



Title: Geometry + Computer Visualization/Simulation

Length of Course: Full Year (2 semesters; 3 trimesters; 4 quarters)

Subject Area – Discipline: Mathematics (“c”) – Geometry

CTE Sector: Information and Communication Technologies

Pathway: Software and Systems Development

Grade/level(s): 10

Overview:

This course seeks to introduce students to a range of careers - including software development, computer programming, game design, digital fabrication - and methods that use computers to visualize geometric information necessary for product design. Using applied geometry, students create a final 3D-printed product that demonstrates the key role that geometry plays in a particular industry. Throughout the course, students will integrate geometric principles with computer-based programs and designs while learning to create and work in a technical environment that facilitates workplace efficiency and online communication. Students use information and communication technology (ICT) practices to share their geometry-based findings and projects with each other, their teachers, family and others. Upon completion of this course, students will be familiar with the many ways that computers can be used to simulate and model geometric concepts using computer programming and modeling, simulations and interactions, and three-dimensional printing.

This course will also have an **ongoing assignment:** to maintain an online portfolio or folder, either in a Google Drive, Dropbox, Wiki, or Blog, where students can store documents, assignments and products from the course.

Course Content:

Unit 1: Computer Programming: Building Blocks of Geometry; Area; Similarity with Circles and Triangles

In this unit students will learn basic computer programming principles, like functions and loops, to write their own visual representations of geometric concepts such as points, lines and polygons. They will understand how a computer processes information through algorithms, in order to efficiently produce a geometric output. After defining and analyzing systems and software requirements necessary for the development of their geometry-based computer programs (in Scratch or Python, as examples), students will write an interactive computer program that includes loops, functions, booleans, inputs and outputs, that will then demonstrate their understanding of fundamental properties of

polygons and geometric objects.

Assignment Summaries:

1. Introduction

Students will learn that computer programs can be used to facilitate and automate tasks, such as drawing a shape or plotting a line. Using a graphical programming interface, such as Scratch or Snap!, students will write simple programs that draw 2D geometric objects on the screen. Students will build foundational knowledge of basic geometric principles of points, lines and planes. As they build geometric objects on the screen, they will learn the fundamental properties of some polygons (including but not limited to triangles and quadrilaterals). Students will demonstrate their understanding of these properties as they program a computer to construct 2-dimensional regular polygons, using the number of angles and their measures as inputs.

- a. Students will begin by programming a digital 'pen' to draw four points as corners on a plane at given coordinates. Students will then programmatically construct lines which will lead to the construction of triangles and quadrilaterals using given points and angle rotations by the required degrees to create regular polygons.
- b. Next, students will learn about looping functions that automate the drawing of a triangle, a square and other regular polygons by repeating the drawing of each side and the rotation of the pen. The functions will reflect an understanding of the geometric construction and attributes of a regular polygon by including commands that require inputting the appropriate number of times that a line needs to be drawn and the appropriate number of degrees that the pen must turn in order to close the polygon.
- c. Students will then learn about keyboard inputs or prompts, such as number of sides from 3-12, and length of each side, to get valid inputs from a user. Finally, students will be able to write their own program that prompts users to enter the number of sides, and length of each side, and will then draw that polygon onto the screen with the correct angles and dimensions. This program requires an understanding of how to calculate the angles of a regular polygon given the number of sides. Extension: write a program that takes in a string value (a word) and converts that into a polygon of specific dimensions.

2. Areas of Convex Quadrilaterals and Triangles

- a. Students will write a program to calculate the area in square units for quadrilaterals. This activity will also teach students about concepts of variables and if/else statements used in computer programming. First, students will extend the program in activity one that draws a square to draw a rectangle with given length and width. Using Scratch or a syntactical language like Python or JavaScript, students will define two variables 'b' and 'h' for example, and another empty variable called 'area'. Prompting the user, the program will take the first input as the length, and the second as the width. The program will then output the Area of the figure either to the command-line, or output.

