



Statewide Framework Document for: 110803

**Video Game Design & Development**

Standards may be added to this document prior to submission but may not be removed from the framework to meet state credit equivalency requirements. Performance assessments may be developed at the local level. To earn state approval, performance assessments must be submitted within this framework. **This course is eligible for (1), third-year mathematics credit.**

Washington Mathematics Standards (Common Core State Standards) support foundational mathematical knowledge and reasoning. While it is important to develop a conceptual understanding of mathematical topics and fluency in numeracy and procedural skills, teachers should also focus on the application of mathematics to career fields to support the three (3) key shifts of CCSS. The Standards for Mathematical Practice develop mathematical habits of mind and are to be modeled and integrated throughout the course. The details about each mathematical standard can be found at [Common Core Mathematics Standards](http://www.corestandards.org/Math/).

The Washington State Computer Science K–12 Learning Standards are designed to enhance teacher understanding and improve student learning so that students are better equipped for college, career, and life. Washington is committed to implementing high-quality computer science instruction to:

* Increase the opportunity for all students to gain knowledge of computer science.
* Introduce the fundamental concepts and applications of computer science to all students, beginning at the elementary school level.
* Make computer science at the secondary level accessible, worthy of a computer science credit, and/or equivalent to math and science courses as a required graduation credit (see Level 3B of computer science standards).
* Offer additional secondary-level computer science instruction that allows interested students to study facets of computer science in-depth and prepare them for entry into a career or college.

|  |
| --- |
| **School District Name** |
| **Course Title:** Video Game Design & Development | **Total Framework Hours:** 180 Hours |
| **CIP Code:** 110803 | **[ ]** Exploratory **x** Preparatory | **Date Last Modified:** August 2025 |
| **Career Cluster:** Arts, A/V Technology & Communications  | **Cluster Pathway:** Visual Arts  |
| **Course Summary**: This comprehensive course introduces students to the fundamental principles of game design and development using Unity software, with a focus on creating engaging 2D games through a carefully structured blend of theory and practical application. Beginning with essential game design concepts, students learn core mechanics, player engagement strategies, and level design principles while simultaneously exploring the underlying mathematical concepts that drive game physics and movement. The curriculum weaves vector mathematics and geometric transformations into hands-on Unity projects, allowing students to learn how mathematical principles translate into gameplay mechanics. As students progress through physics systems, animation techniques, shader programming, and performance optimization, students apply trigonometry and linear algebra to create smooth character movements, realistic particle effects, and dynamic camera behaviors, while also learning professional project management skills through agile methodologies.  As the course progresses, students advance from basic Unity interface navigation and C# scripting to master programming content, with mathematics remaining a constant thread throughout the learning experience. Students apply higher level mathematical concepts to create smooth animations and use matrix transformations for sprite manipulations. The program culminates in professional beta testing and optimization techniques, where students use data analysis to evaluate gameplay metrics and performance data. Through this integrated approach of mathematical theory and practical application, students develop a deep understanding of both the technical and creative aspects of game development, building a portfolio that demonstrates their ability to create efficient, mathematically sound game systems that are both entertaining and technically sophisticated. This course meets the requirements for [RCW 28A.230.300](https://app.leg.wa.gov/RCW/default.aspx?cite=28A.230.300) and aligns with the [Unity Certified User: Programmer Exam](https://unity.com/products/unity-certifications#explore-certifications--2).**Unit 1:** The Components and Concepts of Game Design **Unit 2:** Unity Basics (Physics and Animation, Graphics and Shaders, Optimization and Performance)**Unit 3:** Introduction to Project Management in Game Design**Unit 4:** Geometry for Animation in Unity Game Development **Unit 5:** Introduction to 2D Video Game Programming (Scripting and Programming, Game Mechanics and Design)**Unit 6:** Mastering 2D Game Programming in Unity**Unit 7:** Beta Testing and Game Optimization  |
| **Eligible for Equivalent Credit in 1.0 Third Year Math** | **Total Number of Units:** 7 |
| **Course Resources:**  |

*To duplicate this blank table (for additional units), select the table, select copy, place cursor below the first table, and select paste.*

