(2)	L(1)	L(1)
CO2	M(2)	M(2)	M(2)	L(1)	M(2)
CO3	M(2)	M(2)	M(2)	L(1)	M(2)
CO4	M(2)	M(2)	M(2)	L(1)	S(3)
CO5	M(2)	M(2)	M(2)	M(2)	S(3)
W.AV	**2**	**2**	**2**	**1.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

V – Semester-Elective II					
Elective	**Course Code: 82654B**	**DISCIPLINE SPECIFIC ELECTIVE-II B. GAME ENGINE ARCHITECTURE**	**T**	**Credits: 4**	**Hours: 4**
Objectives	● **Understand game engine differences across genres, runtime engine architecture, tools, asset pipelines, version control, and memory management techniques.** ● **To learn foundational 3D mathematics concepts, engine subsystems, memory management, and the game loop architecture.** ● **Explore types of input and output, HUD systems, debugging facilities, rendering engine architecture, and lighting techniques.** ● **To educate the architecture of game elements, animation systems, interaction mechanics, level handling, and physics simulation.** ● **Understand data-driven engines, gameplay foundation systems, object model architecture, scripting, and high-level game flow.**				
UNIT-I	**Introduction to Game Engine**: Engine Differences across genres - Runtime Engine Architecture- Tools and asset pipeline - Version Control - Profiling Tools- Memory Leaks and CorruptionDetection - Data, Code and Memory in C/C++ - Error handling - Exception Handling in C/C++				
UNIT-II	**3D Maths for Games:** Points and Vectors - Matrices - Quaternions - Engine Support Subsystem- Start-up and Shut-Down - Memory Management - Containers - Engine Configuration - ResourceManagement - Game Loop - Rendering Loop - Architectural Style of Game Loop - Dealing withTime- Multiprocessor Game Loop				
UNIT-III	**Human Interface Devices:** Types of Input - Types of Outputs - Game Engine HUD System- Loggingand Tracing - Debug Facilities - In-Game Menu - In-Game Console - In-Game Profiling - RenderingEngine - Depth-Buffered Triangle Rasterization - Rendering Pipeline - Lighting andGlobal Illumination				
UNIT-IV	**Game elements:** Game Objects, User interaction, Menus, HUD, Background elements - AnimationSystem Architecture - Action State Machines - Animation Controllers - Level handling. Gameplay Scripting - Physics System - Rigid Body				
UNIT-V	**Gameplay Systems:** Data Driven Game Engines - Game World Editor - Runtime GameplayFoundation Systems - Components of Gameplay Foundation Systems - Runtime Object Model Architecture - Loading and Streaming Game Worlds - Updating Game Objects - Events andMessagePassing - Scripting - High-Level Game Flow				

Reference and Text Books:

- Allen Sherrod, "Ultimate 3D Game Engine Design & Architecture", Cengage Learning, 2009.
- David H. Eberly, "3D Game Engine Architecture: Engineering Real-Time Applications withWild Magic", illustrated, Taylor & Francis, 2005.
- James M. Van Verth, "Essential Mathematics for Games and Interactive Applications", ThirdEdition. 3 Edition. A K Peters/CRC Press, 2015.
- Jason Gregory, "Game Engine Architecture", Second Edition, 2, illustrated, CRC Press, 2017.
- Robert Nystrom, "Game Programming Patterns", Genever Benning, 2014.

Online Resources

- Game Engine Architecture

Course Outcomes		**Knowledge level**
CO-1	**Able to differentiate game engine features across genres, comprehend runtime architecture, utilize tools for asset management, implement version control, and detect and manage memory issues in game development.**	**K2**

CO-2	**To use 3D mathematics in game contexts, manage memory and resources, understand engine subsystems, and implement efficient game and rendering loops.**	**K3**
CO-3	**Design HUD systems, utilize debugging tools, comprehend rendering pipeline and lighting principles, and implement rendering techniques like depth-buffered triangle rasterization.**	**K2**
CO-4	**Design game elements, implement animation systems, manage user interactions and level handling, and simulate physics using rigid body dynamics.**	**K5**
CO-5	**Develop data-driven game engines, design gameplay foundation systems, manage object models, implement scripting, and control high-level game flow, showcasing their mastery in game development.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	M(2)	M(2)	M(2)	S(3)	M(2)	S(3)
CO2	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	L(1)	S(3)	M(2)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**2.4**	**2.4**	**2**	**1.8**	**2.8**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	L(1)	L(1)
CO2	S(3)	S(3)	S(3)	L(1)	M(2)
CO3	S(3)	S(3)	S(3)	L(1)	M(2)
CO4	S(3)	S(3)	S(3)	L(1)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**1.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

V – Semester-Elective II					
Elective	**Course Code: 82654C**	**DISCIPLINE SPECIFIC ELECTIVE-II C. EMERGING TRENDS IN GAME DEVELOPMENT**	**T**	**Credits: 4**	**Hours: 4**
Objectives	● **Understand VR goals, definitions, hardware, sensation and perception, geometric modeling, and transformation concepts.** ● **To learn axis-angle representations, quaternions, homogeneous transformations, and viewing transforms.** ● **Explore light interpretation, refraction, depth perception, motion perception, orientation tracking, and correction techniques.** ● **To educate AR classification, image acquisition, feature extraction, matching, and verification techniques.** ● **Understand IoT concepts, sensing, actuation, networking, communication protocols, and data handling.**				
UNIT-I	**Introduction to VR:** Goals and VR Definitions - Birds-eye view - Birds-eye view Software - Bird's-eyeviewHardware - Birds-eye view Sensation and Perception - Geometric modeling - Transformation- Matrices and rotation - Pitch Yaw and Roll				
UNIT-II	**Axis-Angle Representations:** Quaternions - Converting and Multiplying Rotations - HomogeneousTransformations - Viewing Transforms - Eye Transforms - Canonical View Transform- ViewportTransformation				
UNIT-III	**Three interpretations of light:** Refraction - Lens aberrations - Light intensity - Eye movement - Depth perception - Motion perception - Orientation tracking - Tilt Drift Correction - YawDriftCorrection - Tracking with Camera - Perspective n-point Problem - Filtering				
UNIT-IV	**Introduction to AR:** Classification based on Sensor, Vision and Hybrid Tracking - Image Acquisition- Feature extraction - Feature Matching - Geometric Verification - Associated Information Retrieval - Feature Extraction Techniques - SIFT - SURF				
UNIT-V	**Introduction to IoT:** Sensing - Actuation - Networking - Communication Protocols - SensorNetworks - Machine-to-Machine Communication - BCI - Neuro Gaming - Data HandlingandAnalytics - Sensor Cloud - Smart Grid				

