

GRADE 9-12 3D GRAPHICS AND INTRODUCTION TO PROGRAMMING FRAMEWORK

CONTENTS

OVERVIEW:	1
EXPECTATIONS:	1
ACTIVITIES/PROJECTS:	2
RESOURCES:	2
ASSESSMENTS:	2
PERFORMANCE INDICATORS:	2
CREATIVITY AND INNOVATION	2
COMMUNICATION AND COLLABORATION	2
RESEARCH AND INFORMATION FLUENCY	3
CRITICAL THINKING PROBLEM SOLVING AND DECISION MAKING	3
DIGITAL CITIZENSHIP	3
TECHNOLOGY OPERATIONS AND CONCEPTS	3

OVERVIEW:

Introduction to Programming/3D Design is divided in two sections, Introduction to Programming and 3D Design. Students will learn to code in a programming language. Concepts like variables, values, value types, classes and methods will be addressed. The learning experience is very “hands-on” so each student can explore the language, with the given concepts, solving a particular problem of interest. Java will be used as the working programming language to create projects of different types, like desktop and mobile applications.

Students will work with Blender to apply concepts of geometry, physics and computer design. From basic shapes, like cubes, spheres, torus or cylinders, they will get more advanced shapes, combining and transforming them. Students will also animate their creations using different virtual materials, lights and camera options to achieve the best results. Mathematical functions will be applied, not as simple abstractions but as a way of achieving results. Blender will be used to create some physics simulations (e.g., fluid, rigid body, force fields). Other rendering techniques like Freestyle will also be used.

EXPECTATIONS:

The student will be able to:

Design 3D models using an open source application.

Plan and create an animation movie using original models.

Mix royalties free music with video.
Prepare models to print with a 3D printer.
Develop computational solutions (develop simple programs).
Design, prototype and test a proposed solution.
Understand and manipulate abstract concepts.
Develop logical and critical thinking as well as experimental, investigative and problem-solving skills.
Know and understand appropriate methods and techniques.

ACTIVITIES/PROJECTS:

Small projects applying specific techniques.
Final project involving an animation of two minutes.
Print an object with a 3D printer.
Small applications solving specific problems.
Final project involving a solution for a complex problem.

RESOURCES:

Moodle course
Blender application (free)
Online resources:
www.blender.org
www.blenderguru.com
Individually assigned text book.
Eclipse Integrated Development Environment.

ASSESSMENTS:

Small exercises with specific techniques.
Final project.
Write small programs solving different problems.
Class presentations and discussions.
Written tests.
Develop and document a fully operational solution to a real problem.

PERFORMANCE INDICATORS:

CREATIVITY AND INNOVATION

Is able to apply existing knowledge to generate new ideas, products, or processes. DOK 3
Creates original works as a means of personal or group expression. DOK 3

COMMUNICATION AND COLLABORATION

Contributes to project teams to produce original works or solve problems. DOK 4
Writes programs with proper programming style to enhance the readability and functionality of the code by using meaningful descriptive identifiers, internal comments, white space, spacing, indentation, and a standardized program style. DOK 3
Uses Graphical User Interfaces (GUIs) to create interactive interfaces to acquire data from a user and display program results. DOK 4

RESEARCH AND INFORMATION FLUENCY

Uses a variety of resources, including other subject areas, together with various productivity tools to gather authentic data as a basis for individual and group programming projects. DOK 3

CRITICAL THINKING PROBLEM SOLVING AND DECISION MAKING

Create a console application. DOK 1 E

Create small applications with decisions and loops. DOK 2 E

Create applications with arrays of basic types. DOK 1 E

Create methods with arguments DOK 1 E

Define Java classes with “get” and “set” methods. DOK 1 E

Define class constructors. DOK 1 E

Define inherited classes. DOK 1

Override methods in subclasses. DOK 2

Define the “toString()” method to allow debugging and testing of applications. DOK 2 E

Use Java collections (ArrayList and LinkedList) to create applications. DOK 1

Design a GUI using JavaFX. DOK 4

Create a Java application using JavaFX GUIs. DOK 1

Use files to store and retrieve application data. DOK 2

Create programs capable of calculations with date and time. DOK 2

Designs and writes code for short programming tasks to demonstrate the use of operators for assignment, arithmetic, comparison, and Boolean combinations. DOK 3

Discuss and implements core features of structured programming languages, such as variables, operators, loops, decisions, assignment and modules. DOK 3

Debugs and solve problems using error messages, reference materials, language documentation, and effective strategies. DOK 3

Explores common algorithms, including finding greatest common divisor, finding the biggest number out of three, finding primes, making change, and finding the average. DOK 4

Selects the most appropriate algorithm for a defined problem. DOK 4

Develops sequential algorithms to solve non-branching and non-iterative problems. DOK 2

Develops algorithms to decision-making problems using branching control statements. DOK 2

DIGITAL CITIZENSHIP

Advocates and practice safe, legal and responsible use of information and technology. DOK 4

Exhibits a positive attitude toward using technology that supports collaboration, learning, and productivity. DOK 4

TECHNOLOGY OPERATIONS AND CONCEPTS

Create a simple model using scale, rotation and translation transformations. DOK 1E

Change viewports. DOK 1

Applies mirror modifiers when necessary. DOK 3

Alternate between wireframe and solid views when editing models. DOK 1

Create loop cuts in models. DOK 2

Subdivide objects when necessary. DOK 3

Extrude faces, lines and vertices editing objects. DOK 1

Use Blender cameras to achieve specific results. DOK 3

Join and separate objects. DOK 1

Use camera clipping to determine rendered objects. DOK 1

Use color harmonies to select colors. DOK 4

Apply the Screw modifier. DOK 2

Apply the Boolean modifier. DOK 2

Apply the Edge Split modifier. DOK 2

Create clouds. DOK 3

Use the Solidify modifier. DOK 3

Prepare objects to print. DOK 4

Describe the characteristics and applications of a collection. DOK 3
Describes the main components of a computer system (CPU, memory, main storage, I/O devices). DOK 2

Explain how computers are devices for executing programs via the use of programming languages.
DOK 2

Explain the use of the compiler. DOK 2