|  |  |
| --- | --- |
| **Unit 1:** The Components and Concepts of Game Design  | **Total Learning Hours for Unit:** 15 |
| **Unit Summary**: This unit provides a comprehensive foundation in the ethical, legal, and conceptual aspects of game development. It prepares students for the technical aspects of game design by ensuring they understand the broader context, vocabulary, mathematical relationship, and the responsibilities of game developers. The backwards engineering activities help students begin to think analytically about game design, setting the stage for their own creative work in the course. |
| **Performance Assessments**:Week 1: Digital Safety and Software Development Ethics Objective 1: Understand and apply digital safety practices Objective 2: Recognize and discuss software development ethics Topics: Online privacy and security Safe practices for game developers Ethical considerations in game development Responsible AI and data usage Activities: Using a district approved AI Assistant that uses natural language processing and generation to communicate, such as CoPilot or Gemini, students work in small groups to develop prompts, analyze feedback, and log key points found related to digital safety in online, competitive gaming Group discussions on ethical dilemmas in game development Create a digital safety guide for game developers Performance Assessment: Individual assignment: Write a personal code of ethics for game development in journal Group project: Develop and publish a digital safety guide for game developers (class project)  Week 2: Game Design Language and Concepts Objective 3: Use appropriate language and concepts related to game design Objective 4: Create storyboards and flowcharts for game concepts Topics: Basic game design terminology Backward design process Elements of game design (mechanics, dynamics, aesthetics) Storyboarding techniques Creating flowcharts for game logic Activities: Vocabulary building exercises with game design terms Backwards engineer through [Geometry Reconstruction Activities](https://waospi-my.sharepoint.com/personal/holli_ryan-kalaleh_k12_wa_us/Documents/Equivalency/eSports%2024-25/UNIT%201_Geometry%20Reconstruction%20Activities.docx?web=1) Analyze a [popular game using the MDA (Mechanics, Dynamics, Aesthetics) framework](https://waospi-my.sharepoint.com/personal/holli_ryan-kalaleh_k12_wa_us/Documents/Equivalency/eSports%2024-25/UNIT%201_MDA%20Analysis.docx?web=1) Create a storyboard for a simple game concept Develop a flowchart for a game's main menu system Performance Assessment: Individual assignment:  1. Submission of one or more Geometry Reconstruction Activities demonstrating understanding of geometric concepts, critical thinking, and problem-solving skills for backwards engineering.
2. Reflection on how understanding MDA can improve gameplay and the player experience

Group project: Create a storyboard and matching flowchart for a fictitious game’s main menu system using a medium of choice with presentation or publication to the class. Week 3: Legal Considerations and Game Analysis Objective 5: Understand electronic use policies, copyright laws, and plagiarism  Objective 6: Analyze existing games through backwards engineering Topics: Acceptable use policies in schools and workplaces Basics of copyright law and fair use Understanding and avoiding plagiarism Techniques for analyzing and backwards engineering games Activities: Review and discuss sample electronic use policies Case studies on copyright issues in game development Practice citing sources and assets properly Backwards engineer a simple game Backwards Engineering Activity: "Deconstructing Pong" 1. Students play the classic game Pong
2. In groups, they identify and list all the game elements (paddles, ball, scoring, etc.)
3. Create a flowchart of the game's logic
4. Describe the basic physics involved (ball movement, collisions)
5. Sketch a possible code structure for the game
6. Present their analysis to the class, discussing how they would recreate the game

Performance Assessments: Individual assignment:  1. Using personal journal, write an entry explaining the key considerations a game developer must know related to copyright and fair use
2. Analyze the district or school acceptable use policy and explain why schools have these policies

Group project: Backwards engineer a simple mobile game and present findings FOR CONSIDERATION: Ethics scenario of the day - Brief discussions on various ethical situations in game development |
| **Leadership Alignment**: *1.A THINKING CREATIVELY: The student will be involved in activities, such as the backward design process, that require applying theory, problem-solving, and using critical and creative thinking skills while understanding outcomes of related decisions.**4.A.1 The student will demonstrate their ability to access information efficiently (time) and effectively (sources) as they review electronic use policies and evaluate ethical dilemmas.**4.A.2 Students will show their ability to evaluate information critically and competently as they develop a digital safety guide for game developers and describe fair use and copyright laws.* |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.2.a: Manage their digital identity and understand the lasting impact of their online behaviors... make safe, legal and ethical decisions 1.2.d: Take action to protect their digital privacy on devices and manage their personal data and security, make safe, legal and ethical decisions in the digital world 1.3.d: Build knowledge by actively exploring real-world issues and problems, developing ideas and theories 1.3.b: Evaluate the accuracy, validity, bias, origin, and relevance of digital content 1.1.d: Understand fundamental concepts of how technology works" 1.6.a: Choose the appropriate platforms and tools for meeting the desired objectives 1.6.c: Use digital tools to visually communicate complex ideas to others 1.4.a: Know and use a deliberate design process for generating ideas 1.4.b: Select and use digital tools to plan and manage a design process 1.5.c: Break problems into component parts, extract key information, and develop descriptive models 1.6.b: Create original works or responsibly repurpose or remix digital resources into new creations 1.5.a: Formulate problem definitions suited for technology-assisted methods 1.5.c: Break problems into component parts, extract key information, and develop descriptive models to understand complex systems 1.4.d: Exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems 1.3.d: Build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions  |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-NI-06 - Recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. 3B-IC-26 - Evaluate the impact of equity, access, and influence on the distribution of computing resources in a global society. 3B-IC-25 - Evaluate computational artifacts to maximize their beneficial effects and minimize harmful effects on society. 3A-AP-12 - Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-IC-30 - Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.  3A-AP-22 - Design and develop computational artifacts working in team roles using collaborative tools.  |
| **Educational Technology** | 1.2.a - Practice and demonstrate safe, legal, and ethical behavior in the use of information and technology. 2.3.a - Select and use appropriate digital tools or resources to plan, develop, and evaluate solutions to authentic problems. 2.4.a - Evaluate and analyze the ethical and legal implications surrounding access to, and use of, information and technology. 1.2.b - Analyze the capabilities and limitations of current and emerging technologies and assess their potential to address personal, social, lifelong learning, and career needs. 1.3.a - Use models and simulations to explore systems, identify trends, and forecast possibilities. 1.1.a - Generate ideas and create original works for personal and group expression using a variety of digital tools. 2.1.a - Communicate and collaborate using digital tools to support individual learning and to contribute to the learning of others. 1.3.b - Analyze how technology shapes the way information is created, communicated, and used. 2.2.a - Develop cultural understanding and global awareness by engaging with learners of other cultures through digital tools.  |
| **Mathematics: Common Core** | HSG.CO.A.1 - Know precise definitions of angle, circle, perpendicular line, parallel line, and line segment, based on the undefined notions of point, line, distance along a line, and distance around a circular arc. (This can be applied to understanding geometric concepts in game design) HSF.IF.C.7 - Graph functions expressed symbolically and show key features of the graph, by hand in simple cases and using technology for more complicated cases. (This can be applied when creating flowcharts for game logic) HS.CED.A.3 - Represent constraints by equations or inequalities, and by systems of equations and/or inequalities, and interpret solutions as viable or non-viable options in a modeling context. (This can be applied to modeling game mechanics and rules) HSF.BF.A.1 - Write a function that describes a relationship between two quantities. (This can be used in describing relationships between game variables)  |