Reference and Text Books:

- K. S. Hale and K. M. Stanney, “Handbook on Virtual Environments", 2nd edition,CRC Press, 2015.
- Mayer R, Mayer RE, “The Cambridge handbook of multimedia learning”, Cambridge university press; 2005.
- Sadowski W, Stanney K, “Presence in virtual environments”, 2002.
- Weinersmith, K. and Weiner, Z. “Soonish: Ten Emerging Technologies That'll Improve And/orRuin Everything”, 2017.
- Weiss J, Nolan J, Hunsinger J, Trifonas P, “The international handbook of virtual learning environments”, Dordrecht, Netherlands Springer, 2006.

Online Resources

- EMERGING TRENDS Virtual Reality Virtual reality

Course Outcomes		**Knowledge level**
CO-1	**Able to differentiate VR components, describe sensation and perception in VR, and apply geometric transformations and matrices for creating immersive experiences.**	**K2**
CO-2	**To use axis-angle and quaternion representations for rotations, perform transformations, and apply viewing transforms for VR scenes.**	**K3**

CO-3	**Able understand light interactions, depth perception mechanisms, motion perception cues, and implement orientation tracking while considering correction methods for VR experiences.**	**K2**
CO-4	**To classify AR tracking methods, extract features from images, match and verify features, and retrieve associated information in augmented reality contexts.**	**K5**
CO-5	**Explore IoT components, design sensing systems, analyze protocols, handle IoT data, and grasp IoT's impact on networks and data.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	L(1)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)
CO2	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	L(1)	S(3)	M(2)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**2.4**	**3**	**2**	**1.8**	**2.8**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	L(1)	L(1)
CO2	S(3)	S(3)	S(3)	L(1)	M(2)
CO3	S(3)	S(3)	S(3)	L(1)	M(2)
CO4	S(3)	S(3)	S(3)	L(1)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**3**	**3**	**3**	**1.2**	**2.2**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><td colspan="6">V-Semester-Elective-
III</td></tr>
<tr><td>Elective</td><td>Course Code: 82655A</td><td>DISCIPLINE SPECIFIC ELECTIVE-III
A. CINEMATICS IN GAMES-PRACTICAL</td><td>P</td><td>Credits: 4</td><td>Hours: 4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Understand the foundational elements of cinematic design in games, including camera movements, animations, dialogue, and environmental cues.
➢ Apply interactive narrative techniques by developing dialogue systems that allow players to make choices influencing the outcomes of cinematic sequences.
➢ Demonstrate the ability to design and implement dynamic camera systems that automatically follow characters during gameplay to enhance storytelling and immersion.
➢ Create game environments enriched with visual cues and elements that communicate narrative context, creating a more immersive and engaging storytelling experience.
➢ Develop the skills to craft time-lapse cinematics depicting the passage of time or implementing triggered cinematics that respond to specific in-game conditions, effectively enhancing narrative and player engagement.</td></tr>
<tr><td colspan="6">1. Cinematic Cutscene: Create a cinematic cutscene that introduces a game's story or characters using camera movements, animations, and dialogue.
2. Narrative Puzzles: Create puzzle-based cinematics where players must solve challenges in the environment to advance the cinematic story.
3. Dynamic Camera Sequences: Design a dynamic camera system that follows characters during gameplay, enhancing immersion and storytelling.
4. Environmental Storytelling: Construct an environment with visual cues and elements that convey a narrative without relying on direct dialogue or exposition.
5. Time-Lapse Sequences: Craft time-lapse cinematics that showcase the passage of time, such as day-night cycles or the growth of a structure.
6. Endings and Epilogues: Design impactful cinematics that provide closure to the game's story, offering players a satisfying conclusion.</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ To demonstrate proficiency in designing and creating cinematic cutscenes, incorporating camera movements, animations, and dialogue to effectively convey the game's story and characters.
➢ To develop the ability to design and implement interactive dialogue systems that allow players to make choices influencing the outcomes of cinematic sequences, enhancing player engagement and immersion.
➢ Gain the skill to design and apply dynamic camera systems that automatically follow characters during gameplay, contributing to a more immersive and visually engaging player experience.
➢ To construct game environments with visual cues and elements that convey narrative context without relying on direct exposition, contributing to a richer and more immersive storytelling experience.
➢ Develop the capability to craft time-lapse cinematics showcasing the passage of time or triggering scripted events in response to specific in-game conditions, enhancing storytelling and player engagement</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	S(3)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**3**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2.6**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	S(3)
W.AV	**3**	**3**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><td colspan="6">V-Semester-Elective-
III</td></tr>
<tr><td>Elective</td><td>Course Code: 82655B</td><td>DISCIPLINE SPECIFIC ELECTIVE-III
B. LEVEL DESIGN AND ENVIRONMENTAL CREATION-PRACTICAL</td><td>P</td><td>Credits: 4</td><td>Hours: 4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Gain a solid understanding of fundamental principles in level design, including spatial composition, flow, balance, and player navigation within game environments.
➢ Develop skills in using design elements such as lighting, color, textures, and props to create visually appealing and immersive game environments.
➢ Acquire technical proficiency in using game engine tools for terrain sculpting, asset placement, lighting setup, and other aspects of environmental creation.
➢ Learn to incorporate gameplay elements, such as puzzles, obstacles, or interactive objects, to engage players and create dynamic experiences within designed environments.
➢ Understand the concept of environmental storytelling and apply it to design levels that convey narrative elements, evoke emotions, and engage players without relying on explicit exposition.</td></tr>
<tr><td colspan="6">1. Create a Basic Indoor Environment: Design a simple indoor environment with walls, floors, and basic props to learn the fundamentals of level layout and spatial organization.
