C–1616

Sub. Code
82642

B.Sc. DEGREE EXAMINATION, APRIL 2024

Fourth Semester

Game Programming

WEB GAME PROGRAMMING

(2019 onwards)

Duration : 3 Hours Maximum : 75 Marks

Part A (10 × 2 = 20)

Answer **all** questions.

1. Expand HTML.
2. What is canvas?
3. What are the types of arrays?
4. What is form validation?
5. Pen down the scrolling effect.
6. What is sprite?
7. What are the controlled game elements?
8. List the game play programming.
9. What is render?
10. Define 2D box.

Part B (5 × 5 = 25)

Answer **all** questions.

11. (a) Explain HTML lists.

Or

(b) Describe the application of canvas.

12. (a) Compare ID and 2D arrays.

Or

(b) Explain the web development framework.

13. (a) Write down the XML parsing and JSON parsing.

Or

(b) How to develop a sprite animation?

14. (a) Discuss the background scrolling.

Or

(b) How to design a game with user interface?

15. (a) Write note on physics programming.

Or

(b) Explain the request and response in web game development.

Part C

$(3 \times 10 = 30)$

Answer **all** questions.

16. (a) Explain in detail about HTML tables and HTML images.

Or

(b) Explain in detail Java Script Expressions and Operators.

17. (a) Discuss methods and Functions of canvas in detail.

Or

(b) What is User Interface? Explain the process of UI Designing with suitable example.

18. (a) Explain in detail about collision detection.

Or

(b) Explain the application of asynchronous webpage in detail.

———————

C–2358	Sub. Code
	82613

B.Sc. DEGREE EXAMINATION, APRIL 2024

First Semester

Game Programming

FUNDAMENTALS OF PROGRAMMING

(2023 onwards)

Duration : 3 Hours Maximum : 75 Marks

Part A $(10 \times 1 = 10)$

Answer **all** questions.

1. What is the purpose of a loop in programming?

   (a) To perform a one-time calculation

   (b) To execute a set of instructions repeatedly

   (c) To define a function

   (d) To declare constants

2. What does it mean to pass values to a function in programming?

   (a) To execute a function without any input

   (b) To return a value from a function

   (c) To provide data or parameters as input to a function

   (d) To create a new function within an existing function

3. In programming, what are pointers primarily used for?

(a) Representing text data

(b) Storing functions

(c) Managing memory and manipulating addresses

(d) Creating multi-dimensional arrays

4. What is the primary purpose of function pointers in programming?

(a) To store integer values

(b) To create dynamic arrays

(c) To call different functions dynamically at runtime

(d) To point to elements within an array

5. What is the purpose of a destructor in OOP?

(a) To create new objects

(b) To release resources and perform cleanup when an object is destroyed

(c) To define a new class

(d) To implement method overloading

6. What is polymorphism in OOP?

(a) The ability of an object to have multiple data types

(b) The concept of bundling data and methods

(c) The ability to perform the same operation in different ways

(d) The process of defining multiple constructors in a class

7. What is the purpose of namespaces in programming?

(a) To manage file handling operations

(b) To group related code and avoid naming conflicts

(c) To implement exception handling

(d) To define delay and timer functions

8. What is the role of exception handling in a program?

(a) To design and structure the project

(b) To manage data handling using files

(c) To handle and recover from runtime errors and exceptional conditions

(d) To create templates for code reuse

9. Which STL container is typically implemented as a doubly-linked list and allows for efficient insertions and deletions at any position?

(a) Vector (b) List

(c) Deque (d) Queue

10. In STL, what is the purpose of the binary search algorithm?

(a) To search for a value in an unsorted container

(b) To find the maximum element in a container

(c) To efficiently search for a value in a sorted container

(d) To sort a container using a binary approach

Part B (5 × 5 = 25)

Answer **all** questions.

11. (a) Discuss the purpose of constants in programming and explain why they are valuable.

Or

(b) Explain with suitable examples the purpose of conditional statements in programming.

12. (a) Explain with examples the concept of a two-dimensional array in programming.

Or

(b) Explain what dynamic arrays are and how they overcome limitations of static arrays.