|  |  |
| --- | --- |
| **Unit 2:** Unity Basics (Physics and Animation, Graphics and Shaders, Optimization and Performance) | **Total Learning Hours for Unit:** 25 Hours |
| **Unit Summary**: This unit provides a comprehensive introduction to the foundational elements of Unity software used in game development/programming, balancing guided instruction with creative exploration. It's designed to give students a solid grounding in the Unity interface and basic concepts, preparing them for more advanced game development topics in future units. The hands-on approach and creative projects encourage students to experiment and develop their skills while building a practical understanding of Unity's capabilities. This unit covers the following concepts: How to navigate and utilize the Unity interface effectively; Create and manipulate GameObjects, Components, and Prefabs; Import and use assets from the Asset Store; Implement basic physics using Rigidbody components and colliders; Create simple animations using Animator controllers and animation clips; Apply materials, shaders, and lighting to enhance scene visuals; Utilize textures and post-processing effects; Optimize scenes for performance.  |
| **Performance Assessments**:**Week 1: Unity Interface and GameObjects** Objective 1: Navigate and utilize the Unity interface effectively  Objective 2: Create and manipulate GameObjects, Components, and Prefabs Topics: Unity interface overview: Scene view, Game view, Inspector, Hierarchy Creating and manipulating GameObjects Developing the concept of 3D coordinate systems (x, y, z) and how they related to positioning GameObjects in Unity Understanding and using Components Creating and utilizing Prefabs Activities: Guided exploration of Unity interface Create a simple scene with multiple GameObjects placing objects at specific coordinates and calculate distances between objects using the distance formula. Add various Components to GameObjects Create and modify Prefabs Performance Assessments: Individual project: Create a themed scene (e.g., playground, alien planet) using GameObjects and Prefabs, demonstrate the ability to calculate distances between objects using the distance formula Pair project: Working with a partner, analyze the use of Prefabs and provide constructive and relevant feedback; students use rubric for evaluation **Week 2: Asset Store and Basic Physics** Objective 3: Import and use assets from the Asset Store  Objective 4: Implement basic physics using Rigidbody components and colliders Topics: Navigating and using the Asset Store Importing assets into Unity projects Introduction to Rigidbody physics ​​​Explain how vectors are used to represent forces in Unity’s physics system Discover basic ​​projectile motion equations and how they relate to game physics Types of colliders and their uses Activities: Import and use assets from the Asset Store in a scene Experiment with Rigidbody properties (mass, drag, gravity) by ​​applying different force vectors to objects and predict their trajectories Create a simple physics-based game mechanic, such as a projectile launcher and predict where objects will land Performance Assessments: Individual assessment: Journal entry sharing how feedback can affect creativity and progress in game design Group project: Design a small obstacle course using imported assets and physics interactions **Week 3: Animation Basics** Objective 5: Create simple animations using Animator controllers and animation clips Topics: Discuss how mathematical functions (interpolation and easing functions) are used to create smooth animations Creating animation clips Using the Animation window Setting up Animator controllers Basic animation state machines Activities: Experiment creating simple animations for GameObjects using different easing functions (linear, quadratic, cubic) and graph the effects on animation and timing Set up an Animator controller with multiple states Implement animation transitions based on parameters Performance Assessments: Individual assignment:  Animate a character or object with multiple states (e.g., idle, walk, jump) Graph the effects on animation and timing while using different easing functions (linear, quadratic, cubic) **Week 4: Materials, Shaders, and Lighting** Objective 6: Apply materials, shaders, and lighting to enhance scene visuals  Objective 7: Utilize textures and post-processing effects Topics: Creating and applying materials Basic shader properties Explain how colors are represented mathematically in digital systems Types of lights in Unity Discuss how the angle of incidence equals the angle of reflection in lighting calculations Applying textures to GameObjects Introduction to post-processing effects Activities: Experiment with different materials and shaders predict color mixing results Set up various lighting scenarios to demonstrate reflection angles and calculate them mathematically Apply textures to 3D models Add post-processing effects to a scene Performance Assessments: Individual project: Journal entry explaining how application of materials, shaders, and lighting can affect the experience of the game for players/spectators Group project: Create a visually appealing environment showcasing materials, lighting, and effects**Week 5: Optimization and Integration** Objective 8: Optimize scenes for performance Topics: Understanding frame rates and performance metrics through analysis of mean, median, and mode Optimizing draw calls Basic memory management in Unity Integrating all learned elements Activities: Use Unity Profiler to identify performance issues Collect performance data over time and have students calculate and interpret average metrics Implement basic optimization techniques Combine all learned elements into a cohesive scene Performance Assessments: Individual, Pair, or Group: Create a small, interactive Unity scene that incorporates all learned elements Scene must include custom GameObjects, physics interactions, animations, and visual enhancements Optimize the scene for performance Present their scene to the class, explaining their creative choices and techniques used FOR CONSIDERATION: *Ethics scenario of the day* - Brief discussions on various ethical situations when working on team projects  |
| **Leadership Alignment**: 5.B.1 By learning and applying basic Unity software tools, students will understand and utilize the most appropriate media creation tools, characteristics and conventions based on the level of integration for the game development  |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.1. Empowered Learner: * 1.1.c. use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.
* 1.1.d. understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

