An Applied Computer Science Curriculum

Teaching Students a Needed Skill

Andrew Sweeney

George Mason University

© 2016 Andrew David Sweeney All Rights Reserved

ABSTRACT

Within the next couple of years, all public schools across the country will be required to incorporate a computer science curriculum into the overall K-12 curriculum next to math, english, social studies, and science. The purpose of this article is to propose a curriculum that will fulfill the federal and/or state requirements. The proposed curriculum is split into two distinct classes, with one class (to be taken during junior year of high school) being the introductory computer science class, and the other being the more advanced class (to be taken during senior year of high school). Each class will utilize a game design framework in order to teach computer science ideals and practices.

© 2016 Andrew David Sweeney All Rights Reserved

Earlier this year, President Barack Obama announced a brand new Computer Science Initiative during his State of the Union address. This initiative allocated over $4 billion in order to increase access to Computer Science courses in K-12 educational programs in the United States.[1] This initiative was brought about mainly because of how the world has come to rely on technology and computers. Children need to be taught how to use and manipulate computers if they want to excel in the future. While the initiative had great intentions, many problems have come forth that seems to have school systems worried.

While the President's Computer Science Initiative is helping fund Computer Science classes offered in the K-12 realm, one of the biggest problems that a majority of school systems will be facing is how to teach it. A majority of school systems already have that issue. According to a Gallup research study, *Searching for Computer Science: Access and Barriers in K-12 Education*, 75% of the school principals surveyed said that their schools do not offer courses with computer programming. [2] That means that only the remaining 25% of schools offer some sort of Computer Science course, with only half of those classes actually involving computer programming. [2]

Alongside the issue of schools not offering some version of a Computer Science course, in 19 states, a Computer Science course does not provide credit towards High School graduation credit. The ACM Running on Empty report was the first to release the details of this issue in 2013, where Computer Science courses didn't give credit in 37 states. [3] The report is updated yearly as the movement to get Computer Science into K-12 schools grows.

© 2016 Andrew David Sweeney All Rights Reserved

The reasons for schools to offer Computer Science courses have been manifesting themselves for years. The current job market is one of the largest reasons why students need to learn programming early on. According to the Bureau of Labor Statistics Employment Projections, the number of new programming jobs will reach the half a million mark sometime between 2014 and 2024. [4] That sounds like great news, with a lot of great job opportunites, but when looking at the number of people actually qualified to fill those jobs, it tells a different story. Information from the National Center for Education Statistics IPEDS Completions Survey shows that only 8% of all graduates graduate with a Computer Science degree. Looking at the raw numbers, that only means 42,969 students graduate with a Computer Science degree, far below the projected number of jobs. [5] With these numbers alone, it is a good reason to start offering Computer Science courses in K-12 schools, as it will allow students to garner an interest in programming and computers earlier, possibly allowing them to fill in those future job vacancies.

Fighting the good fight to get Computer Science into K-12 schools is the non-profit organization Code.org. [6] Code.org was formed in 2013 and ever since has been dedicated to expanding acces to Computer Science courses by raising awareness and holding events in order to get people involved. The organization has been working hard to get Computer Science courses to be a part of the core K-12 curriculum. It was because of their efforts, President Obama attended one of their ‘Hour of Code’ events, which later lead to the President announcing the initiative for Computer Science. So far, Code.org has inspired millions of students through the Hour of Code events, helped set up policies in over 20 states to support Computer Science

© 2016 Andrew David Sweeney All Rights Reserved

courses, and have even partnered with the 120 of the largest school districts in the United States to add Computer Science courses to their curriculum. [7]

While the organization Code.org offers their own custom created courses for teaching programming, there should be some sort of standard curriculum created for K-12 schools. Over the rest of this article, a newly proposed Computer Science curriculum will be detailed. It should be used as a guideline for setting up Computer Science courses in the future. It should be noted that this curriculum was crafted primarily for high schools. Teaching younger children Computer Science concepts would require a completely different curriculum, for which the courses offered by Code.org are reccommended.

Before going into the details of the curriculum, it is important to outline what kind of philosophy went into it's creation. With the new curriculum, certain philosophical goals needed to be met. The first of the goals was to create a curriculum that was fun, had a high level of engagement, and was something that was easily approachable. A lot of the Computer Science programs in the K-12 realm have the stigma that is a class just about programming, making boring command line programs like calculators or Tic-Tac-Toe games. This curriculum attempts to do away with that notion, by incorporating Game Design elements, where students will be making something that they find enjoyable.