- b. Next, students write a program that calculates the area of the following polygons: triangle, parallelogram, trapezoid, rectangle, and rhombus. The program will prompt user to input a name of a polygon from the list, then using an if/else statement to re-prompt the user in case a trapezoid was chosen, to enter another base. If not a trapezoid, then the program will run a 'calculate area' function that will take the input of what shape was chosen, and apply the appropriate formula and return that calculated value to the user in a statement such as: "The area of your trapezoid is 56 sq. units". The goal of this assignment is to develop student understanding of the if/else statement, using a review of the basic area formulas. An extension of this program will actually draw the polygon to the screen AND calculate the area.
- c. Finally, the students will use a 10 pixel by 10 pixel square sprite in Scratch or Snap!, and programmatically figure out how many squares are necessary to fill in a larger given rectangle. The student will extend their knowledge of loops by including nested loops to solve the problem of filling the rectangle with the square tiles. The program should keep track of the amount of tiles that are used to fill the figure with an area variable, and then output or print to the screen the area in square units.

3. Circles

In this activity, students will create dynamic computer-based simulations to demonstrate their understanding of the similarity of all circles. Using a chosen programming language, students will first write a program that draws a circle after the user is prompted for a radius. Next, the students will explore different types of user interfaces for adjusting or changing the value of the variable, such as a slider, buttons or text boxes, and choose one method that works best to create a program that allows the user to dynamically change the size of the circle on a screen, by adjusting the interface. The program should automatically update as the user changes or updates the input via the interface. Extension: The simulation should then animate to display the circle's radius, center, diameter, and a tangent where the radius intersects the circle when the user interacts with other buttons or interface on the program.

4. Programming a Visualization of Similar Triangles

In this exercise, students will write a program that illustrates how a triangle is dilated through a given center and scale factor. In the program, students will first construct a non-equilateral triangle by applying the properties of triangles to ensure that, given a set of 3 side lengths and 3 angle measures, the resulting polygon will form a triangle. Then, using user-provided values for a center of dilation and scale factor via a programmed prompt, the program should generate the transformed triangle. Finally, the program should include a check which incorporates the definition of similarity to verify that the two triangles are similar.

Unit Two - Game Development: More on Transformations, Congruent Triangles

In this unit, students develop a computer game that focuses on the rigid motion of basic geometric shapes, or transformations. To build knowledge of Game development pre-production workflow, students learn about storyboarding and decision trees. Students also acquire vocabulary and concepts associated with the rotation, reflection,

dilation, and translation of objects. They will explore how the 2-D (x,y) coordinate system is used in computer games to simulate placement and movement of sprites on a screen. Additionally, students gain their first exposure to the role that geometry plays in three-dimensional character modeling.

Assignment Summaries:

1. Shifting Shapes

In this activity students articulate and share how geometry and geometric concepts are used in the animation, drawing, and player interaction of a game on the coordinate plane. First students play a [premade game developed with the Scratch framework](#) that involves moving, rotating and/or placing objects. A rudimentary version of Tetris would be a perfect example. This is followed by investigation of how geometry is embedded in game mechanics, focusing on what is being animated, speculation on how that might be achieved, and how the computer program ‘understands’ the relationship between the image, or sprite, on the screen, with the instructions it needs to perform its movements. Students then:

- a. Students research how polygons, transformations, and coordinate planes are integral components of 2D game mechanics. The object is for students to understand that objects, or sprites are represented and animated as pixels, or collections of pixels on a screen, and how these points can be translated, rotated, and transformed to simulate movement and change on a plane. Students should also explore how triangles specifically are used in character modeling for 3D games. Students then present their finding to their classmates, in a slideshow, Prezi, or other digital format.
- b. In order to make the connection between digital inputs in games and geometric transformations, students create a chart detailing the following transformations: rotations, translations, reflections, and how they translate into the sample game mechanics, (eg. when the left arrow is pressed, the sprite rotates to the left, and preserves its shape and size; when the “A” key is pressed the sprite translates across the x-axis of the plane).