1.3. Knowledge Constructor: * 1.3.d. build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.

1.4. Innovative Designer: * 1.4.a. know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
* 1.4.b. select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
* 1.4.c. develop, test and refine prototypes as part of a cyclical design process.
* 1.4.d. exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

1.5. Computational Thinker: * 1.5.a. formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
* 1.5.c. break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
* 1.5.d. understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

1.6. Creative Communicator: * 1.6.a. choose the appropriate platforms and tools for meeting the desired objectives of their creation or communication.
* 1.6.b. create original works or responsibly repurpose or remix digital resources into new creations.
* 1.6.c. communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. 3A-AP-22: Design and develop computational artifacts working in team roles using collaborative tools. 3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. 3A-DA-10: Evaluate the tradeoffs in how data elements are organized and where data is stored.  |
| **Educational Technology** | 1.1: Demonstrate an understanding of intellectual property issues and use online resources ethically. 2.1: Communicate and collaborate using digital tools to support individual learning and contribute to the learning of others. 3.1: Locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media. 3.2: Plan strategies to guide inquiry using technology. 4.1: Identify and define authentic problems and significant questions for investigation. 4.2: Plan and manage activities to develop a solution or complete a project. 4.3: Collect and analyze data to identify solutions and make informed decisions. 5.1: Use digital tools to create original works as a means of personal or group expression. 5.2: Use digital tools and resources for analyzing data and identifying solutions. 6.1: Demonstrate a sound understanding of technology concepts, systems, and operations. 6.2: Select and use applications effectively and productively.  |
| **Mathematics: Common Core** | HS.N.VM.A.1 Recognize vector quantities as having both magnitude and direction. Represent vector quantities by directed line segments, and use appropriate symbols for vectors and their magnitudes (e.g. ***v, |v|, ||v||,*** *v*) HS.N.VM.A.3 Solve problems involving velocity and other quantities that can be represented by ​​​​vectors. ​​​HS.F.IF.A.2 Use function notation, evaluate functions for inputs in their momains and interpret statements that use function notation in terms of context.  HS.F.IF.C.7 Graph functions expressed symbolically and show key features of the graph by hand or using technology. ​​​​​HS.G.CO.A.2 Represent transformations in the plane using transparencies and geometry software, describe transformations as functions that take points in the plane as inputs and give other points as outputs. Compare transformations that preserve distance and angle to those that do not (e.g. translation versus horizontal stretch). HS.G.SRT.A.2 Given two figures, use the definition of similarity in terms of similarity transformations to decide if they are similar; explain using similarity transformations the meaning of similarity for triangles as the equality of all corresponding pairs of angles and proportionality of all corresponding pairs of sides. HSG-MG.A.3: Apply geometric methods to solve design problems. Aligns with: Objectives 2, 4, 6 (manipulating GameObjects, implementing physics, applying materials)  |

