

**ARTICULATION AGREEMENT**

**DATE DRAFTED:** June 28, 2020

**VALID ACADEMIC YEARS:** 2020-21 & 2021-22

**LMC COURSE:** COMSC-110 Introduction to Gaming

**HIGH SCHOOL COURSE:** Coding and Gaming

**School:** Mt. Diablo High School

**Address:** 2450 Grant St., Concord, CA 94520

- A. COLLEGE COURSE DESCRIPTION:** An introduction to electronic game development with an emphasis on computer animation and programming. Computer animation will introduce the student to an overview of character modeling, rendering, animation, illustrations, storyboarding, and game design. An overview of computer programming will introduce the student to gaming structures, animation techniques, design fundamentals and programming options. Both of these concepts are very important for students entering the Gaming industry.
- B. UNITS: 3**
- C. PRE-REQUISITES: NA**
- D. REQUIRED CONTENT FOR ARTICULATION:**

**Unit 1: Programming Languages**

In this unit students will learn about different programming languages and the ways in which languages evolved from one another. They will learn the history of programming languages and understand similarities and differences between different styles of languages: scripting, procedural, and object-oriented programming, as well as recognize the benefits and limitations of text-based and languages compared to drop-and-drop or WYSIWYG development environments. Students will be introduced to a text-based language such as C++ or a drag-and-drop programming environment (Scratch, Snap!, Blockly, etc.). By the end of the unit the students will know how to create a code project, add files to the project, write, compile, run, and debug a simple program.

**Unit 2: Data Types and Operators**

In this unit students will learn about data types and operators. Students will learn how to declare variables prior to using them in a program. They will learn about basic data types such as integers, doubles, Boolean, and strings, and they will learn about values that are appropriate for each type and recognize the consequences of confusing them. They will understand the difference between local and global variables. They will learn that certain characters have special meaning and the need for character escape sequences. The unit will introduce students to relational and logical operators, truth tables, and how the compiler interprets them in order to perform logic and mathematical operations.

**Unit 3: Control Statements and the Order of a Program**

In this unit students will learn how a program executes based off the order of statements and learn to generate and follow flow charts. They will also learn how to control the order of their program by using

if, if else, if else if else statements. They will also learn how to use for loops, while loops, do-while loops, and switch case statements.

#### **Unit 4: Arrays and Strings**

In this unit students will learn that Arrays and Strings are a collection of variables of the same data type and learn to recognize scenarios in which the use of an array would be desirable. They will learn the necessary steps to create, populate, modify, and access a one dimensional array. Students will learn how to manipulate data types and convert values between them in order to combine data across types. Students will use concatenation and in-string searches, set variables from user keyboard input, and respond to user feedback to control program workflow.

#### **Unit 5: Functions**

In this unit students will learn about functions and subroutines, the building blocks of most programming languages. Students will learn that functions and subroutines are a collection of statements that can be called from anywhere in a program to execute a certain task, and will explore their value in organizing code structure and allowing for more efficient code reuse. Students will learn about the scope of local and global variables in a function. They will learn about function parameters/arguments, return types and function prototypes.

#### **Unit 6: Classes and Object Oriented Programming**

In this unit students will get an introduction to classes and object oriented programming. Students learn what an object is, how a class defines the type, and how the object is the variable. They will learn how a class is similar to a data structure and that classes have access specifiers such as private, protected, and public members.

#### **Unit 7: Gaming Technical Skills**

In this unit students will learn how create games in a drag and drop development environment. They will program by using advanced coding methodologies (variables, functions, arrays, loops). They will learn how to follow technical and increasingly complex programming instructions in order and detail. Students will program original game projects and become familiar and competent in using professional 2D game development engines such as Game Maker Pro, or Construct 2/3. They will learn how to work with key frame animations, sprites, layers, backgrounds, events and behaviors. They will learn how open files, save files, create and program original games, integrate separate files into a final game project, create and edit audio sound effects & music, and publish games to the desktop, web, or mobile devices.

#### **Unit 8: Creative, Innovative, and Critical Thinking**

In this unit students will engage in several larger scale projects to learn how to create an original game based off of the knowledge they gained in the previous units. Students will work both independently and in teams of three to four students. They will use the engineering design cycle within the development team to achieve directive of creating and original game. The students will use creativity and critical thinking, and learn how to communicate and collaborate with each other in a production team environment.

## **E. REQUIRED COMPETENCIES (PERFORMANCE OBJECTIVES) FOR ARTICULATION**

Upon completion of the Coding and Gaming course, students will demonstrate skill in using industry-standard development environments to design and program 2D and 3D games as well as an understanding of computer science concepts that are translatable to many different programming environments and languages. Thus students who take this course are beginning to explore interactive design career pathways such as video game and software application design, website architecture and webpage design, mobile application development, and technical project management. All students will become comfortable and fluent working on computers and with technology, critical for any future career.

### **Big Ideas Explored:**

*Introduction to Programming Languages*

*Data Types and Operators*

*Control Statements and the Order of a Program*

*Arrays and Strings*

*Functions and Parameters*

*Classes and Object Oriented Programming*

*Audio and Visual Components*

*Physics and Math in Game Design*

*Game Genres, Conventions, and Tropes*

*What is Fun?*

*Introduction to 2D Game Design with Construct 2/3*

*Introduction to 3D Game Design with Unity3d*

*Introduction to 3d Modeling with Blender/Maya*

*Storyboarding, Project Planning, and Time Management*

*Asset Management, Compilation, and Release*

*Testing, Debugging, Critique, and Iteration*

## **F. METHODS FOR END OF COURSE ASSESSMENT:**

A final project is designed so the students can demonstrate they have mastered the learning objectives for a coding and gaming course. In this unit students will learn how to create an original game based off of the knowledge they gained in the previous units. They will use the engineering design cycle within the development team to achieve the directive of ideating and producing an original video game. A game proposal will be required along with a game submission, and intermittent status updates will be required over the course of the project. After successful completion of the project the students will present their games to the class, evaluate their classmates' work as well as perform team evaluations in a post-project reflection, and complete a course survey.

