# 20AD41E7 - GAME PROGRAMMING

|  |  |  |  |
| --- | --- | --- | --- |
| Course Category: | Professional Elective | Credits: | 3 |
| Course Type: | Theory | Lecture-Tutorial-Practical: | 3-0-0 |
| Prerequisite: | Basic mathematical knowledge to solve problems and programming. | Sessional Evaluation:  Univ. Exam Evaluation:  Total Marks: | 40  60  100 |
| Objectives: | * To Understand the fundamentals of Game Programming * To Understand the Various Tools used for the Game Programming. * To learn how to apply the Software Engineering Mechanism in the Game Programming. * To Understand the and acquire knowledge of Game Engine Support System and Game Loop and Real Time Simulation. | | |

|  |  |  |
| --- | --- | --- |
| Course Outcomes | Upon successful completion of the course, the students will be able to: | |
| CO1 | Understand the Architecture of the Game Engine. |
| CO2 | Acquire the Knowledge of Version Control for Developing the Game. |
| CO3 | Acquire the Knowledge of Microsoft Visual Studio for Developing the Game. |
| CO4 | Analyse the Engine Support System and Management System |
| CO5 | Apply the Concept of Concept of Containers in the Game Programming. |
| CO6 | Acquire the Knowledge of Game loops and find the necessity of the Human Interfacing Devices in the Game Designing. |
| Course Content | UNIT-I  Structure of the Typical Game Team, what is Game, What is a Game Engine, Engine Differences Across Genres, Game Engine Survey, Runtime Engine Architecture, Tools and Asset Pipeline.  UNIT-II  **Tools of the Trade:** Version Control: Why we use Version Control, Common Version Control System, Overview of Subversion and TortoiseSVN, Setting up Code repository in Google, Installing TortoiseSVN, File version Updating, Committing, Multiple Check Out and Branching and Merging, Deleting Files.  UNIT-III  **Tools of the Trade:** Version Control: Microsoft Visual Studio: Source files Headers and Translation units, Libraries, Executables and Dynamic Link Libraries, Projects and Solutions, Build Configurations, Project Configuration Tutorials, Debugging your code, Profiling tools, Memory leak and Corruption Detection, Other Tools.  UNIT-IV  **Engine Support System: Subsystem Start up and Shut down**: C++ Static Initialization Order (or Lack Thereof), Construct on Demand, A simple Approach that Works, Some Examples from Real Engines, Naughty Dog’s Uncharted Series and The Last of Us.  **Memory Management System**: Optimizing Dynamic Memory Allocation, Memory Fragmentation.  UNIT-V  **Engine Support System: Containers:** Container Operations, Iterators, Algorithmic Complexity, Building Container Classes.  **Engine Configuration:** Loading and Saving Options, Per User Options, Configuration in some real Engines.  UNIT-VI  **Game Loop and Real Time Simulation:** The Rendering Loop, The Game Loop, Game Loop Architectural Styles, Abstract Time Lines, Measuring and Dealing with Time.  **Human Interface Devices**: Types of Human Interface Devices, Interfacing with HID, Types of Inputs, Types of Outputs. | |
| Text Books &  Reference  Books | **TEXT BOOKS:**   1. Game Engine Architecture, 2nd Edition, Jason Gregory, A K Peters, 2014.   **REFERENCE BOOKS:**   1. Best of Game Programming Gems, Mark DeLoura, Course Technology, Cengage Learning, 2014. 2. Rules of Play: Game Design Fundamentals, Katie Salen and Eric Zimmerman, MIT Press, 2003. | |
| E-Resources | 1. <https://www.futurelearn.com> 2. <https://www.codingame.com> | |