|  |  |
| --- | --- |
| **Unit 3:** Introduction to Project Management in Game Design | **Total Learning Hours for Unit:** 15 |
| **Unit Summary**: Students are introduced to the importance of project management in game design with the goal of understanding why effective project management is crucial for successful game development, including aspects like scope, budget, and timeline. Working in pairs or groups, students compare project management methodologies, learn about different project management approaches (such as Waterfall, Agile, or Scrum), and the relevance to game development and the team dynamic, Students gain skills in Gantt chart creation and interpretation as they learn to visually represent project tasks and timelines, and apply time management strategies to both personal and team projects to meet deadlines. Embedded throughout the unit is the use of basic statistical methods to inform project decisions, evaluate risks, and optimize game development processes.   |
| **Performance Assessments**:**Week 1: Introduction to Project Management and Scrum Methodology** Objective 1: Explain the importance of project management in game design  Objective 2: Describe and compare common project management methodologies Topics: Overview of project management in game design Introduce concepts of estimation and probability in project planning Introduction to Scrum methodology Roles in Scrum: Product Owner, Scrum Master, Development Team Scrum artifacts: Product Backlog, Sprint Backlog, Increment Activities: Using the tools learned in prior units (flowchart, software, storyboard), develop a basic game concept and configure it for version control using Git Performance Assessment: Individual quiz: Key concepts of Scrum methodology and basic probability in project planning Group activity: Simulate a Sprint Planning meeting for the simple game project, including task estimation **Week 2: Gantt Charts and Time Management** Objective 3: Create and interpret basic Gantt charts  Objective 4: Apply time management techniques to personal and team projects Topics: Unity timeline basics Introduction to Gantt charts Creating Gantt charts using spreadsheet software Time management techniques: Pomodoro Technique, Time blocking Prioritization methods: Eisenhower Matrix, MoSCoW method Performance Assessment: Individual project:  1. Create a Gantt chart for a hypothetical game development project
2. Create a simple cutscene or sequence using Unity’s Timeline feature

Group activity: Develop a time management plan for a two-week game design sprint **Week 3: Statistics and Data Analysis in Project Management** Objective 5: Utilize basic statistics and data analysis in project planning and evaluation Topics: Basic statistical concepts: mean, median, mode, standard deviation Data collection methods in game development projects Set up Unity Analytics in project and collect basic gameplay data Interpreting data to make project decisions Introduction to burn-down charts in Scrum Activities: Performance Assessment: Individual assignment: Analyze a dataset from a game project, make a design decision, and present findings in journal providing justification for decision based on data collection Group project: Create and interpret a burn-down chart for a simulated Sprint, incorporating data collected from Unity Analytics FOR CONSIDERATION: *Ethics scenario of the day* - Brief discussions on various ethical situations when communication is flawed (missing, unclear, incorrect) and the impact on individuals and/or teams. |
| **Leadership Alignment**: 1.B Work Creatively with Others: As students develop their understanding of project management, they will demonstrate the ability to incorporate and utilize the principles of group dynamics in a variety of settings, including in classrooms, small project-groups, and correlate the connection from the classroom to industry. 3.B Collaborate with Others: Students will demonstrate their ability to communicate, participate, and advocate effectively in pairs, small groups, teams, and large groups in order to reach common goals while maintaining respectful dialog and effective progress as they develop and analyze their team projects.  |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.2. Digital Citizen:* 1.2.b. engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.

1.5. Computational Thinker:* 1.5.b. collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.

1.7. Global Collaborator:* 1.7.b. use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
* 1.7.c. contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.
* 1.7.d. explore local and global issues and use collaborative technologies to work with others to investigate solutions.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.  3B-AP-17: Plan and develop programs for broad audiences using a software lifecycle process.  |
| **Educational Technology** | 2.1: Communicate and collaborate to learn with others. 2.2: Develop cultural understanding and global awareness by engaging with learners of many cultures. 3.1: Identify and define authentic problems and significant questions for investigation and plan strategies to guide inquiry. 3.2: Locate and organize information from a variety of sources and media. 4.1: Use multiple processes and diverse perspectives to explore alternative solutions. 4.2: Collect and analyze data to identify solutions and make informed decisions. 4.3: Use models and simulations to explore systems, identify trends, and forecast possibilities.  |
| **Mathematics: Common Core** | Algebra: HS.A.REI.10: Understand that the graph of an equation in two variables is the set of all its solutions plotted in the coordinate plane. Aligns with: Objective 3 (Creating and interpreting Gantt charts) Statistics and Probability: S-ID.1: Represent data with plots on the real number line (dot plots, histograms, and box plots). Aligns with: Objective 5 (Data analysis and burn-down charts) S-ID.2: Use statistics appropriate to the shape of the data distribution to compare center and spread of two or more different data sets. Aligns with: Objective 5 (Basic statistical concepts and interpreting data) S-ID.3: Interpret differences in shape, center, and spread in the context of the data sets, accounting for possible effects of extreme data points (outliers). Aligns with: Objective 5 (Interpreting data to make project decisions)  |