This curriculum is designed to be both it's own independent course of study, while also complementing the other core curriculum studies. In particular, this Computer Science

© 2016 Andrew David Sweeney All Rights Reserved

curriculum would complement the subjects of Math, Art, Music, English (Literature), and the various Sciences. In tandem with that notion, the curriculum obviously involves more than just programming. The curriculum features story writing, sound and music composition, and art generation.

The curriculum should also serve the purpose of showing the students how Computer Science and everything taught is important in their lives, no matter what they choose to do in the future. All of the skills addressed in the curriculum will have students explore their interests and hopefully get them started down the right path.

One of the larger goals of the curriculum is to emphasize team-work, and working together. A lot of the core K-12 curriculums don't involve any sort of group-based learning. As such, this curriculum seeks to correct that by having students put into groups to complete team-based assignments. This lets students get an idea of what being out in the real world is actually like, where working at a job - especially a technology based job - is not a solo effort. This endeavor hopefully evokes a large collaborative learning environment, where students can thrive.

With all of the base philosophy used to construct this curriculum explained, it is now time to go over the actual curriculum. The curriculum is split effectively into two year-long courses: Introduction to Computer Science and Advanced Computer Science. As stated earlier, the curriculum is meant to be taken any time during High School, hopefully with the two courses

© 2016 Andrew David Sweeney All Rights Reserved

being taken back-to-back. For ease of reading, the curriculum has been split in eight separate sections. This also coincides with public schools commonly being based around a quarterly reporting schema. As such, each section of the curriculum represents around nine weeks of instruction.

© 2016 Andrew David Sweeney All Rights Reserved

Year 1 Quarter 1 Curriculum

Week 1	Course Introduction The History of Games / Game Literacy
Week 2	Board Game Analysis Popular Game Mechanics
Week 3	Board Game Project Groups Formed Explanation of Quarterly Project
Week 4	Board Game Project Work Time Game Goals and Genres
Week 5	Historic Video Game Analysis
Week 6	Board Game Project Play Testing
Week 7	The Art of Selling a Board Game
Week 8	Board Game Project Play Testing
Week 9	Board Game Project Presentations - Competition

A book that is recommended to used for the first two quarters is *Level Up! The Guide to Great Video Game Design* by Scott Rogers.

© 2016 Andrew David Sweeney All Rights Reserved

For the first week of instruction, the class is discussing the course and going through the history of electronic games. Making sure the students are aware and comfortable about what is going to be discussed over the year is very important. The students need to know that they are getting into programming, sound design, and digital art creation. It is also important that students know the history of electronic games, as they need to know where games came from and how they evolved to how they are in current times. The section going through the history will start with the earliest electronic games and hit all of the majors milestones and/or game-changers of the industry.

For the second week of instruction, multiple board games will be brought in, (all at the discretion of the teacher). The games will all be played through while discussing the different mechanics that are occurring, and how those mechanics tie the game together. During this week, a lecture on popular computer game mechanics and how they are created will occur. This week is all about introducing the idea of thinking more about games - any kind of game - as it may not be something that students are initially used to.

For the third week of instruction, the quarterly project will be explained and the groups for that project will be formed. The quarterly project is simply to design and put together a board, dice, or card game - or some sort of combination of those items. Students will have until the end of the quarter to create their game, where they will present to the rest of the class. This week, if there is still time, group brainstorming and initial work will also happen.

© 2016 Andrew David Sweeney All Rights Reserved

For the fourth week of instruction, there will be more work time for the groups to get together and work on their game projects. Another lecture will also be given, discussing the various goals and genres that are used for computer games. The lecture will discuss how certain genres create the goals of the games and vice-versa, and why it is important to determine both of these game characteristics early into development.

For the fifth week of instruction, students will be assigned various historic computer games that they will be required to play and write about. What the students should strive to write about is analysis of game mechanics, how the game changed the industry, and designing a more current version of the game, detailing what would be upgraded and changed.

