

## **Middle School Curriculum Proposal**

PROPOSED COURSE NAME: Creative Computer Programming

OLD COURSE NAME (if applicable) N/A

SUBMITTED BY: Michael Wesbecher

Underline the Appropriate Item: New Course/Major revision to an existing course

Implementation Date: Fall 2014

Person(s) who will write the Curriculum: Michael Wesbecher

Respond to the following:

Duration: Semester

Open to: All students in grades 7/8

Prerequisites: NONE

Pass/Fail Option: NO

Level(s): Grades 7 and 8

### **Where does this course fit into department sequence?**

This course would prepare students for concepts that will be built upon in Web Design/HTML programming courses that are currently offered at the high school level. This course is designed to encourage a wider interest in computer programming as a career.

### **What is the potential impact of this course on other department offerings?**

This course allows students to experience more than one computer course option at the Middle School level. It helps open up other possibilities for students to broaden their experiences and talents.

### **Course Description (Consult the Program of Studies as a model)**

This course will consist of two parts. Part I will use SCRATCH animation software to create a variety of computer applications in the areas of the arts, storytelling, and game design. Part II will use Alice programming software to create virtual worlds in which characters that students design move, talk, and interact with each other.

### **Special Notes (i.e., extended periods, team teaching, etc.)**

None

### **RATIONALE, OBJECTIVES, STRATEGIES**

## **1. Course Rationale**

### **a. What is the evidence of student need for this course?**

As the name of this course implies, Creative Computer Programming will teach computer programming. Typically, when one hears the phrase “computer programming”, it brings to mind a variety of negative images, including hours spent sitting and typing code in a text editing program and many hours of frustration spent de-bugging this same code. As a result, many young people have avoided computer programming courses. Thus, they have missed out on the satisfaction and fun that can come from being a creator of interactive media rather than just a consumer of it.

Over the past few years, a new model of teaching computer programming has emerged. This model uses a graphic interface in which you put blocks or puzzle pieces together in order to write computer programs. These blocks or puzzle pieces fit together like LEGO blocks. As these blocks are put together, the interface writes the code and the creator can instantly view the results of their design choices through the use of a preview window that is paired with the design interface.

This course will utilize two of the best such interfaces currently available: SCRATCH and Alice. The use of these interfaces will allow students to create works that are personally meaningful to them. It will also allow students to work collaboratively with others in the creation of computer applications and animations in the same way that Google Documents currently allows students to work jointly in the creation of a report or presentation.

Current literature also suggests that students achieve their best results when reflection time is built into the curriculum. As a result, this course will also include built in time for students to review and rethink their designs.

This combination of interactivity, collaboration, and reflection represents something new in technology education at North Boone and should result in a very intellectually stimulating and enriching experience for students at our middle school.

### **b. What departmental needs/goals would this course meet?**

As stated previously, the purpose of this course is to develop an interest in computer programming. It is my intention that this will have the result of improving enrollments in Web Design and HTML programming courses at the high school level and for computer courses in general at the high school level.

### **c. What institutional needs/goals would this course meet?**

As stated previously, this course will address the need for a greater variety in exploratory course offerings at North Boone Middle School.

## **2. Course Objectives: What will each student learn and know?**

The following programming concepts will be taught in this course.

- Sequencing and looping
- The design practices of being incremental and iterative
- Parallel programming
- Event programming
- The design practices of reusing and remixing

- Programming using conditionals
- Programming using operators
- The design practices of testing and debugging
- Use of methods and properties in programming
- Use of billboards, sound, and 3D objects in the creation of virtual environments
- Use of parameters in programming

See attached proposed syllabus for definitions of terminology.

### **3. What types of instructional strategies will be used in the class?**

As described previously, this course will utilize the latest technology available for learning programming. It consists of a graphic interface in which users put puzzle pieces or blocks together that work like LEGO blocks. Each block represents a different programming command. As the blocks are put together, the interface generates the code for the program. The results of the programming choices made can be viewed in a preview window which is also part of the graphic interface.

Students will develop programs through the lens of the arts, digital storytelling, and games. A number of the programs will be developed collaboratively. In addition, reflection time will be built into the process so that students have the ability to review what went right and what went wrong. It is also hoped that this will serve as a time to rethink designs and gain input from others to help improve future programs.

### **4. What types of assessment procedures will be used? How will the assessment be used to guide instruction?**

As stated already, the programs that students develop will be done through the lens of the arts, digital storytelling, and game design. The general structure of the assessments will be to provide students with a programming problem that must be solved. Students will then be given instruction in the principles and concepts that will be needed to solve the problem. Afterward, students will be asked to apply these principles and concepts in order to solve the problem.

Final products will be evaluated by the degree to which the programs developed solve the stated design problem. Emphasis will also be placed on the structure of the program. Thus, programs which follow established programming design structure will be evaluated more favorably than those that are haphazardly put together. This is important because in the real-world of computer programming, teams of people work to create and maintain programs over a period of years. Thus, the people who later work on the program need to be able to understand the work of earlier teams.

### **5. List resources needed and approximate costs to implement this course. Include staff development, materials, textbooks, additional staff, equipment, technology resources, facilities, and fees.**

SCRATCH – Free download from Massachusetts Institute of Technology available at <http://scratch.mit.edu/>

Alice – Free download from Carnegie-Mellon University available at <http://www.alice.org/index.php>.

**Include any additional information pertinent to your course.**

See attached syllabus for more in-depth description of units of study and of SCRATCH and Alice.