Geometric transformation in Marble Mayhem

Sprite: player marble control: keyboard

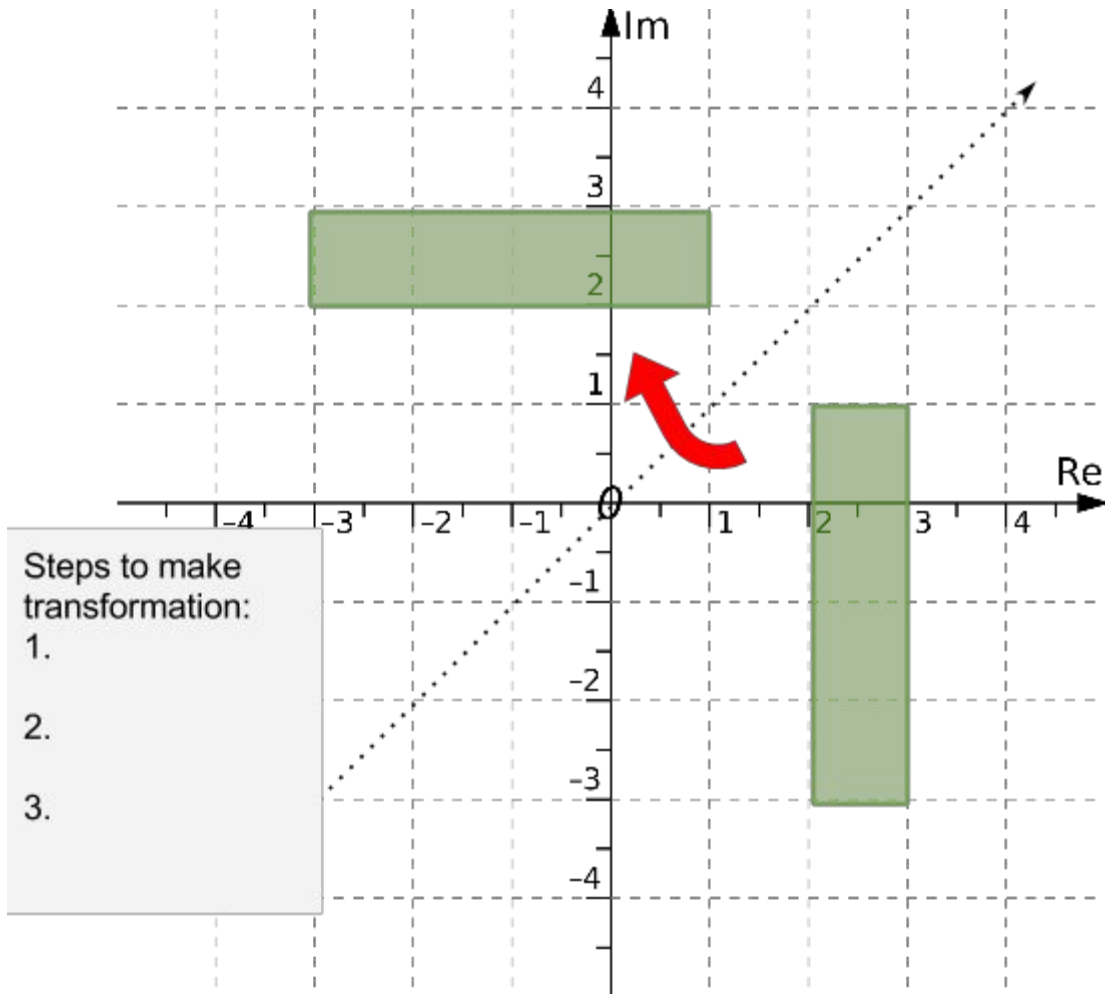
control: sprite action: transformation:	control: sprite action: transformation:
control: sprite action: transformation:	control: sprite action: transformation:
control: sprite action: transformation:	control: sprite action: transformation:



2. Representing And Combining Transformations

In this assignment, students learn that object transformations, such as “Reflect over the x-axis” or “Rotated through 180-degrees about the origin,” involve moving individual points on a plane while preserving their congruence. This activity continues to build on students’ understanding of rigid motion while introducing the skill of plotting shapes on coordinate planes and using rigid motion to prove congruence.

- a. First, students will make their own non-animated representations of geometric transformations using a diagram tool such as Google Drawings, OpenSCAD, or SketchUp. The goal for this section is for students to be aware of how they can use the “Flip horizontal” “Flip Vertical”, dilate, and rotate functions of the image authoring software to represent and share their understanding of polygon transformations. Students should be able to make these in teams or groups, where each student represents one transformation (rotation, dilation, etc.) using the selected tool. Another constraint should be that reflection simulations should be over lines of symmetry that are not horizontal or vertical.



- b. **Game design:** Students work together in pairs to conceptualize and describe a *new* computer game that involves some form of rigid motion. The game concepts do not need to be limited to a tetris-like game of shapes, but they need to be

expressed in terms of the images/sprites and the movement in terms of translation, reflection, rotation, and dilation. Students will write a game design proposal which will include information on what the object of the game is, how the player controls it, and how rigid motion is part of the game. Students then share their proposals, and provide each other with feedback about the potential effectiveness of the games and their proper incorporation of geometric principles.

