Journal of Technology-Integrated Lessons and Teaching
(2024) Volume 3, Issue 1; DOI: 10.13001/jtilt.v3i1.8479

Motion to Immersion: Combining Computer Science, Virtual Asset Design, and Motion Capture for High School Students

William Tai and DeniRob Arnett, University of Idaho

Overview

There are many opportunities for learning under the umbrella of computer science. Coding and programming experiences can overshadow other computational learning programs. Digital asset design for virtual environments is applicable to game design and motion capture laboratories in the movement sciences. This summer camp for high school students featured coding and digital asset development as applied to motion capture and the prevention of musculoskeletal injuries.

Topics: Applied Mathematics, Coding, Computer Science, Motion Capture, Movement Science, Virtual Asset Development, Virtual World Design

Time: The total program took 15 days (3 weeks): 3 hours each morning for computer science and digital asset design and 3 hours each afternoon for motion capture/kinesiology

Materials

- VR capable backpack PC
- VR headset such as the HTC Vive or Oculus Rift
- Virtual asset development environment such as Unity (n.d.) or the Epic Games (n.d.) Unreal Engine
- Free virtual world assets (e.g., sounds, textures)
- Computers running Unity or Unreal Engine
- Google account
- Motion capture/tracking software and cameras
- Motion tracking dots
- OnlineGDB (n.d.) or equivalent cloud-based integrated development environment (IDE)
- Bandicam (n.d.) or equivalent screen recording software
- Print Array Demo (CPP Attachment)

Context-at-a-Glance

Setting
15-day enrichment computer science, digital art, and movement science class for TRIO Upward Bound students in the United States.

Modality
Face-to-face

Class Structure
In class, students worked in groups of five to create digital objects to insert into the developed virtual simulation.

Organizational Norms
The class was the last face-to-face enrichment class before the March 2020 United States lockdown and quarantine period for the Covid-19 pandemic.

Learner Characteristics
Learners were 20 high school students who met TRIO eligibility—low-income, first-generation college students, or both. Learners were also classified as under-resourced or underrepresented.

Instructor Characteristics
The class was designed by a Ph.D. candidate majoring in STEM Education and assisted by a master's student in Career and Technical Education. They were also the instructors during the class.

Development Rationale
To expose students to more STEM content, TRIO programs across the United States received supplemental funds targeting Computer Science enrichment. We developed a summer module highlighting coding, virtual asset design, virtual reality, and its applications in movement science.

Standards

Idaho State Department of Education (ISDE; n.d.) Computer Science Standards:

9-10.AP.01: Design and develop a software artifact by leading, initiating, and participating in a team.

9-10.AP.13: Explain, represent, and understand natural phenomena using modeling and simulation.

9-10.AP.16: Decompose the value of abstraction to manage problem complexity.

9-10.AP.19: Evaluate problems written by others for readability and usability.

9-10.IC.09: Practice responsible digital citizenship (legal and ethical behaviors) in the use of technology systems and software.

9-10.IC.04: Describe how computer science shares features with creating and designing an artifact such as in music and art.

9-10.IC.05: Demonstrate how computing enhances traditional forms and enables new forms of experience, expression, communication, and collaboration (e.g., virtual reality).

9-10.IC.06: Explain the impact of the digital divide on access to critical information (e.g., education, healthcare, medical records, access to training).

11-12.AP.02: Create collaborative software projects using version control systems, Integrated Development Environments (IDEs), and collaborative tools.

11-12.AP.04: Modify an existing program to add additional functionality and discuss the positive and negative implication (e.g., breaking other functionality).

11-12.AP.06: Decompose a computational problem through data abstraction and modularity.

11-12.AP.08: Evaluate efficiency, correctness, and clarity of algorithms.

11-12.AP.09: Compare and contrast simple data structures and their uses (e.g., arrays, lists, stacks, queues, maps, trees, graphs).

11-12.IC.09: Explain the impacts of computing on business, manufacturing, commerce, and society.