13. (a) Define encapsulation and discuss its role in OOP.

Or

(b) Explain the purpose of destructors in OOP and their role in resource cleanup.

14. (a) Explain the importance of designing and structuring a project.

Or

(b) Discuss the concept of exception handling and its significance in dealing with runtime errors.

15. (a) Compare and contrast the vector, list, and deque STL containers in terms of their characteristics and use cases.

Or

(b) Discuss the binary search algorithm in the STL.

Part C

$(5 \times 8 = 40)$

Answer **all** questions.

16. (a) Define recursive functions in programming and discuss their applications.

Or

(b) Define functions in programming and explain their role in modularizing code.

17. (a) Describe the process of passing arrays to functions in programming.

Or

(b) Describe with examples how function pointers are declared and how they can be used to call different functions dynamically.

18. (a) Explain the concept of abstract classes and methods and discuss how it simplifies complex systems.

Or

(b) Describe with suitable example the process of function overloading and how it allows multiple functions with the same name but different parameters.

19. (a) Define enumerations and their use in defining sets of named integer constants. Explain with suitable example, how enumerations can enhance code readability and maintainability?

Or

(b) Discuss the concept of exception handling in the context of file operations. Explain how to handle exceptions such as file not found or permission denied.

20. (a) Discuss the use of random number generation in the STL. Explain how random numbers can be generated using the <random> library and describe the steps involved in creating a random number generator.

Or

(b) Discuss with suitable examples the characteristics of linear data structures such as arrays, vectors, and lists.

———————

C–2359	Sub. Code
	82615

B.Sc. DEGREE EXAMINATION, APRIL 2024

First Semester

Game Programming

GAME ANALYSIS AND DESIGN

(2023 onwards)

Duration : 3 Hours Maximum : 75 Marks

Part A $(10 \times 1 = 10)$

Answer **all** questions.

1. What is the core concept of MDA framework in game design?

   (a) Mechanics, Devices, Aesthetics

   (b) Microtransactions, DLC, Achievements

   (c) Mechanics, Dynamics, Aesthetics

   (d) Multiplayer, Difficulty, Action

2. In the context of game design, what does “circumspection” involve?

   (a) Careful consideration and planning of game mechanics

   (b) Rapid decision-making during game play

   (c) Predicting the outcome of a game

   (d) Game testing and quality assurance

3. What term is used to describe a game narrative with multiple storylines that can be experienced in different sequences?

(a) Linear plot

(b) Braided plot

(c) Network plot

(d) Open world

4. How can game designers effectively integrate emergence and progression in game design?

(a) By eliminating all elements of chance

(b) By restricting player choices

(c) By carefully balancing mechanics and player experiences

(d) By avoiding strategy elements

5. What re common elements found in successful game worlds?

(a) Highly predictable storylines

(b) Linear progression and gameplay

(c) Consistency, believability and interactivity

(d) Frequent changes in game mechanics

6. What is the primary purpose of level design in game development?

(a) To create realistic 3D models

(b) To design the game's storyline

(c) To craft game environments and challenges

(d) To compose background music for the game

7. In game design, what is "empathizing" with the player?

(a) Creating characters that resemble the player

(b) Understanding and considering the player's perspective and emotions

(c) Achieving high scores in the game

(d) Developing cooperative gameplay

8. In game design, what are the methodologies used to balance a game's difficulty and mechanics?

(a) Game aesthetics and art style

(b) Character development and narrative structure

(c) Player feedback and play testing

(d) Achievements and leader boards

9. What do player demographics in game design refer to?

(a) The statistics of in-game items and resources

(b) The physical characteristics of players

(c) The age, gender, location and other characteristics of players

(d) The behaviour of players during game play

10. What characterizes strong player communities in gaming?

(a) Exclusivity and secrecy

(b) Hostility towards new players

(c) Support, cooperation and shared interests

(d) Isolation from the gaming community

Part B (5 × 5 = 25)

Answer **all** questions.

11. (a) Discuss the importance of user-centered design and the role of usability in HCI.

Or

(b) Differentiate between various types of games and game genres.