- c. **Storyboards:** Students create a storyboard describing the essential elements, plot, flow, and functions of the game. Students create a design specification document to include interface and delivery choices (communicated visually as [decision trees](#)), rules of play, navigation functionality and scoring. This will serve two objects, to demonstrate how games are designed in the pre-production conceptual stage, and to scaffold the geometric proofs.
- d. **Proofs as storyboards:** Students learn that Geometric proofs are a way of storyboarding a geometric problem, in that given a set condition, a logical conclusion can be reached. Once students have created their game design storyboard in the previous assignment, they will then translate that approach as a scaffold for becoming familiar with a variety of proof forms. Students will prove a geometric theorem using the format of a storyboard, and then translate it to a flow or other type of geometric proof.

3. Get Gaming

Building on the previous assignment where students plan and storyboard their transformation-based game, students then use Scratch, Snap! or a similar application to develop the game into a working prototype. In it, transformations will be an essential part of the game mechanics. For example a team could build a game where through dilations and transformations, a shape moves through a maze, or a game where dilations and rotations are used to match falling shapes. As prototypes, these games need not be fully fleshed out, with multiple levels, player options, etc. But they should convey that a student understands how different transformations can be use in gameplay. If possible, teams publish their games online and solicit feedback from friends and family. As a final activity, students write a reflection about changes they would make to the game play or game design if they had the chance to do the project again. It should also include an explanation of how transformations were used in the game.

Unit Three - 3D Modeling

In this unit students learn fundamental trigonometric principles and apply this knowledge to design and model a wheelchair ramp using a three-dimensional technical drawing program such as SketchUp, Geogebra, Autodesk Inventor or Solidworks. This task presents a real-world challenge that involves foundational understanding of right triangles and the ways in which computer simulation can save time and money with respect to architecture, construction and manufacturing. Students use a computer application to revolve and/or extrude two-dimensional objects, thus creating three-dimensional objects with volume. Specifically, students will create ramp cross-sections of varying specifications, and extrude their designs to generate

three-dimensional models that can be used to compare the cost of different ramp styles and configurations. Students review basic algebra to solve length-area problems with ratio and proportions as they create similar polygons of different dimensions.

Assignment Summaries:

1. Some Triangles Are Special

In this assignment, students will create triangles using a 2D or 3D CAD program, such as [Geogebra](#), in order to understand the characteristics of special right triangles. Students explore the idea that by similarity, side ratios in right triangles are properties of the angles in the triangle, leading to definitions of trigonometric ratios for acute angles. Students make predictions about the triangle side lengths, given specific angle measures (30-60-90 and 45-45-90), then create a series of technical drawings of triangles. Through the program's return of corresponding side lengths, students can verify their predictions and use the resulting complete set of triangle measurements (3 sides, 3 angles) to derive the trigonometric ratios for special right triangles.

2. Ramp Up Your Knowledge

In this assignment, students apply their developing understanding of right triangles and trigonometry to begin the design of a wheelchair ramp (inclined plane) in a specified location. Students are presented with a plot of land, property boundaries and a structure with a floor above ground level.

- a. Students apply their derivations of the Law of Sines and the Law of Cosines to find unknown measurements in right and non-right triangles, the results of which specify the shape of potential ramp designs. Students compare their different designs, taking into consideration constraints such as the ADA's specification of a maximum rise and run, in order to determine the best design for the specific context. Students give and get feedback on their design, (which they have documented using a presentation tool or software), including feedback on safety and location constraints of the wheelchair.
- b. After designing several possible ramps, students present one of their designs and explain why they choose that particular ramp to recommend to the client. The product is a slide-pitch presentation that discusses the problem, and the ramp that will best meet the needs of the user, detailing the angles, location, and constraints.

3. Model What You See

In this assignment, students learn about extruding circles and rectangles into cylinders and rectangular prisms through rotation or consideration of cross-sections by creating three-dimensional representations of everyday objects.

- a. First students are presented with realia or digital photographs that include representations of three-dimensional objects. Students pick an object and use a computer application to create a two-dimensional sketch of one of the object's faces or cross sections. Students learn to extrude or rotate their two-dimensional objects, thus creating three-dimensional models of the items in the photographs.