ISDE (2022a) Mathematical Practice Standards:

2. Reason abstractly and quantitatively.

4. Model with mathematics.

6. Attend to precision

8. Look for and express regularity in repeated reasoning.

ISDE (2022b) Science Content Standards:

MS-PS-2.2: Plan and conduct an investigation to provide evidence that the change in an object's motion depends on the sum of the forces on the object and the mass of the object.

MS-PS-3.5: Construct, use, and present arguments to support the claim that when the kinetic energy of an object changes, energy is transferred to or from the object.

PS3.C: Relationship between energy and forces.

Setup

The students followed a pre-arranged daily schedule provided by participating Upward Bound projects. The three-week summer camp facilitated immersive college experiences for students, including staying in campus housing, classes in financial literacy, and team building activities in the evening.

The morning was divided into two class sessions. Two hours were reserved for algorithmic analysis and coding in C++. The next two hours were used for digital asset design in Unity. Students worked in groups of four to five for digital asset design due to the lack of computers capable of running the design engines. The morning classes were held in the Educational Technology classroom where all staff and instructors gathered to help students, should they have any questions or concerns. There was always at least one staff member present who was first aid certified in case of an emergency.

Lunch was held between the morning and afternoon classes. It served as a way for students to recharge from the rigorous morning classes.

The afternoon class was located at the Integrated Sports Medicine Movement Analysis Laboratory (ISMMAL) where movement science instructors facilitated topics in anatomy, musculoskeletal structure, motion, motion capture, and motion capture analysis.

Context and Setting

Summer Enrichment Program Context

The summer enrichment program was hosted by the University of Idaho TRIO Upward Bound as a collaborative effort by two Upward bound projects and the Department of Movement Sciences at the university. Each of the collaborating projects received one-time supplemental funding from the U.S. Department of Education (2023) for STEM programming in 2019. The one-time cash infusion resulted in an integrated math, computer science, and movement sciences module that took place at the main campus of the University of Idaho in the College of Education, Health, and Human Sciences building.

Participants

Student participants in the TRIO projects meet one of three eligibility requirements: (a) low-income, (b) first-generation college students, or (c) both. Program participants lived in North Central Idaho and/or the Lewis-Clark Valley on the boarder of Idaho and Washington state.

PROJECT GOAL

The foundations of this project was based on a basic understanding of coding/scripting manipulation of digital assets, placement and positioning in a virtual three-dimensional plane, and the translation of physical real-world objects into the virtual environment (Figure 1). Students engaged in a project that helped them gain an introductory understanding of mathematical and science applications as they are applied to human body mechanics, analyses, and motion capture fields.

The project featured the Unity Engine, which is primarily used for game design and development to render and produce virtual worlds (Unity, n.d.). Free educational licenses were secured through Unity for this project.

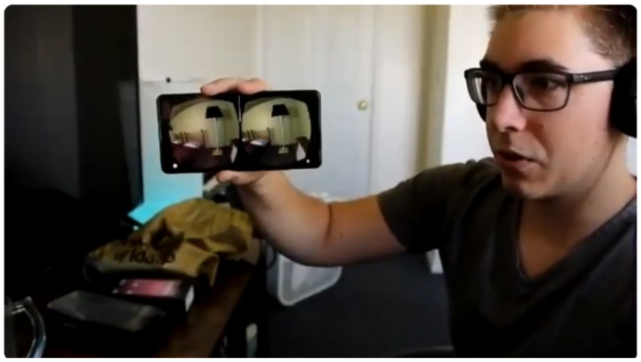

Figure 1: The virtual world, as shown through a mobile device, as part of the developer test that was executed prior to student participation.

Three units were developed by the authors (foundations of programming in C++/C#, virtual asset manipulation and management, and human motion mechanics and analyses) to facilitate learning for the students.

Instructor Expertise

Instructors need to have a solid understanding of the C programming language family as well as a basic understanding of the positional manipulation of virtual assets in virtual world development environments. To support student questions regarding basic programming, conditional statements, loops, and debugging, instructors should have fluency in mathematical concepts considered foundational to computer science, such as truth/logic tables and conditional statements.