12. (a) Discuss the key dramatic elements commonly found in video games, such as character development, plot twists and narrative arcs.

Or

(b) Describe the various channels of information used to convey game elements and events to players during game play.

13. (a) Identify discuss the common elements that are often found in successful game worlds.

Or

(b) Differentiate between real architecture and virtual architecture within game environments.

14. (a) Explain how game designers use the concept of "focusing" to direct the player's attention.

Or

(b) Discuss the role of game mechanics in shaping player interactions and experiences.

15. (a) Discuss the methods and mechanics that can change a player's type during gameplay.

Or

(b) Explain the role of player taxonomy in tailoring gameplay.

Part C $(5 \times 8 = 40)$

Answer **all** questions.

16. (a) Explain the Mechanics-Dynamics-Aesthetics (MDA) framework and how it helps in understanding the player's experience.

Or

(b) Discuss the role of tension maps in game design and how they visualize player engagement.

17. (a) Differentiate between strategy and skill in the context of game design. Explain how player strategy and skill development are essential for success in various types of games.

Or

(b) Define emergence in the context of game design and how it leads to unexpected player experiences. Describe game progression and its role in guiding players through a game's challenges and objectives.

18. (a) Discuss the role of audio, including background music and sound effects, in creating immersive game environments. Explain how audio contributes to world-building and atmosphere.

Or

(b) Discuss the importance of balancing art and technology in game development. Explain how art and technology are interconnected and must work harmoniously to create a successful game.

19. (a) Define “skill” and ‘chance” in the context of gameplay. Discuss how these elements affect the player's performance and outcomes in games.

Or

(b) Explain the concept of “interest curves” in game design. How do these curves depict player engagement and challenge levels over time?

20. (a) Discuss the ethical challenges that game developers may face and the benefits of designing ethically responsible games.

Or

(b) Define the “flow of influence” in game design. How does this concept impact player agency and decision-making within games?

———————

C–2360	Sub. Code
	82623

B.Sc. DEGREE EXAMINATION, APRIL 2024

Second Semester

Game Programming

GRAPHICS PROGRAMMING

(2023 onwards)

Duration : 3 Hours Maximum : 75 Marks

Part A $(10 \times 1 = 10)$

Answer **all** questions.

1. What is the primary purpose of graphics libraries in computer graphics?
   (a) Audio processing
   (b) Image recognition
   (c) Enhancing graphical operations
   (d) Network communication
2. Which term is used to represent a location in a two-dimensional space in computer graphics?
   (a) Node (b) Vertex
   (c) Pixel (d) Segment
3. What is the basic purpose of "Hello Triangle" in computer graphics?
   (a) Displaying text
   (b) Drawing a basic geometric shape
   (c) Animating characters
   (d) Creating 3D models

4. In graphics programming, what is the purpose of an index buffer?

(a) Storing texture data

(b) Specifying the color of vertices

(c) Defining the order of vertex rendering

(d) Managing screen resolution

5. In computer graphics, what does a rotation matrix primarily handle?

(a) Changing the size of an object

(b) Altering the position of vertices

(c) Modifying the orientation of an object

(d) Adjusting the color of an object

6. Which matrix operation is responsible for resizing an object in computer graphics?

(a) Transformation matrix

(b) Rotate matrix

(c) Scale matrix

(d) Skew matrix

7. What is the purpose of back face culling in 3D graphics?

(a) Enhancing lighting effects

(b) Improving texture resolution

(c) Removing invisible faces from rendering

(d) Adjusting color gradients

8. Which technique is commonly used for player movement in 3D games?

(a) Keyframe animation

(b) Height mapping

(c) Skeletal animation

(d) Terrain generation

9. Which shading technique focuses on calculating lighting at each pixel of a 3D model?

(a) Flat shading (b) Gouraud shading

(c) Phong shading (d) Cell shading

10. What is the primary purpose of a skybox in 3D graphics?

(a) Storing texture data

(b) Simulating atmospheric effects

(c) Defining terrain boundaries

(d) Enhancing specular highlights

Part B $(5 \times 5 = 25)$

Answer **all** questions.

11. (a) Explain the term "vertex" in computer graphics. Provide an example of its usage in a graphical context.