For the sixth week of instruction, students will play-testing their quarterly projects along with the teacher. Feedback will be provided on which mechanics work and which don't work. Students will play-test other group's projects as well, giving all groups the feedback needed to continue improving the design.

For the seventh week of instruction, students will have time to work on their projects, as well as learning about the business pitching process. Students will learn how to create a presentation meant to sell their game in the real world. The take-away is that students should be crafting their presentations as if they were pitching their game to business executives.

© 2016 Andrew David Sweeney All Rights Reserved

For the eighth week of instruction, another play-testing session will be carried out amongst all groups in order to iron out all of the wrinkles of their design.

For the ninth week of instruction, students will be pitching theirs games to the class. To make presentations more interesting, the students will be given a certain amount of fake currency. Students will use this currency to vote for which group they would invest in, with students being allowed to allocate their currency to whoever they want. Students will not be able to invest in their own project. The group with the most currency invested wins, where the prize could be extra points or something within the context of the class.

© 2016 Andrew David Sweeney All Rights Reserved

Year 1 Quarter 2 Curriculum

Week 10	Video Game Player Elements Introduction to Game Maker
Week 11	Game Maker Lesson - The Basics of Making a Platformer
Week 12	Game Maker Lesson - Platformer Power-ups
Week 13	Game Maker Lesson - Platformer Enemies and AI
Week 14	Game Maker Lesson - Platformer Extras
Week 15	Game Maker Lesson - Platformer Extras 2
Week 16	Game Maker Lesson - Particle Effects Quarterly Project Discussion
Week 17	Game Maker Project - Come with Custom Elements Video Game Story Development
Week 18	Game Maker Project Exhibition

For the tenth week of instruction, a lecture on the various player elements will be given. The lecture will discuss all of the most commonly used elements and how they can be

© 2016 Andrew David Sweeney All Rights Reserved

implemented. A discussion on how the elements could be improved will also be coordinated. An introduction to the game engine Game Maker will also be given. [8] The introduction will be an over-view of the engine, including the strengths and weakness of the engine, what the engine is meant to do, and an introduction to the interface of the engine.

For the eleventh week of instruction, lessons for creating a two-dimensional platformer game will be given. The lesson will involve the programming of the core platformer features, such as movement, collision, and general set-up of the project framework. Programming topics include variables, functions, and basic logic statements.

For the twelfth week of instruction, the lessons on creating platformer elements continue. The lessons for this week will cover the creation of power-up elements, and the creation of basic user interface elements. Elements such as health and score elements will be programmed, as well as score and health power-ups. The programming concepts learned this week are more advanced logic structures.

For the thirteenth week of instruction, the lessons on creating platformer elements continues The topics covered during this week include the creation of enemy elements along with the artificial intelligence that will control them. The programming concepts learned this week includes basic artificial intelligence, and conditional loops.

© 2016 Andrew David Sweeney All Rights Reserved

For the fourteenth week of instruction, platformer creation lessons continue. The topics covered this week include extra elements that are common to platformer games, including elements such as advanced movement, adding double-jumping and wall-jumping. Adding in a variable jump height will also be introduced.

For the fifteenth week of instruction, the final elements crucial to platformer games will be created. Elements include creation of a checkpoint system, and adding in new platform types, such as moving platforms.

For the sixteenth week of instruction, a lesson on the creation of particle effects will be conducted. Particle effects will be programmed into the platformer using Game Maker's built in particle system. During this week, the quarterly project will be introduced. The project will consist of the student's working individually and taking their platformer games to a new level. Students will be tasked with figuring out how to create new elements using what they have learned.

For seventeenth week of instruction, each student should have an idea of what they want to add to the current platformer project. Teachers will double-check to make sure that the elements wanted are doable in the time-frame, and help the student figure out a framework for creating those elements. There will also be lecture given on game story development, discussing where stories come from, and how they are developed for the game industry.

© 2016 Andrew David Sweeney All Rights Reserved

For the eighteenth week of instruction, students will be showing off their quarterly projects, allowing the other students to play their creations. Students will provide feedback to each other and vote on their favorite mechanics. (Specifically mechanics, and not games will be voted on.)