Additionally, collaborating instructors in the movement sciences must have an expert understanding of movement and human body mechanics, motion capture software, and motion capture analysis.

Instructors should also have expertise in virtual world building for the creation of the physical space in the digital world that students will move through for motion capture and analyses.

Camp Overview

The project consisted of two separate classes, computer science and body mechanics, each with three developed units for instruction and laboratory work. The computer science classes were held in the morning and the body mechanics classes were held in the afternoon after lunch.

The first unit of instruction in the computer science (CS) class was a brief introduction to logic tables in mathematics, C/C++ programming and debugging in an online Integrated Development Environments (IDE). The second unit of instruction in the CS class was a brief introduction to copyright as a developer, loops in programming, and the design process with virtual assets. The third unit of instruction in the CS class was a brief introduction of the manipulation of virtual objects in a pre-designed VR-ready environment.

The body mechanics (BM) class had three units of instruction: (a) hands-on identification and palpation, (b) preparation of the laboratory environment, and (c) static and dynamic data trials. The hands-on identification and palpation section introduced human anatomical landmarks for the placement of reflective markers. The preparation of the laboratory environment section introduced lab readiness for data collection using a motion capture system. The static and dynamic data trials gave students experience in data collection and post-processing in a movement laboratory.

Week 1 - First Computer Science Activity: Computational Decisions in Code

An introduction to functional programming took place before lessons about conditional and repetitive programming. During computational thinking topics, simple programs such as user-initiated text output, were used as an intermediary to demonstrate computational decisions then broken down in terms of mathematical truth/logic tables.

Day 1 Morning

Students engaged in a pre-test activity, which served as a baseline for comparing their results to post-course knowledge.

An introduction to the IDE, OnlineGDB (n.d.), was facilitated along with language syntax in the family of C-based programming. Instructors provided written sample programs to students for the sole purpose of trial-and-error experimentation. Student groups were also formed in anticipation of the activities in the virtual asset design portion of the class. Topics in console input and output to demonstrate the students' first interaction with text-based programming operations were introduced.

Days 2 & 3 Morning

On the second and third days, students learned the importance and uses of libraries in an IDE. A sample program with built-in errors was given to students as a formative assessment to determine their ability to problem solve and debug errors using the debug function of OnlineGDB.

Days 4 & 5 Morning

On days four and five, students were introduced to mathematical truth tables and computational decision making. Conditional statements using flow charts to demonstrate the flow of computational logic, and loops, demonstrated with flow charts, were also introduced. We showed code fragments of loops to students but only tasked them to demonstrate initial understanding through flow charts. Coding in the online IDE was reserved for Week 2. We gave a unit summative assessment using flow charts and code fragments to students to measure their understanding.

Week 2 - Second Computer Science Activity: Virtual Assets

Day 6 Morning

On day six, we divided the class into two equally timed sections. In the first half, students were introduced to iterative loops in programming, specifically "for loops" and "while loops." "Do/while loops" were excluded given that there is a higher chance of error for the beginning programmer. In the second half, a brief introduction of copyright in the developer's role was introduced because copyright has a continuing impact on many professionals in the industries concerning digital assets.

Days 7 & 8 Morning

On day seven, students created a portfolio, a space where they accumulated their in-progress and completed work. Portfolios are gaining momentum in the hiring process as it is a showcase of their abilities as a digital asset designer and their evolution from student to professional (Fuselier, 2015). Examples of art and digital asset portfolios were provided to give students an idea of portfolio creation, organization, and maintenance.

On day eight, students searched for and collected digital assets. Digital assets, in this case, were defined as two- or three-dimensional components that make up a virtual world including models, textures, materials, and/or lighting (Unity Documentation, 2024, 2D or 3D projects). Prior to their search and collect exercise, an introduction into the assessment of three-dimensional objects and their environmental interactions was presented. Many digital assets must be paid for, but giving students the option to choose free or using creative commons licensed assets gets students into working with the game engine quicker.