|  |  |
| --- | --- |
| **Unit 4:** Geometry for Animation in Unity Game Development  | **Total Learning Hours for Unit:** 25 |
| **Unit Summary**: In this unit, students will acquire a range of skills related to Unity-based game development. They will focus on how to apply project management techniques learned in unit 3, specifically within the context of game development projects. Additionally, they will explore fundamental geometric principles relevant to creating game assets and environments in both two-dimensional (2D) and three-dimensional (3D) spaces. Students will gain the ability to implement animations by manipulating game objects through geometric transformations within Unity’s interface. Finally, they’ll apply their knowledge of geometry and animation to develop a basic game prototype, integrating these principles into their project.  |
| **Performance Assessments**:TO MAINTAIN AND EXPAND PROJECT MANAGEMENT SKILLS: During all weeks of this unit, student groups should conduct daily stand-up meetings and weekly sprint reviews; maintain a Scrum board throughout the development process; create and update burn-down charts to track project progress. **Week 1: Integrating Project Management in Game Design** This week will focus on applying all learned concepts to a mini game design project. Activities & Performance Assessments:  PLANNING PHASE| Students will work in small teams to: 1. Plan a simple game design project using Scrum methodology (students may build upon previous games developed or create a new game)
2. Create a Gantt chart for the project
3. Implement time management techniques during development
4. Present their project and management process to the class
5. Version Control with Unity Collaborate: Set up Unity Collaborate (practice branching and merging workflows)

**Week 2: Project Setup and Basic Unity Interface** Objective 1: Apply project management skills to a Unity-based game development project  Objective 2: Create and manipulate game objects using Unity's interface Topics: Setting up a Unity project using Scrum methodology Unity interface overview: Scene view, Game view, Inspector, Hierarchy Creating and manipulating basic game objects Introduction to Unity's coordinate system Basic Unity Scripting with C# Activity: Introduce 2D and 3D vectors for positioning and movement in Unity Create objects that move along vector paths, calculating distances and velocities Write simple scripts to manipulate game object transforms using vector math Performance Assessments: Individual assignment: Set up a Unity scene with basic geometric shapes, implementing vector-based positioning and rotation scripts Group activity: Create a project backlog for group project (Unity game) **Week 3: 2D Geometry and Transformations** Objective 3: Create (programmatically using mathematical functions), import, and manipulate Sprites in Unity  Objective 4: Apply 2D physics and collision detection to sprites Objective 5: Understand sprite rendering order and sorting layers Topic:  Introduction to Sprites in Unity Activities: Sprite Creation and Manipulation:  * Students create a simple character using a sprite editor tool (like Piskel or Aseprite).
* Import the created sprite into Unity.
* Learn to slice sprite sheets in Unity's Sprite Editor.
* Create animations using Unity's Animation window with the sliced sprites.

Procedural Sprite Generation * Use sine waves to generate terrain outlines
* Implement ​​​​simple fractals (e.g., Koch snowflake) as sprite shapes

Sprite Rendering and Sorting:  * Set up a 2D scene with multiple background layers using sprites.
* Experiment with Sprite Renderer component properties.
* Learn to use Sorting Layers and Order in Layer to control rendering order.

Performance Assessment: Individual project:  1. Create a 2D character with idle, walk, and jump animations using sprites
2. Create a procedurally generated 2D level using mathematical functions

Group activity: Create a parallax scrolling background effect using multiple sprite layers. **Week 4: 3D Geometry and Animations** Objective 6: Explain and utilize basic geometric concepts in 3D game environments  Objective 7: Implement basic animations using geometric transformations in Unity Topics: D coordinate systems and vector operations 3D transformations and rotations (Euler angles, quaternions) Unity's Animation system basics Rigging and animating a simple 3D character Performance Assessments: Individual assignment: Create a 3D environment with animated camera movement Group project: Develop and animate a simple 3D game character **Week 5: Game Prototype Development** Enhanced Final Project: Individual project: Journal data and feedback from presentation and share ideas for improvement Group project: Include the following sprite-related requirements in the final game prototype: 1. Custom Sprite Creation: Students must create at least one custom sprite for their game (character, item, or environmental element).
2. Sprite Animation: Implement at least one animated sprite sequence in the game (e.g., character movement, item pickup animation).
3. Sprite-based UI Elements: Use sprites to create at least two UI elements (e.g., health bar, score display, menu buttons).
4. Advanced Sprite Techniques (Optional Challenge): Encourage advanced students to explore:
* Sprite masking for special effects
* Procedural sprite generation using Unity's Scripting API
* Sprite-based particle systems
1. Present final project (game) to class
2. Analyze data and feedback

 FOR CONSIDERATION: *Ethics scenario of the day* - Brief discussions on using Artificial Intelligence in the workplace  |
| **Leadership Alignment**: 3.A Communicate Clearly: Students will demonstrate oral, interpersonal, written, and electronic communication and presentation skills, and demonstrate understanding of how to apply those skills as they develop their unit project, analyze data and feedback, and present the project to the class. 5.A.2 The student outcome will include the ability to examine how individuals interpret messages differently, how values and points of view are included or excluded, and how media can influence beliefs and behaviors.  |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.1. Empowered Learner:* 1.1.a. articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.
* 1.1.b. build networks and customize their learning environments in ways that support the learning process.