2. Outdoor Scene with Terrain: Build an outdoor scene using terrain sculpting tools to create hills, valleys, and landscape features, and populate it with natural elements like trees and rocks.
3. Interior Lighting and Atmosphere: Focus on lighting and ambiance by setting up dynamic lighting, adding light sources, and creating a specific mood for an indoor environment.
4. Environmental Storytelling: Craft a level that tells a story without using text or dialogue, relying solely on visual cues and environmental details to convey narrative elements.
5. Puzzle or Obstacle Course: Design a level featuring puzzles or obstacles that challenge players' problem-solving skills and spatial awareness.</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Demonstrate proficiency in designing game environments, showcasing an understanding of spatial layout, flow, and player navigation to create engaging and immersive spaces.
➢ To create visually appealing and artistically coherent game environments, effectively utilizing lighting, textures, props, and other design elements to enhance the overall aesthetic.
➢ Exhibit technical mastery in using game engine tools and features to sculpt terrains, place assets, set up lighting, and implement environmental details that contribute to the desired gameplay experience.
➢ Integrate interactive gameplay elements within designed environments, demonstrating their ability to create engaging challenges, puzzles, or obstacles that enhance player engagement.
➢ Apply the concept of environmental storytelling to their projects, skillfully crafting game environments that convey narrative elements and emotions without overt exposition, thus enhancing player immersion and understanding.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	S(3)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	S(3)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**3**	**3**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2.6**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	S(3)	S(3)	S(3)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	S(3)
W.AV	**3**	**3**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><td colspan="6">V-Semester-Elective-
III</td></tr>
<tr><td>Elective</td><td>Course Code: 82655C</td><td>DISCIPLINE SPECIFIC ELECTIVE-III
C. GAME TESTING AND PROFILING - PRACTICAL</td><td>P</td><td>Credits: 4</td><td>Hours: 4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Develop a solid understanding of key performance metrics such as frame rate, input lag, memory usage, and loading times, and their impact on gameplay experience.
➢ Learn various profiling techniques to identify performance bottlenecks, memory leaks, and inefficient code segments within game projects.
➢ Acquire knowledge of different testing strategies, including manual testing, automated testing, and simulation of real-world scenarios, to ensure the reliability and stability of game systems.
➢ Develop skills to optimize code, shaders, and resource usage, improving the overall performance and responsiveness of the game.
➢ Enhance problem-solving skills by diagnosing and addressing issues related to performance, collisions, input response, network latency, and other gameplay aspects.</td></tr>
<tr><td colspan="6">1. Frame Rate Counter: Develop a program that measures and displays the frame rate of a game in real-time. This is a fundamental metric for assessing game performance.
2. Input Lag Tester: Create a tool to measure and visualize the input lag between user actions (keyboard/mouse/controller) and the corresponding in-game response. Input lag can greatly affect gameplay experience.
3. Memory Profiler: Build a memory profiling tool that monitors the memory usage of your game in various scenarios. This can help identify memory leaks and inefficient memory usage patterns.
4. Load Time Analyzer: Design a program that measures and analyzes the loading times of different game scenes. This can help identify bottlenecks and optimize loading processes.
5. Collision Tester: Develop a tool that visualizes collision detection and physics interactions in your game. This can aid in identifying collision-related bugs and performance issues.</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ To generate detailed performance analysis reports that highlight critical metrics, areas for improvement, and actionable recommendations to enhance game performance.
➢ Demonstrate the ability to identify and document bugs related to performance, collisions, and gameplay responsiveness, along with providing steps to reproduce these issues.
➢ Optimized code to game projects, showcasing their proficiency in addressing performance bottlenecks and implementing efficient algorithms.
➢ Develop an automated testing suite that can simulate user interactions and verify expected outcomes, streamlining the testing process and improving game stability.
➢ Implement profiling and optimization strategies in real game projects, resulting in noticeable improvements in frame rates, loading times, and overall player experience.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.4**	**2.6**	**2.4**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	M(2)	M(2)	S(3)	M(2)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	S(3)	S(3)	M(2)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**2.6**	**2.6**	**3**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

V-Semester- Core					
Core	**Course Code: 82656**	**PORTFOLIO AND PRESENTATION**	**P**	**Credits: 4**	**Hours: 8**
Objectives	➢ Further develop the student's ability to develop their design and presentation skills by ➢ producing a range of work. ➢ Develop student's ability to critique their own work and that of their peers in a professional manner. ➢ Demonstrate a range of techniques and work that the student had developed in the course of their study. ➢ Give students further opportunity to demonstrate their ability to show the progression of ideas from the concept stage to completion.				
1. Create a logo and graphic signature for representing yourself. 2. Create your resume for a professional corporate company. 3. Create your blog for showing your personal development. 4. Create a Game trailer using the given gameplay footage. 5. Create your audio track to assist your demo reel. 6. Create and author a interactive portfolio using any authoring tool					
Outcomes	➢ Produce a portfolio that, demonstrates understanding and articulation, through ➢ drawings, concepts sketches, design documents and presentation an understanding of ➢ the design elements of the medium of their specialization. ➢ Develop Game that, will demonstrate the critical aspects of development as a media ➢ professional in the medium of specialization. ➢ Respond effectively to questions following oral presentation.				