Days 9 & 10 Morning

On day nine, students were introduced to the design process for digital asset creation. The design process was emphasized earlier in the introduction of student portfolios. The steps for the design process, unique to the instructor, were as follows:

1. Identify the problem.
2. Research the problem.
3. Ideate solutions.
4. Evaluate and select your solution.
5. Create a prototype.
6. Test and troubleshoot.
7. Make improvements and release the final product.

On day 10, students were introduced to file organization and naming conventions, an important, but often overlooked topic in the technology and design fields. File organization is integral in digital asset creation and development because, for different development environments to communicate, there must be a way (that is logical to the student) to locate shared assets. Therefore, to aid the students in developing their individual way of fast asset location inside a system, naming conventions are emphasized as shorthand object descriptors.

Day 11 Morning

On day 11, students separated into groups to complete a challenge assessment. The challenge assessment was a formative assessment designed to measure student's understanding of basic game engine and asset management/interaction skills.

Students created a virtual Rube Goldberg machine (2024) inside the game engine as an assessment. A Rube Goldberg machine is defined as a chain reaction machine performing a task in an overly complicated manner. Students had to cite, create, and use accumulated assets to render a playable/workable prototype. The students added their finished projects to their portfolio.

Third Computer Science Activity: The Virtual World

Days 12 & 13 Morning

On days 12 and 13, students were tasked to create a unique digital asset to be inserted into the virtual world (Figure 2). Students worked together in groups to facilitate collaboration. Maximum parameters (15 feet by nine feet) were given to students because the physical space that was replicated inside the virtual space had size restrictions.

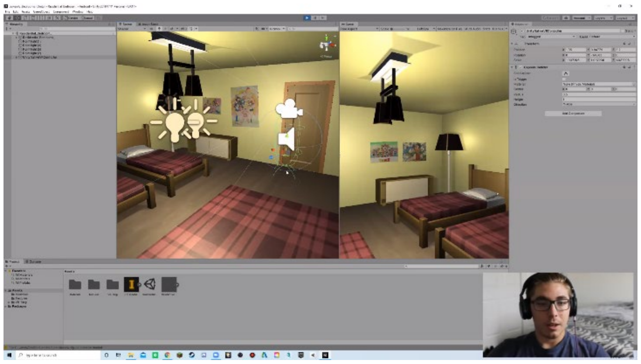

Figure 2: Screen capture of the virtual world simulation scene.

Days 14 & 15 Morning

On days 14 and 15, CS class time was conducted within the movement lab. At the lab a physical representation of the virtual world, including navigable obstacles for students to physically maneuver through, was created. The replication was precisely measured to reduce potential student injuries while moving through the course. At this point, student-created digital assets were inserted into the virtual scene so that students could see and interact with their creations.

First Body Mechanics Activity: Human Anatomy and Palpation

Days 1 to 5 Afternoon

On day one, students were introduced to the physical laboratory space where movement data was collected by the motion capture system. The dimensions of the lab space were recorded for later use by the students in designing their digital assets. At this time, it was important to stress caution that the motion capture cameras could not be damaged or touched by students in any way. The motion capture system is integral to providing a full experience for the students. Any accidents can decalibrate the system and result in system failure. We presented further safety procedures to students so that physical space and action awareness were emphasized.

On day two, students were introduced to basic anatomical motion. Landmarks on the body on non-student models were shown so that students would better understand what to look for. Motion capture dots, which allowed the motion capture system to collect data, were passed around for the students to examine.

On days three, four, and five, students worked in groups placing motion tracking dots on anatomical landmarks, gaining hands-on experience. During this time, student groups rotated into the motion capture lab space, performing different movements, to get familiar with calibration and the process of doing motion capture activities.

Second Body Mechanics Activity: Preparing the Laboratory

Days 6 to 10 Afternoon

On days six, seven, and eight, students took turns in the motion capture laboratory. With the motion capture system active, the students performed simple gestures and movements.

On days nine and 10, students were introduced to how post-processing works in a motion capture lab. Post-processing is the process of connecting the motion tracking dots in a simulated computer environment for gait and other movement analysis (Vicon, 2022). Students then gained hands-on experience by preparing their motion tracked representations through post-processing.