### **Participation:**

Each day students are expected to participate during instruction. Students are encouraged to follow along in guided practice and to ask/answer question. Keeping track of participation holds students accountable for learning on a day to day basis and makes contribution a part of the classroom environment.

**Presentations:**

Students present their research findings, inspiration, or code product to the rest of the class for peer feedback and teacher feedback. This occurs at least once per unit-- usually as the culmination of their learning. Students are graded with a rubric based on communication skills: public speaking, visual presentation and depth of understanding of the topic. This method holds students accountable for both individual and group work and forces students to take ownership of their work in front of their peers. Two more substantial end-of-semester projects provide higher stakes: these presentations are made to a mixture of family, community members, and professional judges.

**Written Assignments:**

Students are consistently asked to demonstrate their understanding in written format. These range from small daily assignments (warm ups, exit cards) to more involved assignments (research essays, written critiques, written self-evaluations). Students are graded with a rubric based on their ability to meet the goal of the assignment, which range from persuasion, objective criticism, informed analysis and reflection.

**Individual Assignments:**

Throughout the course students are given daily or weekly individual assignments to demonstrate their understanding of a particular programming or design skill. Many of the assignments happen in class but some require work outside of class. Students are graded with a rubric based on their ability to meet the goal of the assignment. These assessments enable students to explore tools and strategies on their own and sharpen their creative problem solving skills. In other words, these assignments allow students to discover learning and problem solving on their own.

**Group Assignments:**

These are assignments and projects that students work on in teams of two to four members. Each member will be expected to take on a specific role that complements their strengths. These projects enable students to demonstrate teamwork, communication, leadership, and many other CTE goals.

**Project Based Learning:**

A student and group-based approach that focuses on creative problem solving. The course consistently asks students to think of solutions to real world problems. As an assessment method, project based learning tests students in their capacity to take on real world roles in this CTE field. That is, these assessments measure not only student talent but also their capacity to take on collaborative career roles. Can students become effective leaders? Can they push themselves to work with others? Can they complete difficult work under competitive pressure?

**Grading Scale:**

Grades will be assigned using the following scale:

90% - 100%..... A

80% - 89%..... B

70% - 79%..... C

60% - 69%..... D

#### **G. PROCEDURES AND/OR CRITERIA FOR COURSE ARTICULATION:**

1. Complete the Coding & Gaming course at Mt. Diablo High School with a grade of “B” or better.
2. Receive a “B” or better on the agreed upon college/high school final exam procedure.
3. Be recommended for credit by the high school teacher.
4. Apply for admission at Los Medanos College.
5. Register for CATEMA for electronic recommendation of college credit **within the academic year in which credit was earned.**
6. Upon completion of the above, the student will receive on his/her LMC and CCCCD (California Community College District) transcript the units of credit for LMC’s COMSC-110 course.
7. College transcripts will reflect the **FINAL EXAM GRADE** earned and will be notated as \*Credit by Exam.

#### *\*Distance Learning Circumstances:*

*Final Exam “Procedure” will still need to be fulfilled whether the high school class meets in person or moves to a distance learning platform. If the high school class moves to an online learning environment, all efforts will be made to enable students to earn college credit, however due to circumstances beyond the high school/college control, course content may not be able to be completed in order to fulfill the articulation agreement requirements.*

#### **H. TEXTBOOKS OR OTHER SUPPORTING MATERIALS**

- C++ Programming From Problem Analysis to Program Design, D.S. Malik, Course Technology, Cengage Learning, 4th Edition / 2009
- A Theory of Fun for Game Design, Raph Koster, O’Reilly Media, 2<sup>nd</sup> Edition/ 2013

## ARTICULATION AGREEMENT

DATE DRAFTED: June 28, 2020

VALID ACADEMIC YEARS: 2020-21 & 2021-22

**LMC COURSE:** COMSC-110 Introduction to Gaming

**HIGH SCHOOL COURSE:** Coding and Gaming

**School:** Mt. Diablo High School

**Address:** 2450 Grant St., Concord, CA 94520

### COLLEGE SIGNATURES

\_\_\_\_\_  
Natalie Hannum  
LMC Vice President of Instruction

\_\_\_\_\_  
Date

\_\_\_\_\_  
Ryan Pedersen  
LMC Dean of Mathematics & Sciences

\_\_\_\_\_  
Date

\_\_\_\_\_  
Louis Giambattista  
LMC Computer Science Department Chair/Faculty

\_\_\_\_\_  
Date

\_\_\_\_\_  
Karen Stanton  
LMC Faculty

\_\_\_\_\_  
Date

### HIGH SCHOOL/ROP/DISTRICT SIGNATURES

\_\_\_\_\_  
Lorne Barbosa  
Principal, Mt. Diablo High School

\_\_\_\_\_  
Date

\_\_\_\_\_  
Dr. Robert Martinez  
MDUSD Superintendent

\_\_\_\_\_  
Date

\_\_\_\_\_  
Josie Kirkland  
Faculty, Mt. Diablo High School

\_\_\_\_\_  
Date