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M(2)	M(2)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
CO2	M(2)	M(2)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
CO4	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)	S(3)

W.AV	**2.4**	**2.6**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	S(3)	S(3)	M(2)	M(2)	S(3)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)
CO3	M(2)	S(3)	M(2)	M(2)	S(3)
CO4	S(3)	M(2)	S(3)	S(3)	S(3)
CO5	S(3)	S(3)	S(3)	M(2)	S(3)
W.AV	**2.6**	**2.6**	**2.4**	**2.2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">VI – Semester-Core</th></tr>
<tr><td rowspan="2">Core</td><td rowspan="2">Course Code: 82661</td><td rowspan="2">GAME WRITING ESSENTIALS</td><td rowspan="2">T</td><td>Credits:</td><td>Hours:</td></tr>
<tr><td>4</td><td>4</td></tr>
<tr><td>Objectives</td><td colspan="5">● Understand the fundamental role of narrative in video games and differentiate between linear and interactive storytelling approaches.
● Analyze the application of classic storytelling structures and character development techniques in creating compelling game narratives.
● Evaluate the components of immersive game worlds and how consistent lore and environmental storytelling contribute to player engagement.
● Apply principles of branching narratives and player choices to create an interactive dialogue system that shapes the player's experience.
● Evaluate the adaptation of narrative techniques to various game genres and identify how narrative elements interact with gameplay mechanics.</td></tr>
<tr><td>UNIT-I</td><td colspan="5">Introduction to Game Writing: Understanding the role of narrative in games - Differentiating between linear and interactive storytelling - Exploring the impact of narrative on gameplay and player engagement - Introduction to key terminology in game writing.</td></tr>
<tr><td>UNIT-II</td><td colspan="5">Storytelling Techniques for Games: The hero's journey and other classic storytelling structures - Writing compelling characters: protagonists, antagonists, and supporting cast - Building arcs and character development over the course of a game - Using dialogue and monologue effectively to convey information and emotions.</td></tr>
<tr><td>UNIT-III</td><td colspan="5">World Building and Immersion: Creating believable and immersive game worlds - Developing consistent lore, history, and cultures - Integrating environmental storytelling to enhance player experience - Balancing player agency with maintaining the narrative vision.</td></tr>
<tr><td>UNIT-IV</td><td colspan="5">Interactive Narrative Design: The concept of branching narratives and player choices - Constructing meaningful choices with consequences - Non-linear storytelling: multiple endings and pathways - Implementing dialogue trees and interactive dialogue systems.</td></tr>
<tr><td>UNIT-V</td><td colspan="5">Writing for Different Game Genres: Adapting narrative techniques to different game genres (e.g., RPGs, action-adventure, visual novels) - Balancing narrative with gameplay mechanics in genre-specific ways - Collaborating with designers, artists, and programmers to achieve a cohesive vision - Guest speakers or case studies featuring professionals from the game industry.</td></tr>
<tr><td colspan="6">Text Book:
● "The Ultimate Guide to Video Game Writing and Design" by Flint Dille and John Zuur Platten.
References:
● "Game Writing: Narrative Skills for Videogames" by Chris Bateman
● "Creating Compelling Characters for Film, TV, Theatre, and Games" by Rib Davis
● "The Art of Game Design: A Book of Lenses" by Jesse Schell
● "Interactive Storytelling for Video Games: A Player-Centered Approach to Creating Memorable Characters and Stories" by Josiah Lebowitz and Chris Klug
● "Writing for Video Game Genres: From FPS to RPG" by Wendy Despain
● "Character Development and Storytelling for Games" by Lee Sheldon
● "Game Writing Handbook" by Rafael Chandler.</td></tr>
<tr><td colspan="6">Online Resources
● game-writing-essentials</td></tr>
</table>

Course Outcomes		Knowledge level
CO-1	**Summarize the significance of narrative in games and explain how interactive storytelling enhances player engagement and immersion.**	**K2**
CO-2	**Construct a plot outline and character profiles for a game concept that effectively utilizes classic storytelling elements to engage players.**	**K3**
CO-3	**Develop a detailed game world including lore, cultures, and environmental storytelling elements that create a cohesive and immersive player experience.**	**K4**
CO-4	**Assess the effectiveness of interactive dialogue choices in influencing the game's narrative direction and reflect on the implications of different player decisions.**	**K5**
CO-5	**Design a narrative-driven game concept for a specific genre that seamlessly integrates narrative and gameplay mechanics to deliver a unique player experience.**	**K6**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)
CO2	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	M(2)	M(2)	S(3)
CO3	L(1)	L(1)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)	M(2)	S(3)
CO4	L(1)	L(1)	S(3)	M(2)	L(1)	M(2)	S(3)	S(3)	M(2)	S(3)
CO5	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)
W.AV	**1.6**	**1.6**	**3**	**2.6**	**1.6**	**2**	**3**	**2.8**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	S(3)	M(2)	L(1)	M(2)
CO3	S(3)	S(3)	L(1)	L(1)	M(2)
CO4	S(3)	S(3)	L(1)	L(1)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	M(2)
W.AV	**3**	**3**	**1.6**	**1.2**	**1.8**

S–Strong (3), M-Medium (2), L-Low (1)

VI – Semester-Core					
Core	**Course Code: 82662**	**ADVANCED GAME MECHANICS**	**T**	**Credits: 4**	**Hours: 4**
Objectives	● **Understand the foundational role of gameplay mechanics in game design, and categorize different gameplay mechanic types.** ● **Explain the Mechanics-Dynamics-Aesthetics (MDA) framework and break down its components.** ● **Identify core dynamics in different game genres and strategies for designing compelling core dynamics.** ● **Examine factors that contribute to player flow state and engagement.** ● **Synthesize different types of fun and propose strategies to cater to diverse player preferences.**				
UNIT-I	**Introduction to Gameplay Mechanics -** Understanding the Role of Gameplay Mechanics in Game Design - Historical Evolution of Gameplay Mechanics - Core Concepts of Gameplay Mechanics - Basic Terminologies in Gameplay Mechanics - Exploration of Different Gameplay Mechanic Categories				
UNIT-II	**Mechanics-Dynamics-Aesthetics (MDA) Framework -** Introduction to the MDA Framework - Breakdown of Mechanics, Dynamics, and Aesthetics Components - Application of MDA Framework - MDA Framework's Role in Game Design and Iteration - Tuning Gameplay Mechanics using the MDA Framework				
UNIT-III	**Core Dynamics -** Defining Core Dynamics - Role of Core Dynamics - Identifying Core Dynamics in Different Game Genres - Strategies for Creating Compelling Core Dynamics - Balancing Challenge and Player Skill				
UNIT-IV	**Achieving Flow and Engagement -** Understanding Flow State and Player Engagement - Factors Influencing Flow State - Strategies for Facilitating Flow in Gameplay - Maintaining Engagement through Pacing and Progression - Feedback Systems and their Impact on Flow				
UNIT-V	**Diversity of Gameplay Experiences -** Exploring Different Types of Fun - Catering to Different Types of Players - Addressing Skill vs. Difficulty Balance - Incorporating Affordability in Gameplay Design - Implementing Orthogonality and Tension Maps in Game Mechanics				

Reference and Text Books:

Text Book:

- "Advanced Game Design: Mechanics, Dynamics, and Aesthetics", Jane Doe, GamePress Publishing.