Third Body Mechanics Activity: Static and Dynamic Data Trials

Days 11 to 15 Afternoon

On days 11, 12, and 13, students continued to prepare their physical lab environment to represent their virtual environment counterparts. Precise measurements of objects that match both in the real world and the virtual world is essential in preventing injury since the virtual world had obstacles in the course.

On days 14 and 15, the students filled their day with laboratory activities. Students were each fitted with the backpack PC and VR equipment one at a time before having motion tracking dots placed on their bodies, including the VR headset (Figure 3). The lab captured each of the students' data trials, once while load bearing with the backpack PC and once without the backpack PC. After each data trial, the student performed their post-processing while setting up the next student. During each students' trial, a recording was created via the Bandicam software running in the background in the backpack PC while the virtual environment was active (Figure 4).

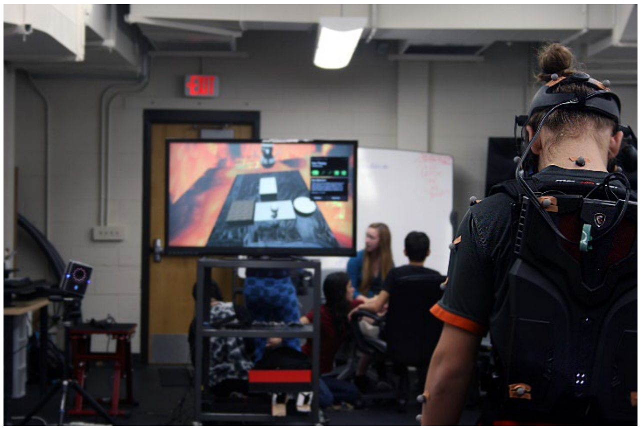

Figure 3: Student equipped with VR backpack and motion tracking dots, getting ready to execute the course.

Figure 4: Video recording of student navigating the course in VR and the physical space.

Curriculum Adaptation in a Public High School

This three-week curriculum was adapted by the authors for a public high school technology class. The technology class focused on beginning game design; thus, only parts were adapted from the original summer curriculum. The technology class obtained free student licenses of Unity Engine, provided that the students met the prerequisite requirements from Unity.

Due to the social distancing rules K-12 schools adopted after the COVID-19 pandemic quarantine period, the class was held in hybrid and asynchronous modalities. Students demonstrated their knowledge at the end of the class by creating a simple working version of a game.

Learning Representation

In this section, the learning representations for the morning CS classes during days six and eight are presented. Each day is split into two segments. We detail each segment for each day of instruction, below.

Day 6 Morning CS Classes

Student Objectives

Students will be able to:

1. Achieve a basic understanding of loops in C++.
2. Understand the differences between the terms copyright, copyleft, and creative commons.

First Segment Instructions

Since day six is the start of the second week of camp, briefly remind students of the materials from the previous week. This review should take place at the beginning of class and should last no more than 15 minutes. Take the time to answer any student questions that linger after the review of the previous week's concepts. Answering student questions should take about ten minutes.

To begin class, start by asking the students a question as a hook: *How do programmers write code to execute repetitive instructions while keeping it efficient?*

Next, introduce loops as a general concept. Explain to students what loops do and what conditions loops must have to function correctly in coding. Show examples of loops including "for loops," "while loops," and "do-while loops." However, for this class, in student exploration, only "for loops" were focused on to keep it simple.

Provide the two examples of loops detailed below in Figure 5 and Figure 6. Instruction time for this segment should take no more than 30 minutes.

```cpp
/*******************************************************************************

Intro to for Loops with Loop fragment

*******************************************************************************/
#include <iostream>

using namespace std;

int main()
{
    for (int i = 0; i < 5; i++)
    {
        cout << i << endl;
    }

    return 0;
}
```

Figure 5: Screen capture of the first for loop.