© 2016 Andrew David Sweeney All Rights Reserved

Year 1 Quarter 3 Curriculum

Week 19	Introduction to Paint.net
Week 20	Paint.net Pixel Art Lesson
Week 21	Advanced Paint.net Features
Week 22	Game Maker Lesson - Animation
Week 23	Game Maker Final Project Explanation Game Maker Final Project Groups Formed and Brainstorm Session
Week 24	MU.Lab Introduction Sound Design and Music in Games
Week 25	MU.Lab Lesson - Custom Sound Effects
Week 26	MU.Lab Lesson - Basic Music Creation MU.Lab Quarterly Project Explanation and Work
Week 27	MU.Lab Quarterly Project Presentations Game Maker Final Project Pitches

© 2016 Andrew David Sweeney All Rights Reserved

For the nineteenth week of instruction, students will be introduced to the art creation program: Paint.net. [9] Students will learn the interface and how the basic tool set works. By the end of the week, students should have a good grasp on how to use the program.

For the twentieth week of instruction, students will continue working in Paint.net. A lesson involving the creation of pixel art will be given, where basic principles will be taught. Ideas such as color theory and pixel placement will be discussed.

For the twenty-first week of instruction, Paint.net will still be used to explore art creation. This week, some advanced concepts, such as layering, and filters will be discussed in order to give pixel art pieces more uniqueness.

For the twenty-second week of instruction, students will return to Game Maker in order to learn one final lesson: animation. Students will combine their knowledge of Game Maker and Paint.net in order eto create their own custom animated characters.

For the twenty-third week of instruction, the final projects of the course will be discussed. The final projects will consist of the students again forming teams in order to create a playable computer game, using Game Maker as the engine. The groups will be pitching their games at the end of the quarter to the rest of the class. Students will be forming groups during this week, as well as starting work on their ideas.

© 2016 Andrew David Sweeney All Rights Reserved

For the twenty-fourth week of instruction, a new topic will be introduced: sound and music design. Students will be introduced to the program MU.Lab. [10] A basic introduction to software will be conducted, with students learning the interface and how to create basic sounds. A lecture about sound and music design for computer games will be given, discussing the importance of sound and music, as well some music fundamentals.

For the twenty-fifth week of instruction, more lessons on how to work with MU.Lab, specifically when it comes to the creation of sound effects. The MU.Lab built in tools will be used to record and modify sounds to fit a computer game. Students will be tasked with making their own sound effects for their final projects.

For the twenty-sixth week of instruction, students will learn how to use MU.Lab to create custom music. Students will use MU.Lab's tools to synthesize music pieces. The quarterly project will also be introduced, where students will be creating their own three-to-four minute music piece. Students will present their pieces and receive feedback.

For the twenty-seventh week of instruction, students will be presenting the MU.Lab pieces they composed. Students will receive feedback in order to improve on their works. Student groups will also be pitching their game ideas to the class. These pitches will serve as a way for the rest of the class to understand what each group to trying to accomplish.

© 2016 Andrew David Sweeney All Rights Reserved

Year 1 Quarter 4 Curriculum

Week 28	Game Maker Final Project Work Session
Week 29	Video Game World Creation and Character Development
Week 30	Game Maker Final Project Work Session
Week 31	Video Game Level Design Principles
Week 32	Game Maker Final Project Work Session
Week 33	Game Maker Final Project Work Session
Week 34	Game Maker Final Project Work Session
Week 35	Game Maker Final Project Presentations
Week 36	Final Play Week

For the twenty-eighth week of instruction, initial work of the final project will being. Student groups will use in-class time to work on their games, while the teacher watches for progress and helps groups where needed.

For the twenty-ninth week of instruction, a lecture about computer game world creation and character development will be given. The lecture will detail the process of world creation and character archetypes as they affect that world.

© 2016 Andrew David Sweeney All Rights Reserved

For the thirtieth week of instruction, students will continue working on their group projects using in-class time, with the teacher providing assistance and feedback where needed.

For the thirty-first week of instruction, a lecture on the ideology of level design will be given. The lecture will feature ideas for level generation and discussion of how to properly balance levels throughout a larger game will be occur.

For weeks thirty-two through thirty-five, students will continue working on their group projects using in-class time, with the teacher providing assistance and feedback where needed.

For the thirty-sixth week of instruction, students will present their final projects, allowing the rest of the class play their games. Feedback will given to all groups from both the teacher and other students. If possible, other faculty should be invited to the presentations to see the final products.