References:

- "Challenges for Game Designers" by Brenda Brathwaite and Ian Schreiber
- "Game Mechanics: Advanced Game Design" by Ernest Adams and Joris Dormans
- "Level Up! The Guide to Great Video Game Design" by Scott Rogers
- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
- "Game Feel: A Game Designer's Guide to Virtual Sensation" by Steve Swink

Online Resources

- https://www.oreilly.com/library/view/game-mechanics-advanced/9780132946728/

Course Outcomes		**Knowledge level**
CO-1	**Summarize the historical evolution of gameplay mechanics and explain how core concepts shape player experiences in various game genres.**	**K2**
CO-2	**Apply the MDA framework to analyze and evaluate gameplay mechanics in existing games, identifying how mechanics contribute to player dynamics and aesthetics.**	**K3**

CO-3	**Analyze how core dynamics influence player engagement and retention in various game genres, and propose modifications to enhance gameplay experiences.**	**K4**
CO-4	**Design a game-level pacing and progression system that facilitates player flow, and evaluate its impact on player experiences.**	**K4**
CO-5	**Evaluate the skill vs. difficulty balance in various games, considering player types and discussing the effectiveness of the implemented mechanics.**	**K5**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)
CO2	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)
CO3	S(3)	S(3)	S(3)	S(3)	L(1)	M(2)	S(3)	S(3)	M(2)	S(3)
CO4	S(3)	S(3)	S(3)	M(2)	S(3)	M(2)	S(3)	S(3)	M(2)	S(3)
CO5	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)
W.AV	**2.6**	**3**	**3**	**2.6**	**2**	**2**	**3**	**3**	**2**	**3**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	L(1)	L(1)
CO2	S(3)	S(3)	M(2)	L(1)	M(2)
CO3	S(3)	S(3)	L(1)	M(2)	M(2)
CO4	S(3)	S(3)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	M(2)
W.AV	**3**	**3**	**2**	**1.6**	**1.8**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><td colspan="6">VI-Semester -
Core</td></tr>
<tr><td>Core</td><td>Course Code: 82663</td><td>GAME MECHANICS DEVELOPMENT - PRACTICAL</td><td>P</td><td>Credits: 4</td><td>Hours: 6</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Understand the fundamental concepts of player movement, collision, physics, camera control, interactions, combat mechanics, inventory, environmental mechanics, puzzle mechanics, resource management, and AI in game development.
➢ Interpret and analyze various game mechanics and their components, such as character movement, collision detection, camera control, and environmental interactions.
➢ Develop and implement functional game mechanics related to player movement, interactions, combat, puzzle-solving, and resource management using Unreal Engine.
➢ Evaluate the effectiveness of different game mechanics, such as combat systems, environmental puzzles, and AI behaviors, in enhancing player engagement and overall gameplay experience.
➢ Combine different game mechanics and systems to design and develop a cohesive gameplay experience, demonstrating creativity and problem-solving skills.</td></tr>
<tr><td colspan="6">1. Player Movement: Create a basic character movement (keyboard, controller) with movement mechanics like jumping, double-jumping, crouching, crawling, running.
2. Camera Control: Basic camera follow and movement, third-person camera orbiting and first-person camera perspective.
3. Interactions: Implement these simple game mechanics, object interaction (picking up, throwing, placing), door opening and closing, lever pulling and switch toggling.
4. Combat Mechanics: Create basic combat mechanics involving simple melee attacks and combos, ranged attacks (shooting, aiming), health and damage systems.
5. Environmental Mechanics: Create physics-based puzzles (rolling a boulder to open a path), leveraging elements in the environment (using a fire source to burn obstacles).</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Able to explain the basic concepts of player movement, collision, physics, camera control, interactions, combat mechanics, inventory systems, environmental puzzles, puzzle mechanics, resource management, and AI in video games.
➢ To describe and differentiate between different character movement mechanics, collision detection techniques, camera control methods, and environmental interaction systems in video games.
➢ To create and apply practical game mechanics, including character movement systems, object interactions, combat mechanics, puzzle-solving elements, and resource management mechanisms using Unreal Engine.
➢ To assess how various game mechanics contribute to player engagement, analyze the impact of combat systems and environmental puzzles on gameplay, and critique AI behaviors in creating challenging gameplay scenarios.
➢ To design and implement game mechanics that work together harmoniously, creating a well-rounded gameplay experience that combines player movement, combat, interactions, puzzles, resource management, and AI behaviors.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)	M(2)	S(3)	S(3)	S(3)
CO3	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	S(3)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	S(3)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**3**	**2.8**	**2.6**	**2.2**	**2.2**	**2**	**2.2**	**2.6**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	M(2)	M(2)	M(2)
CO2	S(3)	S(3)	M(2)	M(2)	S(3)
CO3	S(3)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	S(3)	M(2)	S(3)	M(2)
CO5	S(3)	S(3)	M(2)	M(2)	S(3)
W.AV	**3**	**3**	**2**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">VI-Semester-Elective-III</th></tr>
<tr><td>Elective</td><td>Course Code: 82664A</td><td>DISCIPLINE SPECIFIC ELECTIVE-IV A.ANIMATION FOR GAMES - PRACTICAL</td><td>P</td><td>Credits:4</td><td>Hours:4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Understand the fundamental principles of animation, including concepts like timing, spacing, and keyframes.
➢ Explain the significance of animation principles such as stretch and squash in creating dynamic and visually appealing character movements.
➢ Apply animation techniques to develop idle, attack, and movement sequences for the assigned character in a game environment.
➢ Evaluate the character's traits and style to design a heavy attack animation that aligns with their personality and the gameplay mechanics of a 2.5D fighting game.
➢ Combine multiple animation principles to create a cohesive animation set for the provided ball, demonstrating a deep understanding of how different principles interact to enhance motion realism.</td></tr>
<tr><td colspan="6">1. Create game Ready Animation for all of the 3 given movements. a. Idleb. Attack c. Forward walk
2. Create an acrobatic action Animation of 5 seconds for a parkour game
3. Create an animation using only the 2 principles stretch and squash . for the given Rigged Ball
4. Create an heavy attack animation for a 2.5D Fighting Game
5. Animate the given rigged Ball using multiple principles of animation example . anticipation, stretch and squash , follow through etc .</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Produce animations for idle, attack, and movement that exhibit proficiency in using animation software and tools.
➢ Develop a short acrobatic action animation that effectively showcases the character's abilities and captivates the audience within a limited 5-second timeframe.
➢ Implement the stretch and squash technique to the pillow's jumping animation, resulting in a dynamic and visually convincing depiction of elasticity and physics.
➢ Design a heavy attack animation that not only emphasizes power but also reflects the character's personality and fits seamlessly into the context of a 2.5D fighting game.
➢ Produce a comprehensive animation set for the ball, integrating principles like anticipation, follow-through, and secondary motion to simulate realism and enhance engagement in the game environment.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.4**	**2.6**	**2.4**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	M(2)	M(2)	S(3)	M(2)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	S(3)	M(2)
CO5	M(2)	S(3)	M(2)	M(2)	S(3)
W.AV	**2.4**	**2.6**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