The first "for loop" shows iteration by setting the integer *i* equal to zero (Figure 5). The terminating condition is set to *i < 5* and the counter for the integer *i* is set to increase by one. Within the loop, the output statement is set to print the values of the integer *i* each time the loop iterates until it meets the terminating condition. By compiling and running the loop example, the possible values (that a value can be set to once the loop starts), increments, and when it meets the terminating condition, was demonstrated to students.

```cpp
/*******************************************************************************

Intro to for Loops with Loop fragment

*******************************************************************************/
#include <iostream>

using namespace std;

int main()
{
    for (int i = 0; i <= 5; i++)
    {
        cout << i << endl;
    }

    return 0;
}
```

Figure 6: Screen capture of the second for loop with the terminating condition changed to add an equivalency comparison.

The second "for loop" shows iteration by setting the terminating condition to *i <= 5* (Figure 6)*.* The differences between the two loops were shown to students by running them simultaneously.

Student Exploration Instructions

To support student exploration of coding, provide students with the two code samples. Task them to explore the code samples by reminding them of two procedures:

1. Log any changes made to the original code before saving and compiling the code.
2. Write down the output before running the code to see if their guesses match the program's output.

The procedures additionally count as an impromptu formative assessment, examining students' skills in debugging changes made in the code. The student exploration activity for the first half of class should take approximately 30 minutes. It is suggested to ask students after the activity to explain the differences between the two code samples.

Second Segment Instructions

Students were divided into three groups with each group responsible for the terms copyright, copyleft, and creative commons. Students were asked to define the term and share examples. Let students spend as much time as needed to complete the task. However, plan on the activity taking 15 minutes. This time limit is imposed to keep the class moving at a steady pace and to keep the students on task.

Once the students complete the task, ask them to present their assigned term and examples to the rest of the class. These presentations count as the day's summative assessment, showing students' depth of understanding in the new concepts.

If there is still time remaining at the end of class, answer any questions the students might have regarding the rest of the week or regarding topics presented in class.

Day 8

Student Objectives

Students will be able to:

1. Demonstrate an understanding of loops.
2. Explain loop application using a two-dimensional array and nested loops.
3. Identify digital assets as free or creative commons licensed objects.
4. Manipulate digital assets in the game engine interface using changed values in the object code and user interface functions.

First Segment Instructions

To start the day, remind students of loops, introduced on Day 6. Answer any questions students may have regarding loops. This review and questions and answer session should take about ten minutes.

Next, distribute [Print Array Demo (CPP attachment)] to students. Give students the opportunity to familiarize themselves with it. Then ask them, after five to ten minutes, to explain what they think the program is doing without running it.

Student Exploration Activity

Task the students to compile and execute the program, Print Array Demo, in multiple iterations. While the students are exploring, guide them using the following questions:

1. *What is the purpose of the two-dimensional array?*
2. *How do the nested loops work in association with the two-dimensional array?*
3. *The character object in the program is dynamic, meaning that it is based on user input. If the character object was changed to be static, how would you change the position of the object?*

This student exploration activity should take approximately 30 minutes. If students need extra time to complete the task, allocate another ten minutes. After completing the activity and answering the questions, have students volunteer to explain their answers to the rest of the class.

Second Segment Instructions

Start the second segment of class by giving students an assessment of their preliminary understanding regarding asset creation and collection. Below are the questions for the assessment:

1. *Define asset creation in the context of 3D object.*
2. *Explain the importance of texture mapping in 3D modeling.*
3. *Discuss the concept of collision detection in environmental interactions.*
4. *Differentiate between static and dynamic lighting in 3D environments.*
5. *Describe the process of asset collection and integration in 3D environments (incorporate copyright laws into your answer).*

The assessment serves as a baseline of student knowledge. The same assessment will be given at the end of the asset creation and collection process to assess the differences in student knowledge.

Student Activity

Using search engines like Google, task the students to search for and collect digital assets of their choice. Remind the students that the objects they collect must be free or under creative commons licensure. As the students are searching and collecting, give them instructions to log each downloaded object and note their free or creative commons usage. If the students need guidance on which websites are reputable and have free or creative commons licensed objects, suggest going to the following:

- Turbosquid (shutterstock, n.d.)
- Sketchfab (n.d.)
- Free3D (n.d.)
- Blender (n.d.)

There are other websites that host similar materials, but make sure to check for malicious content before suggesting it to students.