Each student must use both extrusion and rotation (for a single object or multiple objects) in order to demonstrate their understanding of the relationship between the 2D and 3D objects.

4. Into The Third Dimension

The culminating assignment for the unit builds on student understanding of the generation of 3D objects through extrusion and rotation in the wheelchair ramp application. In preparation for printing out scale models of potential ramp configurations, students extrude drawings of two-dimensional triangles to create triangular prisms. Students discuss the merits (based on industry considerations such as material cost, available space, construction costs) of creating a ramp with multiple switchbacks of gradual rise or a single long ramp with a steeper angle. Students print out technical drawings of their ramps using standard isometric planes. This project will be picked up again in Unit 4 when students use 3D printers to study the differences in volume of various ramp shapes/configurations.

Unit Four - 3D Printing

In this unit, students review, apply and expand their knowledge of algebra and geometry as they learn the basics of 3D printing for manufacturing, given certain design constraints such as desired volume and minimal use of materials. They will calculate and model similar, regular polyhedrons by applying concepts of scale factor to create larger and smaller packages for certain objects. As they work with polyhedrons on paper and by computer simulation, they will practice visualizing the relationships between two-dimensional and three-dimensional objects. They will then measure the volume of those three-dimensional objects they print.

Assignment Summaries:

1. Calculating volume for manufacturing under given constraints

Students apply the concepts of volume studied in Unit 3 to the manufacturing of food containers. Students are given various three-dimensional objects and asked to calculate their volume. They then determine the most efficient shape to hold a given amount of fluid as is done in the manufacturing of canned food. They will assess which shapes to print based on the goals of minimizing production costs and maximizing the volume of a given container. For example, students will discover, through the design process, that the surface area of a cube is greater than the surface area of a cylinder of the same volume, and therefore requires more material and higher production costs. The final assignment will be a proposal for container design based on the provided constraints, which includes quantitative arguments, in the form of calculations, to show that the constraints for a given volume and minimal area have been met.

2. Scaling up: The effect of a scale factor on 3-dimensional objects

Students will determine how a scale factor affects the length, area, and volume of a three-dimensional object by reproducing their proposed container from the previous assignment at different dimensions, as a manufacturing company would produce similar packages at different sizes to hold different amounts of a similar product. They will demonstrate understanding of the exponential effects of the scale factor on the

different dimensions by designing and presenting their modified package on Google Sketch-up or a similar CAD program.

3. Conceptualizing fabrication: Visualizing relationships between 2D and 3D objects

In order to prepare for the 3D printing of their proposed package, students will produce blueprints of their regular solid using a compass and straightedge or CAD software. They will demonstrate the relationship between this “net” and their desired 3D package by gluing up the edges of the cut-outs to produce the solid of what they will print.

4. Fabricating regular solids by 3D printing

Students will finalize their 3D design process by using a CAD software such as Google SketchUp to produce the file (“instruction”) for the 3D printer. The final product will be a 3D printed version of their object as well as a technical presentation which describes the process followed, including how geometry was used as a tool in design, leading up to this final fabrication on the 3D printer.

Course Materials:

ICT + Geometry Assignment Resources

Texts:

[The Use of Mathematics in Computer Games Stage: 5 Article by Dan Goodman Published May 2000, February 2011.](#)
[Math animations using JavaScript](#)

Interactive Lessons:

[Geometry in Computer Games? An Exploration of Tessellations Used in neXtu](#)
[Khan Academy: transformations](#)
[Intro to Drawing with JavaScript: Khan Academy](#)

Videos:

[How is geometry used in Video games?](#)

Sites:

[Geometry in Action](#)
[GeoGebra](#)
[Autodesk Fusion](#)
[SketchUp: 3D for Everyone](#)
[Tinkercad | Create 3D digital designs with online CAD](#)
[OpenSCAD](#)
[Google Drawings](#)
[NCTM - Illuminations - Geometry 9-12](#)

Presentations:

[Use of geometry in Video game design](#)

Geometry games:

[Tangled Web](#)

[Transtar](#)

3D printing:

[Teaching Geometry with Google Sketchup](#)

Scratch:

[Getting started with Scratch : Drawing 2D shapes](#)

[Scratch Tetris](#)

[Teaching geometry w/ Scratch](#)

[Scratch and Geometry](#)

[Math Projects in scratch](#)

Simulations/interactives:

[Overlaid](#)

[Dilation of a polygon](#)