Or

(b) Describe the structure of a typical graphics program. Highlight the key components and their roles in the program's execution.

12. (a) Explain the process of drawing a quad in computer graphics. Provide a step-by-step description of the necessary stages and functions involved.

Or

(b) Discuss the concept of buffers in graphics programming. Highlight the types of buffers used and their respective roles in the rendering pipeline.

13. (a) Explain the concept of a transformation matrix in computer graphics. Provide examples of how it can be used to manipulate objects in a 3D space.

Or

(b) Discuss the significance of the eye, look at, and up vectors in defining the orientation and position of a virtual camera. How do these vectors contribute to camera positioning in 3D space?

14. (a) Explain the process of loading 3D models in a graphics application. Discuss the common file formats and considerations involved in loading 3D models.

Or

(b) Describe the concept of terrain creation in computer graphics. Discuss the steps involved in simple terrain creation and the role of height maps in this process.

15. (a) Explain the concept of multi-texturing in the context of terrain rendering. Discuss the advantages and challenges of using multiple textures on a terrain.

Or

(b) Discuss the role and implementation of blend maps in multi-texturing of terrain. How do blend maps contribute to achieving realistic and varied terrain surfaces?

Part C

$(5 \times 8 = 40)$

Answer **all** questions.

16. (a) Elaborate on the concept of coordinate spaces in computer graphics. How do different coordinate spaces contribute to the rendering of graphical elements? Provide examples.

Or

(b) Examine the role of graphics libraries in the development of computer graphics applications. Discuss the advantages and disadvantages of using graphics libraries in programming.

17. (a) Explore the "Hello Triangle" concept in computer graphics. Discuss its significance in learning graphics programming and provide an example implementation.

Or

(b) Examine the role of the rendering pipeline in graphics programming. Discuss each stage of the pipeline and how it contributes to the final image creation.

18. (a) Explore the operations involved in the MVP (Model-View-Projection) matrix in computer graphics. Discuss how it is used in transforming 3D models from model space to screen space.

Or

(b) Discuss various camera operations in computer graphics. Include details about camera translation, rotation, and zooming. Provide examples to illustrate these operations.

19. (a) Explore the techniques and considerations involved in player movement in a 3D gaming environment. Discuss the challenges and solutions for implementing smooth and responsive player controls.

Or

(b) Investigate the concept of multi-textured terrain in computer graphics. Discuss how multiple textures can be applied to enhance the visual appearance of a terrain.

20. (a) Discuss the concept of using multiple shaders in a graphics application. Explore how different shaders can be employed for various visual effects, and their interactions in a rendering pipeline.

Or

(b) Examine the techniques involved in implementing per-pixel lighting in 3D graphics. Discuss the advantages and potential challenges in achieving realistic lighting effects.

———————

C–2361	Sub. Code
	82625

B.Sc. DEGREE EXAMINATION, APRIL 2024

Second Semester

Game Programming

ALGORITHMS AND DATA STRUCTURES

(2023 onwards)

Duration : 3 Hours Maximum : 75 Marks

Part A $(10 \times 1 = 10)$

Answer **all** questions.

1. What is the primary purpose of algorithmic problem-solving?

   (a) Aesthetic design

   (b) Efficient solutions to computational problems

   (c) Software development

   (d) Network optimization

2. Which of the following is a common analysis framework used to evaluate algorithm efficiency?

   (a) Syntax analysis

   (b) Complexity analysis

   (c) Object-oriented analysis

   (d) Linguistic analysis

3. Which algorithmic paradigm does Binary Search belong to?

(a) Brute Force

(b) Divide and Conquer

(c) Greedy Algorithms

(d) Dynamic Programming

4. In the context of string matching, which algorithm involves systematically checking each position in a text for a pattern match?