After students have collected at least five digital assets, task them to import their objects into the game engine and begin experimenting with them in a virtual environment.

After students gain a basic familiarity with the virtual design space, guide them through the manipulation of the digital assets. A basic familiarity means that students can move the object(s) in the design space and can rotate the camera view of the design field to view the scene from multiple angles.

Demonstrate the two methods of asset manipulation to students in a virtual design space. First showcase the click and drag method, then the value change method in the object source code (Figure 7). After the demonstration, let students explore both methods.

```
Transform:
  m_ObjectHideFlags: 1
  m_CorrespondingSourceObject: {fileID: 0}
  m_PrefabInternal: {fileID: 100100000}
  m_GameObject: {fileID: 1029230104914618}
  m_LocalRotation: {x: -0.000000059604638, y: -0, z: -0, w: 1}
  m_LocalPosition: {x: -0.012105884, y: 0.003030072, z: 0.15606086}
  m_LocalScale: {x: 1.5151515, y: 1.5151521, z: 1.5151521}
  m_Children: []
  m_Father: {fileID: 4223747152790226}
  m_RootOrder: 0
  m_LocalEulerAnglesHint: {x: 90.00001, y: 0, z: 0}
--- !u!23 &234333322831403398
```

Figure 7: Screen capture of an object's source code. It denotes the position, rotation, and scale of the object under transformation.

Students may take as much time as they need to explore the functions of the design space. Make sure to allocate the last 15 minutes of class time to wrap up and answer any remaining questions students may have.

Critical Reflection

Overall Reflection

In most computer science coding courses, there are limited real-world examples of how it is applied. For students with limited resources (e.g., TRIO eligible populations), it is more difficult to facilitate learning with continued interest after completing the program. This project helped to mitigate STEM disinterest commonly associated with program completion.

In the beginning of the project, students showed difficulties in understanding the syntax of coding in the C-family programming languages. However, after demonstrating applications of coding such as digital asset manipulation, students showed more interest by asking more questions in class. However, there was still a clear divide among students in the program. The morning curriculum interested students who regularly participated in gaming. Students who engaged more in the afternoon curriculum regularly participated in afterschool sports. However, we achieved the objectives in the beginning sections based on the results of the first formative assessment. Students demonstrated their learning by being able to map out a flow chart based on a segment of code (see Days 4 & 5 Morning section).

During week two of the program, when students found out they would be integrating their experiences in digital asset creation into the movement sciences lab, their interests aligned. Students were engaging more enthusiastically than in the previous week. The Rube Goldberg machine assignment also served as an assessment of the students' learning (see Day 11 Morning section). It measured their understanding of digital asset creation, gathering, and management. Even partial completions of the assignments were counted as successes because it showcased students' abilities in digital asset creation.

During the last week of the program, we needed to address many safety concerns. Students worked with a physical environmental representation of a virtual world, which they navigated during their final project. Therefore, all available staff were needed during the data collection trials in the motion capture lab. At the end of the program, students achieved all set program expectations and learning goals by participating in a showcase. In addition to participating in the program, we asked students to find relevant STEM career opportunities based on the contents presented in the program. This additional activity supplemented their presentations during the showcase at the end of the program.

Considering the course's scope, the overall outcomes were a success. The outcomes included brief introductions to coding, connections to digital asset creation and design, and applications in movement science. Students and instructors of the course learned many ways to incorporate different technologies collaboratively. During instruction, we found that there was student interest in different aspects of the course that we did not anticipate and did not cover due to time constraints. Therefore, for future iterations of the program, we suggest that there be an interest survey at the end of each week to assess potential topics for further exploration.

Critical Reflection For Day 6

During day six, the instructors used a new concept in a program for demonstration purposes without including previous code concepts written in the driving function or in modular form. This allowed students to focus on the new material without the previous concepts confusing them. The students grasped onto the concepts immediately. However, there were no indication that students understood or did not understand loops.

In the second half of the class, asking students to define terms in their own words was a better way to engage students in activities important to their understanding of key terms. In hindsight, it was a good way to start the second week of the summer camp.