VI-Semester-Elective-III					
Elective	**Course Code: 82664B**	**DISCIPLINE SPECIFIC ELECTIVE-IV B. STORYBOARDING FOR GAMES-PRACTICAL**	**P**	**Credits:4**	**Hours: 4**
Objectives	➢ **Understand the fundamental concepts of storyboarding in game development, including its role in conveying narrative, gameplay, and emotions.** ➢ **Explain the significance of effective storyboarding in creating engaging and immersive game experiences for players.** ➢ **Apply storyboarding techniques to visually depict game scenes, characters, and interactions through a series of frames.** ➢ **Analyze the impact of different visual storytelling choices on player engagement, narrative flow, and gameplay progression.** ➢ **Combine narrative elements, gameplay mechanics, and visual design principles to create comprehensive and compelling storyboards for various game scenarios.**				

1. **Basic Scene Setup:** Create a simple storyboard depicting the opening scene of a game, including the background, characters, and initial interactions.
2. **Character Introduction:** Design a storyboard that introduces a new game character, showcasing their appearance, abilities, and personality through a sequence of frames.
3. **Narrative Choices:** Develop a storyboard for a branching narrative moment, where the player's choices impact the direction of the story. Visualize how the choices unfold through different frames.
4. **Cutscene Transition:** Create a storyboard for a transition between gameplay and a cutscene. Illustrate the seamless transition from player control to a scripted sequence.
5. **Environmental Storytelling:** Design a series of storyboard frames that reveal the backstory of a game's environment, such as a deserted town or a ruined castle, without using explicit dialogue.

Outcomes	➢ **Produce well-structured storyboards that effectively communicate game scenes, characters, and interactions, showcasing proficiency in using visual storytelling techniques.** ➢ **Develop storyboards that effectively convey the game's narrative elements, drawing players into the world and creating emotional connections with the characters.** ➢ **Demonstrate the ability to convey complex ideas and concepts through visual mediums, showcasing a clear and coherent progression of events in the storyboards.** ➢ **Evaluate the effectiveness of different storyboard approaches in conveying emotions, gameplay mechanics, and narrative elements to enhance player immersion and engagement.** ➢ **Synthesize game design concepts and storytelling techniques to create innovative and unique storyboards that captivate players and enhance their overall game experience.**

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.4**	**2.6**	**2.4**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	M(2)	M(2)	S(3)	M(2)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	S(3)	M(2)
CO5	M(2)	S(3)	M(2)	M(2)	S(3)
W.AV	**2.4**	**2.6**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

<table>
<tr><th colspan="6">VI-Semester-Elective-III</th></tr>
<tr><td>Elective</td><td>Course Code: 82664C</td><td>DISCIPLINE SPECIFIC ELECTIVE-IV
C. GAME USER INTERFACE DESIGN-PRACTICAL</td><td>P</td><td>Credits: 4</td><td>Hours:4</td></tr>
<tr><td>Objectives</td><td colspan="5">➢ Understand the key principles of user interface design in the context of game development, including layout, readability, and user interaction.
➢ Explain the importance of user-friendly and intuitive UI design in enhancing player engagement, accessibility, and overall gameplay experience.
➢ Apply user interface design principles to create visually appealing and functional game UI elements that contribute to effective player interaction.
➢ Analyze the impact of different UI design choices on player immersion, ease of navigation, and clarity of information presentation.
➢ Synthesize design concepts, usability considerations, and aesthetic preferences to develop creative and innovative game UI solutions that cater to specific gameplay needs.</td></tr>
<tr><td colspan="6">1. Main Menu Design: Create a user-friendly main menu screen for a game, incorporating buttons for starting the game, accessing options, and quitting. Ensure an intuitive layout and appealing visual design.
2. HUD Elements: Design the heads-up display (HUD) for a game, including elements like health bars, score counters, and ammunition indicators. Make sure the HUD is informative and doesn't obstruct the gameplay.
3. Inventory System: Develop a UI for managing a player's inventory, allowing them to view, equip, and use items. Ensure the inventory layout is organized and easy to navigate.
4. Dialogue Interface: Design a dialogue system UI for in-game conversations, incorporating character portraits, text boxes, and response options to facilitate interactive storytelling.
5. Settings and Options: Create a settings menu where players can adjust graphics, audio, and gameplay options. Ensure clarity in labeling and offer sliders, checkboxes, and dropdowns for customization.</td></tr>
<tr><td>Outcomes</td><td colspan="5">➢ Understand the key principles of user interface design in the context of game development, including layout, readability, and user interaction.
➢ Explain the importance of user-friendly and intuitive UI design in enhancing player engagement, accessibility, and overall gameplay experience.
➢ Apply user interface design principles to create visually appealing and functional game UI elements that contribute to effective player interaction.
➢ Analyze the impact of different UI design choices on player immersion, ease of navigation, and clarity of information presentation.
➢ Synthesize design concepts, usability considerations, and aesthetic preferences to develop creative and innovative game UI solutions that cater to specific gameplay needs.</td></tr>
</table>