© 2016 Andrew David Sweeney All Rights Reserved

Year 2 Quarter 1 Curriculum

Week 1	Course Introduction Initial Formation of Groups
Week 2	The Game Design Process
Week 3	Introduction to Game Design Documents
Week 4	Quarterly Project Description First Assignment - A Game One Sheet Due
Week 5	Group Project Work Session
Week 6	Group Project Work Session Second Assignment - Game Ten-Pager Due
Week 7	Group Project Work Session Third Assignment - Game Asset List Due
Week 8	Group Project Work Session
Week 9	Game Design Document Presentations - Pitching The Game

For the first week of instruction, students will be presented with the overview of the class. Throughout the year, students will be working as a group to create a three-dimensional

© 2016 Andrew David Sweeney All Rights Reserved

game using the Unity game engine. Design and work on this game will begin during the first quarter, where during the first week of class, student groups will be formed. These groups, unless issues arise, will remain the same the entire year.

For the second week of instruction, a lecture about the ins and outs of the game design process will be given. This lecture will detail the full process of game creation, starting with having the initial idea, moving through document creation, and ending up in the production phase. The lecture will focus on why it is important to have plan moving forward every step of the way when putting together a computer game.

For the third week of instruction, a lecture on the various game design documents will be given, teaching students the purpose of each document and who to format them. Groups will be given their first task in settling on a game idea, where the first assignment is the Game One-Sheet, a single page document with the bare minimum of what their game is going to be.

For the fourth week of instruction, the first assignment will be due. Groups will have time to work together in class and brainstorm what they want to create. The quarterly project will also be outlined, where at the end of the quarter, each group will evolve their One-Sheet into a full Game Design Document.

For the fifth week of instruction, groups will be given time to work on their projects, with feedback from the teacher.

© 2016 Andrew David Sweeney All Rights Reserved

For the sixth week of instruction, groups will be given time to work on their projects, along with the second assignment being due: The Ten-Pager. Essentially, this document is a ten page, condensed version the Game Design Document. It is at this time that groups will have added more to their game design, and have their core features and mechanics thought out and defined.

For the seventh week of instruction, groups will be given time to work on their projects, along with the third assignment being due: The Game Asset List. Groups will be tasked about taking what they want to make and break it down into the various assets that make up a game. Groups will then create a list of all of the art, sound, and programming assets that they will be required to make or obtain in order to make their game.

For the eighth week of instruction, groups will be given time to work on their projects, with feedback from the teacher.

For the ninth week of instruction, the groups will be presenting their game designs along with turning in their final game design documents. Other students and the teacher will provide feedback.

© 2016 Andrew David Sweeney All Rights Reserved

Year 2 Quarter 2 Curriculum

Week 10	Introduction to Unity - Unity Basics
Week 11	Unity - First Person Platformer Lesson 1 Initial Project Setup Creating a First Person Controller
Week 12	Unity - First Person Platformer Lesson 2 Creating Terrain Creating Ranged Weapons
Week 13	Unity - First Person Platformer Lesson 3 Enemy Creation and Artificial Intelligence
Week 14	Unity - First Person Platformer Lesson 4 Creating The User Interface
Week 15	Quarterly Project Description Quarterly Project Work Session
Week 16	Unity - First Person Platformer Lesson 5 Particle Systems Quarterly Project Work Session
Week 17	Unity - First Person Platformer Lesson 6 Power-ups and Extra Features

© 2016 Andrew David Sweeney All Rights Reserved

	Quarterly Project Work Session
Week 18	Quarterly Project Presentations Voting For The Best New Mechanic

The recommended book for this quarter through the end of the year is *Learning C# by Developing Games with Unity 5.x - Second Edition* by Greg Lukosek.

For the tenth week of instruction, students will get their first taste of the Unity Game Engine, with a full introduction to the program. [11] A lesson on the interface and how to navigate the engine will have the students familiar enough to begin working with it by the end of the week.

For the eleventh week of instruction, the first lesson on creating a First Person Platformer will be given. Students will be taught how to setup the initial project settings, as well as creating a fully programmed first person controller. Students will taught how to make use of Unity's physics libraries as well as being introduced to the programming language C#.