If we were to repeat this class again, day six would look a little different. Instead of beginning the class with coding, we would start the class with term definitions before moving onto the coding segment. Since the students are high school students and fresh from being picked up and dropped off, it would be easier to ease them into the day. Additionally, we suggest creating a formative assessment during the coding segment of the class to check for student understanding.

Critical Reflection for Day 8

At first, we thought that using a pre-written program to demonstrate two-dimensional spaces via a character array was too complex for students to grasp. However, we were proven wrong in our assumptions. Students worked through the problem quickly and grasped the concept of nested loops. When questioned about the character array, students explained the overarching concept of a two-dimensional array with ease. The only concept students struggled with was the terminating conditions in the nested loops. However, when we explained that arrays started with zero as the initial index number, it made sense to them very quickly.

The results of the formative assessment were what we expected to see from novice students. It also served as a measurement of baseline knowledge of the students.

During the student activity, we checked the digital objects for free or creative commons licensure since this activity was the first time students put their knowledge to use. While the day's agenda and its overall progress was excellent, we did not anticipate the students' asking questions about other topics. For future iterations, we suggest taking a survey of the students to see what topics they would like to explore, should there be time left over at the end of the camp.

References

Fuselier, M. (2015, November 20). Using portfolios to recruit talent. *LinkedIn*. https://www.linkedin.com/pulse/using-portfolios-recruit-talent-fuselier-ms-cpa-rn-cns-fache/

Idaho State Department of Education. (n.d.). *Idaho content standards computer science.* https://www.sde.idaho.gov/academic/shared/computer-science/ICS-Computer-Science-Standards.pdf

Idaho State Department of Education. (2022a, January 3). *Idaho content standards mathematics*. https://www.sde.idaho.gov/topics/admin-rules/files/negotiated-rulemaking/7-Math-Proposed-Content-Standards-Draft2.pdf

Idaho State Department of Education. (2022b, January 3). *Idaho content standards science.* https://www.sde.idaho.gov/academic/shared/science/ics-science-legislative.pdf

Rube Goldberg machine. (2024, May 3). In *Wikipedia*. https://en.wikipedia.org/wiki/Rube_Goldberg_machine

Unity Documentation. (2024, June 14). *Unity manual.* https://docs.unity3d.com/Manual/2Dor3D.html

U.S. Department of Education. (2023, December 5). *Federal TRIO programs – Home page*. Retrieved December 17, 2023, from https://www2.ed.gov/about/offices/list/ope/trio/index.html

Vicon. (2022, April 21). *Nexus 2 tutorial – Simple workflow – Post processing* [Video]. YouTube. https://youtu.be/iv6BifOcW48?si=2QgfmIooC1JwXCNh

Support Materials

Bandicam. (n.d.). https://www.bandicam.com/

Blender. (n.d.). https://www.blender.org/

Epic Games. (n.d.) *Unreal engine*. https://www.unrealengine.com/en-US

Free3D. (n.d.). https://free3d.com/

OnlineGDB. (n.d.). https://www.onlinegdb.com/

Shutterstock. (n.d.). *Turbosquid*. https://www.turbosquid.com/

Sketchfab. (n.d.). https://sketchfab.com/

Unity. (n.d.). https://unity.com/

About the Authors

William Tai is a Ph.D. Candidate at the University of Idaho majoring in STEM Education. He has worked with the TRIO Upward Bound program as a summer program instructor since 2017. His research interests include integrated methods for secondary mathematics and secondary science for teacher preparation, cognitive demand in secondary mathematics and secondary science classrooms, and content knowledge analyses in mathematics education teacher preparation.

DeniRob Arnett is a master's student in Career and Technical Education and is the Outreach Coordinator for Benewah & Latah Upward Bound at the University of Idaho. He received his Bachelor of Science in Virtual Technology and Design from the University of Idaho. His skills include design, animation, and virtual world building. His goals are to cultivate a dynamic learning environment in Idaho, prepare students for success in their chosen industries, and to contribute to their journeys in higher education.

Sharing & Modification Permissions