1.2. Digital Citizen:* 1.2.c. demonstrate an understanding of and respect for the rights and obligations of using and sharing intellectual property.

1.3. Knowledge Constructor:* 1.3.a. plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits.
* 1.3.b. evaluate the accuracy, perspective, credibility and relevance of information, media, data or other resources.
* 1.3.c. curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-AP-19 - Systematically design and develop programs for broad audiences by incorporating feedback from users. 3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects. 3A-CS-02: Compare levels of abstraction and interactions between application software, system software, and hardware layers.  |
| **Educational Technology** | Digital Citizenship (DC)  1.1.1 Demonstrate ethical and respectful behavior when using technology.  1.1.2 Identify legal, ethical, and safe behaviors related to technology.  1.2.1 Demonstrate compliance with the school's Acceptable Use Policy (AUP). Communication and Collaboration (CC)  2.1.1 Communicate and collaborate using digital tools.  2.1.2 Use digital tools to gather and share information. Research and Information Fluency (RIF)  3.1.1 Use digital tools to identify and define authentic problems and significant questions for investigation.  3.1.2 Plan strategies to guide inquiry using digital tools. Critical Thinking, Problem Solving, and Decision Making (CTPS)  4.1.1 Identify and define authentic problems and significant questions for investigation.  4.1.2 Plan and manage activities to develop a solution or complete a project. Technology Operations and Concepts (TOC)  5.1.1 Recognize, define, and use technology processes, systems, and applications.  5.1.2 Select and use applications effectively and productively. Creativity and Innovation (CI)  6.1.1 Create original works as a means of personal or group expression.  6.1.2 Use models and simulations to explore complex systems and issues.  |
| **Mathematics: Common Core** | HS.N.VM.A.1 Recognize vector quantities as having both magnitude and direction. Represent vector quantities by directed line segments, and use appropriate symbols for vectors and their magnitudes (e.g. ***v, |v|, ||v||,*** *v*) HS.N.VM.A.3 Solve problems involving velocity and other quantities that can be represented by ​￼​​￼​vectors. HS.A.REI.10: Understand that the graph of an equation in two variables is the set of all its solutions plotted in the coordinate plane. Aligns with: Objective 3 (Creating and interpreting Gantt charts) HS.F.TF.B.5 Choose trigonometric functions to model periodic phenomena with specified amplitude, frequency, and midline. HS.G.CO.A.2 - Represent transformations in the plane using, e.g., transparencies and geometry software; describe transformations as functions that take points in the plane as inputs and give other points as outputs. Compare transformations that preserve distance and angle to those that do not. HS.G.CO.A.5 - Given a geometric figure and a rotation, reflection, or translation, draw the transformed figure using, e.g., graph paper, tracing paper, or geometry software. Specify a sequence of transformations that will carry a given figure onto another. HS.G.GPE.B.4 - Use coordinates to prove simple geometric theorems algebraically. |

|  |  |
| --- | --- |
| **Unit 5:** Introduction to 2D Video Game Programming (Scripting and Programming, Game Mechanics and Design) | **Total Learning Hours for Unit:** 50  |
| **Unit Summary**: Time to increase essential skills for Unity game development. Students learn to write basic C# scripts tailored for Unity projects, implement MonoBehaviour scripts, and attach them to GameObjects. Students further develop skills using variables, functions, and events within Unity scripts, and dive into creating game loops, handling input, and designing player controls. A primary goal is for students to gain proficiency in level design, camera movement controls, and UI element creation, plus explore particle systems and learn how to integrate audio effectively into Unity games. |
| **Performance Assessments**:Week 1: Introduction to C# and MonoBehaviour Scripts Objective 1: Write basic C# scripts for Unity projects  Objective 2: Implement MonoBehaviour scripts and attach them to GameObjects Topics: C# basics: variables, data types, operators, control structures MonoBehaviour class and its lifecycle methods (Start, Update, etc.) Creating and attaching scripts to GameObjects Using the Unity documentation Activities: Interactive C# coding exercises in Unity Create a simple script to move a GameObject Experiment with different MonoBehaviour methods Performance Assessment: Individual project: Create a script that changes a GameObject's properties over time Individual quiz: C# basics and MonoBehaviour concepts **Week 2: Variables, Functions, and Events in Unity** Objective 3: Use variables, functions, and events in Unity scripts Topics: Public vs. private variables Functions and methods in C# Unity's event system (OnCollisionEnter, OnTriggerExit, etc.) Coroutines for time-based events Activities: Create scripts using various types of variables and functions Implement collision and trigger events Use coroutines to create timed game mechanics Performance Assessment: Individual assignments:  1. Develop a simple collection game using events and coroutines
2. Write a script demonstrating the use of different variable types and functions