For the twelfth week of instruction, the second lesson on creating a First Person Platformer will be given. Students will be taught how to use Unity's built in terrain editor to create the landscapes that their games will take place in. Students will also be programming some very basic weaponry into the game, which can easily be extended.

© 2016 Andrew David Sweeney All Rights Reserved

For the thirteenth week of instruction, the third lesson on creating a First Person Platformer will be given, where students will be programing enemies and the conditional statements that control those enemies. This lesson will expand on the artificial intelligence lesson from the previous course.

For the fourteenth week of instruction, the fourth lesson on creating a First Person Platformer will be given, where students will learn how to use Unity's built in user interface system. Students will programing custom user interface elements, such as health systems, score systems, ammo systems, and weapon changing systems.

For the fifteenth week of instruction, students will be introduced to the quarterly project. The project will be to take the First Person Platformer and add new mechanics to the game. The students will modify what has already been created and expand it to include features and mechanics that have not been taught. Students will be given in-class time to begin work on this project.

For the sixteenth week of instruction, the fifth lesson on creating a First Person Platformer will be given, where students will be learning how to use Unity's particle systems in order to add more flair and visuals to their current game. Students will also be given in-class time to work on their quarterly project.

© 2016 Andrew David Sweeney All Rights Reserved

For seventeenth week of instruction, the sixth lesson on creating a First Person Platformer will be given, where students will take their programming skills further by creating power-ups and other extra environment features. Students will also be given in-class time to work on their quarterly project.

For the eighteenth week of instruction, students will be presenting their quarterly projects and other students will vote on what they think to be the most clever feature or mechanic created.

© 2016 Andrew David Sweeney All Rights Reserved

Year 2 Quarter 3 Curriculum

Week 19	Introduction to Blender
Week 20	Creating Basic Architecture using Blender
Week 21	Incorporating Textures made using Paint.net into Blender - UV Lesson
Week 22	Creating Basic Bipeds using Blender
Week 23	Creating Basic Animations using Blender - Skeletons and Skinning Discussion
Week 24	Integrating Blender Models into Unity
Week 25	Quarterly Project Description Quarterly Project Work Session
Week 26	Quarterly Project Work Session
Week 27	Quarterly Project Presentations

The recommended book for this quarter is *The Complete Guide to Blender Graphics: Computer Modeling & Animation, Third Edition* by John M. Blain.

For the nineteenth week of instruction, students will be introduced to the 3D Modeling software, Blender. [12] Students will learn about the interface and how to navigate the software,

© 2016 Andrew David Sweeney All Rights Reserved

as well as learning about what the software is used to do. Students will be creating basic geometrical primitives during this week.

For the twentieth week of instruction, students will learn how to model basic architecture using Blender. Students will get more familiar with the program by recreating popular architecture such a Greek and Roman buildings, and modern American buildings.

For the twenty-first week of instruction, students will learn how paint textures returning to Paint.net. Students will also be be shown how to apply those textures to the 3D models they have created in Blender. The topic of UV mapping will also be introduced, as it is important to the texturing process. Students should be able to create well-crafted UV maps and paint textures on top of them by the end of the week.

For the twenty-second week of instruction, students will be learning on how to create basic biped models in Blender. Early models will include creatures such as robots, and will move onto simple humanoid creatures.

For the twenty-third week of instruction, students will be taking the humanoid models they have creating and learning about the animation process. Students will be creating a skeleton for their model, as well as attaching the skeleton to the model via the skinning process. Students will then learn how to create basic animations through keyframing in Blender.

© 2016 Andrew David Sweeney All Rights Reserved

For the twenty-fourth week of instruction, students will learn how to take the models they created using Blender and integrate them into Unity. Students will learn how to connect the newly created models to the existing code in order to have a fully working character. Students will also be integrating their architecture models to add more to their scene.

For the twenty-fifth week of instruction, students will be given their quarterly projects. The project is to coordinate with their groups and create 3D models that will end up being important to their final projects. Once each student has a series of at least three models to create, students will be given in-class time to begin work on their projects.

For the twenty-sixth week of instruction, students will be given in-class time to work on their quarterly projects.

For the twenty-seventh week of instruction, students will be presenting the models that hey have created to the class. Students will receive feedback from other students as well as the teacher.