(a) Quick Sort

(b) Binary Search

(c) Sequential Search

(d) Brute Force String Matching

5. Which data structure is commonly used to implement a priority queue in heapsort?

(a) Linked list

(b) Stack

(c) Array-based heap

(d) Binary search tree

6. In the context of searching, which algorithm operates by repeatedly dividing the search interval in half?

(a) Linear Search

(b) Depth-First Search

(c) Breadth-First Search

(d) Binary Search

7. Which algorithm is used for finding the Minimum Spanning Tree (MST) in a connected, undirected graph?

   (a) Warshall's algorithm

   (b) Prim's algorithm

   (c) Kruskal's algorithm

   (d) Dijkstra's algorithm

8. In dynamic programming, what is the purpose of memoization?

   (a) Minimizing memory usage

   (b) Storing and reusing computed results to avoid redundant calculations

   (c) Optimizing algorithmic complexity

   (d) Balancing load distribution

9. Which problem involves finding the most efficient way to assign a set of tasks to a set of resources with minimum cost?

   (a) Knapsack Problem

   (b) Assignment Problem

   (c) Hamiltonian Circuit Problem

   (d) Travelling Salesman Problem

10. In the context of complexity classes, which class represents problems that can be verified quickly once a solution is provided?

   (a) P

   (b) NP

   (c) NP-Complete

   (d) P and NP

Part B (5 × 5 = 25)

Answer **all** questions.

11. (a) Explain the fundamentals of algorithmic problem-solving. Discuss the key steps involved in approaching and solving computational problems algorithmically.

Or

(b) Identify and discuss important problem types in algorithmic problem-solving. Provide examples of problems falling into different categories and explain their characteristics.

12. (a) Explore the concept of the closest pair problem in computational geometry. Describe the exhaustive search algorithm for finding the closest pair of points and discuss its time complexity.

Or

(b) Examine the sequential search algorithm and its application in string matching. Discuss scenarios where sequential search is an appropriate solution and its limitations.

13. (a) Describe the "Transform and Conquer' approach in algorithm design. Provide examples of algorithms that use this technique, such as heapsort.

Or

(b) Explain the concept of presorting and its application in algorithmic problem-solving. Provide examples where presorting is advantageous.

14. (a) Explain the concept of optimal binary search trees in the context of dynamic programming. Discuss the construction and optimization process for such trees.

Or

(b) Describe the greedy technique in algorithm design. Provide examples of algorithms that utilize the greedy approach, such as Prim's algorithm.

15. (a) Describe the Assignment Problem and how it is solved using the branch and bound technique. Discuss the steps involved in the branch and bound algorithm.

Or

(b) Discuss the Subset Sum Problem and its applications. Explain how the backtracking algorithm can be used to find a subset that sums to a given target.

Part C $(5 \times 8 = 40)$

Answer **all** questions.

16. (a) Discuss the significance of dynamic programming in algorithmic problem-solving. Provide examples of problems where dynamic programming is particularly useful and explain the approach taken.

Or

(b) Examine the mathematical analysis of algorithms using recurrence relations. Discuss the steps involved in solving recurrence relations and their application in analyzing recursive algorithms.

17. (a) Investigate the Binary Search algorithm. Explain how it works and discuss its time complexity. Highlight the conditions under which Binary Search is most effective.

Or

(b) Explore the concept of exhaustive search in algorithmic problem-solving. Discuss its application in solving problems and analyze the trade-offs between accuracy and computational efficiency.

18. (a) Examine the divide and conquer approach in algorithm design, focusing on the merge sort algorithm. Discuss its efficiency, advantages, and limitations.

Or

(b) Discuss the concept of heaps and heap sort. Explore the heapify operation and the steps involved in the heap sort algorithm. Compare heap sort with other sorting algorithms in terms of efficiency and use cases.

19. (a) Investigate Kruskal's algorithm for finding the Minimum Spanning Tree (MST) of a graph. Discuss its advantages, limitations, and compare it with other MST algorithms.

Or

(b) Explain the concept of computing a binomial coefficient using dynamic programming. Discuss the algorithmic steps involved, its time complexity, and scenarios where it is particularly useful.

20. (a) Explain the concept of Decision Trees and their application in solving optimization problems. Discuss how decision trees are constructed and used for problem-solving.

Or

(b) Investigate the complexity classes P, NP, and NP-Complete. Discuss their definitions, relationships, and the implications of a problem being classified as NP-Complete.