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.4**	**2.6**	**2.4**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	M(2)	M(2)	S(3)	M(2)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	S(3)	M(2)
CO5	M(2)	S(3)	M(2)	M(2)	S(3)
W.AV	**2.4**	**2.6**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

VI-Semester- Core					
Core	**Course Code: 82665A/ 82665B**	**PROJECT/ DISSERTATION**	**PR/ D**	**Credits: 6**	**Hours: 12**
Objectives	➢ **Develop a comprehensive and functional game prototype that demonstrates mastery of chosen programming languages and tools.** ➢ **Apply theoretical knowledge to address practical challenges within game development, showcasing problem-solving abilities.** ➢ **Demonstrate creativity and innovation in designing gameplay mechanics or features that exhibit a deep understanding of gaming concepts.** ➢ **Create a cohesive documentation outlining the development process, decision-making rationale, and technical aspects of the project.** ➢ **Present and defend the project's technical aspects and design choices through a well-structured dissertation or presentation.**				
Outcomes	➢ **Students will demonstrate a high level of proficiency in game development, showcasing skills in programming, game design, and implementation.** ➢ **Acquiring the ability to analyze complex problems within game development and devise effective solutions, displaying critical thinking and problem-solving capabilities.** ➢ **Demonstrating creativity in applying theoretical knowledge to create innovative gameplay mechanics, features, or visual elements.** ➢ **Producing comprehensive documentation that details the project's development process, methodologies used, challenges faced, and solutions implemented.** ➢ **Improved abilities to communicate technical concepts effectively, both in writing (documentation) and orally (presentations), fostering clearer articulation of ideas and technical decisions.** ➢ **Developing skills in project management, including time management, task prioritization, and resource allocation to successfully complete a substantial project within a specified timeline.** ➢ **Gaining familiarity with industry standards and best practices in game development, preparing students for potential careers in the field.** ➢ **Instilling confidence in their abilities to independently conceptualize, plan, execute, and present a significant project within the realm of game programming.**				

AIM OF THE PROJECT WORK

1. The aim of the project work is to acquire practical knowledge on the implementation of the programming concepts studied.
2. Each student should carry out individually one project work and it may be a work using the software packages that they have learned or the implementation of concepts from the papers studied or implementation of any innovative idea focusing on application oriented concepts.
3. The project work should be compulsorily done in the college only under the supervision of the department staff concerned.

VivaVoce

1. Viva-Voce will be conducted at the end of the year by both Internal (Respective Guides) and External Examiners, after duly verifying the Annexure Report available in the College, for a

total of 100 marks at the last day of the practical session.

2. Out of 100 marks, 25 marks for CIA and 75 for CEE (50 evaluation of project report + 25 Viva Voce).

Project Report Format

PROJECT WORK

TITLE OF THE DISSERTATION

Bonafide Work Done by

STUDENT NAME

REG. NO.

GUIDE NAME

Dissertation submitted in partial fulfillment of the requirements for the award of

<Name of the Degree>

ICAT Design and Media College, Chennai.

College Logo

Signature of the Guide Signature of the HOD

Submitted for the Viva-Voce Examination held on _____

Internal Examiner External Examiner

Month – Year

University Logo

CONTENTS

Course Outcome VS Programme Outcomes

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	M(2)	L(1)
CO2	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)	M(2)	M(2)	S(3)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)	M(2)	S(3)	M(2)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	M(2)	M(2)	L(1)	M(2)	S(3)	M(2)	M(2)
CO5	M(2)	S(3)	S(3)	S(3)	M(2)	M(2)	M(2)	M(2)	S(3)	S(3)
W.AV	**2.4**	**2.6**	**2.4**	**2.2**	**2.2**	**2**	**2.2**	**2.2**	**2.4**	**2**

S–Strong (3), M-Medium (2), L-Low (1)

Mapping Course Outcome VS Programme Specific Outcomes

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	S(3)	S(3)	S(3)	M(2)	M(2)
CO2	M(2)	M(2)	S(3)	M(2)	S(3)
CO3	M(2)	S(3)	S(3)	M(2)	M(2)
CO4	S(3)	M(2)	M(2)	S(3)	M(2)
CO5	M(2)	S(3)	M(2)	M(2)	S(3)
W.AV	**2.4**	**2.6**	**2.6**	**2.2**	**2.4**

S–Strong (3), M-Medium (2), L-Low (1)

Passing minimum

- A candidate shall be declared to have passed in each course if he/she secures not less than 40% marks in the End Semester Examinations and 40% marks in the Internal Assessment and not less than 40% in the aggregate, taking Continuous assessment and End Semester Examinations marks together.
- The passing minimum for CIA shall be 40% out of 25 marks (i.e.10 marks) in Theory/ Practical Examinations.
- The passing minimum for University Examinations shall be 40% out of 75 marks (i.e. 30 marks) for Theory /Practical papers.
- The candidates not obtain 40% in the Internal Assessment are permitted to improve their Internal Assessment marks in the subsequent semesters (2 chances will be given) by writing the CIA tests or by submitting assignments.
- Candidates, who have secured the pass marks in the End-Semester Examination and in the CIA but failed to secure the aggregate minimum pass mark (E.S.E + C I.A), are permitted to improve their Internal Assessment mark in the following semester and/or in University examinations.
- A candidate shall be declared to have passed in the Dissertation/Project report/Internship report if he/she gets not less than 40% marks in the Internal Assessment and End Semester Examinations and not less than 40% in the aggregate, taking Continuous assessment and End Semester Examinations marks together.
- A candidate who gets less than 40% in the Dissertation / Internship/ Project Report must resubmit the thesis. Such candidates need to take again the Viva-Voce on the resubmitted report/thesis.

