



Course Identification

Name of program – Code:	COMPUTER SCIENCE TECHNOLOGY – VIDEO GAME PROGRAMMING (420.BX)
Course title:	INTERNET PROGRAMMING
Course number:	420-PRM-AS
Group:	05860
Teacher's name:	ZOHREH MOEINI
Duration:	Extended
Semester:	FALL 2025

Student Identification

Name: _____	Student number: _____
Date: _____	Result: _____

Standard of the Evaluated Competency

Statement of the evaluated competency – Code

Develop native applications with a database – 00SS

Evaluated elements of the competency

- 1. Analyze the application development project.
- 2. Prepare the computer development environment
- 3. Prepare the database(s)
- 4. Create the graphical user interface
- 5. Program the application logic
- 6. Control the quality of the application

Instructions

- Read each question carefully and answer as specified
- Write your answers CLEARLY and CONCISELY
- Show all the steps to your work and number your answers and pages
- Class notes are not allowed, and students may not use the dictionary.
- No break is allowed during this exam. Students are not allowed to exit the examination room before half of the allotted time has passed. Once a student has exited the classroom, he/she may not re-enter (IPEL – Article 5.12.4).
- The teacher will not answer questions during the exam.
- Students must remain silent during the exam.
- It is the teacher’s responsibility to identify language errors. If such errors are found, teachers may deduct up to 20% of the final grade (IPEL – Article 5.7).
- Plagiarism, attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. (IPEL – Article 5.16).

Mark Breakdown

This evaluation is worth 100 points, distributed as follows:

- | | |
|----------------------------------------------------|---------------|
| • Setup (Scene, camera, renderer, resize handling) | For 15 points |
| • Lighting (Ambient and point light properly used) | For 15 points |
| • Objects(Sun + 3 planets with colors/sizes) | For 20 points |
| • Animation (Smooth and distinct orbits) | For 20 points |
| • Controls(OrbitControls working properly) | For 10 points |
| • Creativity(Visual polish and realism) | For 10 points |
| • Code Quality(Readability and comments) | For 10 points |
| TOTAL: 100 Points | |

Final Project Instructions — Three.js Mini Solar System

Goal

Create a **small interactive Solar System** scene using **Three.js**. The goal is to demonstrate your understanding of the basics — scene setup, camera, lighting, materials, animation, and interactivity. Your project should be creative and functional while keeping the code clean and simple.

Requirements

1. Scene Setup

- Use Scene, PerspectiveCamera, and WebGLRenderer.
- Handle window resize events properly.

2. Objects

- Add a **Sun** in the center.
- Add at least **3 planets** orbiting around it.
- Give each planet a different **size** and **color**.

3. Lighting

- Include at least one PointLight (for the Sun).
- Include an AmbientLight for overall brightness.

4. Animation

- Each planet should **orbit around the Sun** smoothly.
- Use Math.cos() and Math.sin() to calculate orbit positions.

5. Controls

- Use OrbitControls to move and zoom the camera with the mouse.

6. Code Organization

- Keep your code readable with comments.
- Use functions for repeated logic (like creating planets).

4-Week Plan

Week 1: Learn the basics of Three.js (scene, camera, renderer).

Week 2: Add Sun and planets with proper materials and colors.

Week 3: Implement orbits, animation, and lighting.

Week 4: Add camera controls, polish visuals, and finalize your project.

Tools

- **Three.js CDN:** <https://unpkg.com/three@0.168.0/build/three.module.js>
- **OrbitControls:** <https://unpkg.com/three@0.168.0/examples/jsm/controls/OrbitControls.js>
- **Editor Recommendation:** VS Code with Live Server extension.

How to Run

1. Save your project as index.html.
2. Open the folder in VS Code.
3. Right-click → **Open with Live Server.**
4. The project should open in your browser.

Submission Guidelines

- Submit a **ZIP folder** containing your index.html and any additional files.
- Github Account

Element of competency: Develop transactional applications- 00SU	
Performance criteria	weight
4.1 Appropriate use of markup language (5 points) 4.2 Suitable creation and use of style sheets (5 points) 4.3 Proper integration of images (5 points) 4.4 Adaptation of the interface based on the display format and resolution (5 points)	/20
2.1 Proper customizing of the spreadsheet interface (6 points) 2.2 Appropriate choice of the type of table and graph to be produced (6 points) 2.3 Appropriate choice and use of search, logic and calculation functions (6 points) 2.4 Development of appropriate mathematical formulas (6 points) 2.5 Compliance with presentation standards (6 points)	/30
5.1 Appropriate choice of clauses, operators, commands or parameters in database queries(6 points) 5.2 Correct handling of database data (6 points) 5.3 Proper programming of the conversion of data into information (6 points)5.4 Proper application of internationalization techniques (6 points) 5.5 Precise application of secure programming techniques (6 points)	/30
6.1 Correct manipulation of DOM objects (10 points) 6.2 Proper programming of interactions between the Web interface and the user (5 points) 6.3 Proper programming and integration of animations and widgets (5 points)	/20
Total :	/100