© 2016 Andrew David Sweeney All Rights Reserved

Year 2 Quarter 4 Curriculum

Week 28	Final Project Group Work Session Group Game Design Document Revision
Week 29	Final Project Group Work Session
Week 30	Final Project Group Work Session
Week 31	Final Project Group Work Session
Week 32	Final Project Alpha Due Project Playtesting with Student Feedback
Week 33	Final Project Group Work Session
Week 34	Final Project Group Work Session
Week 35	Final Project Final Playtesting Session
Week 36	Final Project Presentations

For the twenty-eighth week of instruction, students will be revisiting their Game Design Documents they created during the first quarter. After learning how to use Unity and Blender, groups will be tasked to go back through their design and alter it into something that can be created in a few weeks. Students should take away that it is alright to go back and revise and remove features because of time and/or difficulty restraints. Groups will also begin work on creating their games.

© 2016 Andrew David Sweeney All Rights Reserved

For weeks twenty-nine through thirty-one, groups will use in-class time to work on their projects.

For the thirty-second week of instruction, the project in alpha form will be due. Projects should be at a playable state at this point, where other students can play through their game and give them feedback on how the game plays.

For week thirty-three and thirty-four, groups will use in-class time to work on their projects.

For the thirty-fifth week of instruction, groups will have one final week of play-testing, and getting the last bit of feedback from other students. Groups should use this information and feedback to put the final amount of polish on their projects.

For the thirty-sixth week of instruction. Groups will be presenting their projects in a public presentation. School faculty will be invited to come celebrate the student's accomplishments as well as play their finished projects.

Using this curriculum as a guideline, schools around the nation should be able to integrate a Computer Science course that not only gets students involved and excited about the subject, but also does the job of teaching them the correct fundamentals. Using some sort of

© 2016 Andrew David Sweeney All Rights Reserved

Applied Computer Science is the way to teach the subject, instead of just traditional programming education.

© 2016 Andrew David Sweeney All Rights Reserved

References

1. Why Computer Science Education in K-12 Settings Is ... (n.d.). Retrieved October, 2016, from http://www.huffingtonpost.com/acm-the-association-for-computing-machinery/why-computer-science-educ_b_12010476.html

2. Landscape of K-12 Computer Science Education in the U.S. (n.d.). Retrieved October, 2016, from http://csedu.gallup.com/183725/landscape-computer-science-education.aspx

3. A Decade of ACM Efforts Contribute to Computer Science for ... (n.d.). Retrieved October, 2016, from http://cacm.acm.org/magazines/2016/4/200155-a-decade-of-acm-efforts-contribute-to-computer-science-for-all/fulltext

4. Bureau of Labor Statistics. (n.d.). Retrieved October, 2016, from http://www.bls.gov/news.release/pdf/ecopro.pdf

5. WebCASPAR - Webcaspar - Data.gov. (n.d.). Retrieved October, 2016, from https://catalog.data.gov/dataset/webcaspar/resource/580740b2-da6b-4267-b470-bd13c2ac2740

6. About Us | Code.org. (n.d.). Retrieved October, 2016, from https://code.org/about

7. Code.org's partner school districts. (n.d.). Retrieved October, 2016, from https://code.org/educate/partner/district-partners

8. Game Maker in Schools | GameMaker Blog. (n.d.). Retrieved October, 2016, from http://gamemakerblog.com/2009/05/30/game-maker-in-schools/

9. Paint.NET - Free Software for Digital Photo Editing. (n.d.). Retrieved October, 2016, from http://www.getpaint.net/index.html

10. MuLab DAW and MUX Modular Synth and Effect - MUTOOLS. (n.d.). Retrieved October, 2016, from http://www.mutools.com/index.html

11. Unity - Game Engine. (n.d.). Retrieved October, 2016, from http://unity3d.com/

© 2016 Andrew David Sweeney All Rights Reserved

12. Blender.org - Home of the Blender project - Free and Open ... (n.d.). Retrieved October, 2016, from https://www.blender.org/

13. Cracking The Code To More K-12 Computer Science In Washington State. (n.d.). Retrieved October, 2016, from http://thelens.news/2016/09/13/more-k12-computer-science-washington-state/

14. Education | YoYo Games. (n.d.). Retrieved October, 2016, from https://www.yoyogames.com/education

© 2016 Andrew David Sweeney All Rights Reserved