18.2 Grading of the Courses

The following table gives the marks, Grade points, Letter Grades, and classifications meant to indicate the overall academic performance of the candidate.

Conversion of Marks to Grade Points and Letter Grade (Performance in Course / Paper)

RANGE OF MARKS	GRADE POINTS	LETTER GRADE	SCRIPTION
- 100	**9.0 – 10.0**	**O**	**tstanding**
- 89	**8.0 – 8.9**	**D+**	**cellent**
- 79	**7.5 – 7.9**	**D**	**tinction**
- 74	**7.0 – 7.4**	**A+**	**y Good**
- 69	**6.0 – 6.9**	**A**	**od**
- 59	**5.0 – 5.9**	**B**	**erage**
- 49	**4.0 – 4.9**	**C**	**isfactory**
- 39	**0.0**	**U**	**appear**
SENT	**0.0**	**AAA**	**SENT**

a) Successful candidates passing the examinations and earning a GPA between 9.0 and 10.0 and marks from 90 – 100 shall be declared to have Outstanding (O).

b) Successful candidates passing the examinations and earning GPA between 8.0 and 8.9 and marks from 80 - 89 shall be declared to have Excellent (D+).

c) Successful candidates passing the examinations and earning GPA between 7.5 – 7.9 and marks from 75 - 79 shall be declared to have Distinction (D).

d) Successful candidates passing the examinations and earning GPA between 7.0 – 7.4 and marks from 70 - 74 shall be declared to have Very Good (A+).

e) Successful candidates passing the examinations and earning GPA between 6.0 – 6.9 and marks from 60 - 69 shall be declared to have Good (A).

f) Successful candidates passing the examinations and earning GPA between 5.0 – 5.9 and marks from 50 - 59 shall be declared to have Average (B).

g) Successful candidates passing the examinations and earning GPA between 4.0 – 4.9 and marks from 40 - 49 shall be declared to have Satisfactory (C).

h) Candidates earning GPA between 0.0 and marks from 00 - 39 shall be declared to have Re-appear (U).

i) Absence from an examination shall not be taken as an attempt.

From the second semester onwards the total performance within a semester and continuous performance starting from the first semester are indicated respectively **by** Grade Point Average (GPA) and Cumulative Grade Point Average (CGPA).

These two are calculated by the following formulate

GRADE POINT AVERAGE (GPA) = $\Sigma_i C_i \; G_i / \Sigma_i C_i$

GPA = Sum of the multiplication of grade points by the credits of the courses

Sum of the credits of the courses in a Semester

18.3 Classification of the final result

The final result of the candidate shall be based only on the CGPA earned by the candidate.

a) Successful candidates passing the examinations and earning CGPA between 9.5 and 10.0 shall be given Letter Grade (O+) and those who earned CGPA between 9.0 and 9.4 shall be given Letter Grade (O) and declared to have First Class –Exemplary*.

b) Successful candidates passing the examinations and earning CGPA between 7.5 and 7.9 shall be given Letter Grade (D), those who earned CGPA between 8.0 and 8.4 shall be given Letter Grade (D+) and those who earned CGPA between 8.5 and 8.9 shall be given Letter Grade (D++) and declared to have First Class with Distinction*.

c) Successful candidates passing the examinations and earning CGPA between 6.0 and 6.4 shall be given Letter Grade (A), those who earned CGPA between 6.5 and 6.9 shall be given Letter Grade (A+), and those who earned CGPA between 7.0 and 7.4 shall be given Letter Grade (A++) and declared to have First Class.

d) Successful candidates passing the examinations and earning CGPA between 5.0 and 5.4 shall be given Letter Grade (B) and those who earned CGPA between 5.5 and 5.9 shall be given Letter Grade (B+) and declared to have passed in the Second Class.

e) Successful candidates passing the examinations and earning CGPA between 4.0 and 4.4 shall be given Letter Grade (C) and those who earned CGPA between 4.5 and 4.9 shall be given Letter Grade (C+) and declared to have passed in the Third Class.

f) Absence from an examination shall not be taken as an attempt.

Final Result

CGPA	Grade	Classification of Final Result
9.5 – 10.0 9.0 and above but below 9.5	**O+** **O**	First Class – Exemplary*
8.5 and above but below 9.0 8.0 and above but below 8.5 7.5 and above but below 8.0	**D++** **D+** **D**	First Class with Distinction*
7.0 and above but below 7.5 6.5 and above but below 7.0 6.0 and above but below 6.5	**A++** **A+** **A**	First Class
5.5 and above but below 6.0 5.0 and above but below 5.5	**B+** **B**	Second Class
4.5 and above but below 5.0 4.0 and above but below 4.5	**C+** **C**	Third Class
0.0 and above but below 4.0	**U**	Re-appear

CUMULATIVE GRADE POINT AVERAGE (CGPA) = $\Sigma_n \Sigma_i C_{ni} G_{ni} / \Sigma_n \Sigma_i C_{ni}$

CGPA = Sum of the multiplication of grade points by the credits of the entire programme

Sum of the credits of the course for the entire Programme

Where '**Ci**' is the Credit earned for Course i in any semester; '**Gi**' is the Grade Point obtained by the student for Course i and 'n' refers to the semester in which such courses were credited.

CGPA (Cumulative Grade Point Average) = Average Grade Point of all the Courses passed starting from the first semester to the current semester.

Note: * The candidates who have passed in the first appearance and within the prescribed Semesters of the UG Programme (Major, Allied, and Elective courses alone) are eligible for this classification.