



# SCHOOL DISTRICT OF THE CHATHAMS CURRICULUM PROFILE TEMPLATE



**CONTENT AREA(S):** Design & Technology

**GRADE LEVEL(S):** 9-12

**COURSE:** Animation and Movement

**TIME FRAME:** Semester

## **I. Course Overview**

Animation and Movement provides an introduction to the field of computer animation programming through the utilization of humanoid robots and block based programming environments. The course emphasizes problem solving in a real-world context and computational thinking utilizing both gaming and robotics platforms. This class is an alternative for those students who have an interest in computer science but are not interested in the Advanced Placement track.

## **II. Units of Study**

Unit 1: Introduction to Block Based Programming

Unit 2: Creating an Animation

Unit 3: Using built-in functions and expressions

Unit 4: Simple Control Structures

Unit 5: Writing methods and using parameters

Unit 6: Interactive Programming

## **III. Essential Questions** *(The open-ended, provocative questions that help frame inquiry)*

### **Unit 1:** Introduction to Block Based Programming

- What are the types of animations?
- How do we navigate a block based language?
- What are objects and how do we add them to the world?
- How can we manipulate an object's 'parts'?
- What are classes vs. instances?

### **Unit 2:** Creating an Animation

- What are the steps to solving a problem (or writing a program)?
- What is the storyboard design technique?
- What are action blocks?
- What are methods?
- What are arguments?
- What is nesting?
- What is the incremental development process?
- What are comments and how do we use them?



# SCHOOL DISTRICT OF THE CHATHAMS CURRICULUM PROFILE TEMPLATE



## **Unit 3:** Using built-in functions and expressions

- What is an instruction (method)?
- What is a control structure?
- What is a function?
- What is an expression?
- How do we implement instructions, control structures, functions and expressions in a block based programming language?

## **Unit 4:** Simple Control Structures

- What is a conditional execution (if/else)?
- What is a boolean?
- How can we use math operators to create our own boolean statement?
- How can we create a repetition (loop)?

## **Unit 5:** Writing methods and using parameters

- What is the difference between a class, object and method?
- How do we write our own methods?
- What is stepwise refinement?
- What are parameters?
- How do we implement methods with parameters?

## **Unit 6:** Interactive Programing

- What is flow of control and how do we alter it?
- What is an input and how can a user provide input?
- What is an event and how do we 'handle' them?

## **IV. Learning Objectives**

Objectives align with the NJSLS Standard 8.2 Strand E. It incorporates concepts of mathematical processes of problem solving in Mathematics and the content areas of Technological Literacy and 21st Century Life and Careers.

## **V. Instructional Materials**

Alice or equivalent block based programming language with associated text. NAO Robot.

## **Supplementary Materials**

- Online documentation for block based programming language
- Online documentation for the NAO Robot



## **VI. Key Performance and Benchmark Tasks**

### **Unit 1: Introduction to Block Based Programming**

- Understand what a block based programming language is.
- Understand what animations are.
- Understand what 'objects' are and their 'parts'.
- Differentiate between classes and instances.
- Create a 'scene'.

### **Unit 2: Creating an Animation**

- Understand how to implement the storyboard design technique to write a program.
- Differentiate between action blocks, methods and arguments.
- Understand how to nest blocks.
- Understand the value of commenting in the design process.
- Create a basic animation.

### **Unit 3: Using built-in functions and expressions**

- Understand what an instruction (method) is.
- Differentiate between a control structure, function and expression.
- Create an animation implementing control structures, functions and expressions.

### **Unit 4: Simple control structures**

- Understand what a conditional execution is (if/else).
- Understand how to use booleans and construct booleans from math operators.
- Understand the flow of a loop.
- Create an animation implementing if and loop conditionals.

### **Unit 5: Writing methods and using parameters**

- Differentiate between a class object and method.
- Understand the stepwise refinement process.
- Implement the stepwise refinement process to create basic methods and methods with parameters.
- Create animations involving user defined methods.

### **Unit 6: Interactive Programming**

- Understand how to alter flow of control.
- Understand various types of user input and ways of processing those inputs.
- Create fully responsive animations.