**Week 3: Game Loops, Input Handling, and Player Controls** Objective 4: Implement game loops, handle input, and create player controls Topics: Understanding the game loop (FixedUpdate vs Update) Input management in Unity (Input.GetKey, Input.GetAxis, etc.) Implementing basic player movement and actions Introduction to Unity's new Input System Activities: Create a player controller script with keyboard and mouse input Implement a simple game state machine Experiment with different types of player movement (2D platformer, top-down, first-person) Performance Assessment: Individual project: Develop a character controller with multiple movement types Group activity: Create a mini-game focusing on unique input mechanics **Week 4: Level Design, Camera Movement, and UI Elements** Objective 5: Design basic levels, control camera movement, and create UI elements Topics: Principles of level design in Unity Camera control scripts (follow player, orbit, etc.) Creating UI elements (buttons, text, health bars) Canvas and anchors in Unity UI Activities: Design and implement a simple game level Create a camera control script Develop a game UI with interactive elements Performance Assessment: Individual assignment: Create a menu system for a game Group project: Design a level with custom camera behavior and UI Individual assignment: Create a menu system for a game **Week 5: Particle Systems and Audio Integration** Objective 6: Utilize particle systems and integrate audio in Unity games Topics: Creating and customizing particle systems Audio sources and listeners 2D and 3D sound in Unity Basic audio mixing and management Activities: Create particle effects for various game scenarios (explosions, environmental effects) Implement background music and sound effects in a game scene Experiment with 3D audio positioning Performance Assessments: Individual assignment: Students journal how the feedback received from other students will guide future game design projects Group assignment: Students will work in small teams to: 1. Develop a small game that incorporates all learned elements
2. Implement custom scripts for game mechanics, UI, and audio
3. Design at least one level with custom particle effects and sound
4. Create a player controller with smooth movement and camera follow
5. Present their game to another team/group and explain the scripting concepts used
6. Students will evaluate opposing team’s project and provide analysis based on the following criteria:
* Quality and efficiency of C# scripts (20%)
* Implementation of game mechanics and player controls (20%)
* Level design and camera work (15%)
* UI design and functionality (15%)
* Use of particle systems and audio (15%)
* Presentation and explanation of concepts (15%).

FOR CONSIDERATION: *Ethics scenario of the day* - Brief discussions on ethical behaviors at the corporate level  |
| **Leadership Alignment**: 5.B.1 Create Media Products: Understand and utilize the most appropriate media creation tools, characteristics and conventions as students create a small game that incorporates learned elements 5.B.2 Create Media Products: Understand and effectively utilize the most appropriate expressions and interpretations in diverse, multi-cultural environments while working in their teams and modeling appropriate communication while establishing ethical practices |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.2. Digital Citizen:* 1.2.a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.
* 1.2.d. manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.

1.6. Creative Communicator:* 1.6.d. publish or present content that customizes the message and medium for their intended audiences.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3B-AP-20: Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project. 3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.  3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. 3A-AP-15: Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. 3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. 3A-AP-20: Evaluate licenses that limit or restrict use of computational artifacts when using resources such as libraries. 3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.  |
| **Educational Technology** | Digital Citizenship Standard 1.1 - Innovate: Demonstrate creative thinking, construct knowledge and develop innovative products and processes using technology. * 1.1.HS.1: Apply existing knowledge to generate new ideas and processes.
* 1.1.HS.2: Create original works as a means of personal or group expression.

Standard 1.2 - Collaborate: Use digital media and environments to communicate and work collaboratively to support individual learning and contribute to the learning of others. * 1.2.HS.1: Interact and collaborate with peers, experts, or others employing a variety of digital environments and media.

Digital Knowledge and Skills Standard 2.1 - Practice: Demonstrate a sound understanding of technology concepts, systems, and operations. * 2.1.HS.1: Understand and use technology systems.
* 2.1.HS.2: Select and use applications effectively and productively.
* 2.1.HS.3: Troubleshoot systems and applications.

Standard 2.2 - Plan: Develop skills to use technology effectively and productively. * 2.2.HS.1: Identify capabilities and limitations of contemporary and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.

Standard 2.3 - Design: Design and produce digital materials addressing a range of purposes. * 2.3.HS.1: Design, develop, and test a digital learning game to demonstrate knowledge and skills related to curriculum content.

Standard 2.4 - Evaluate: Use critical thinking skills to conduct research, solve problems, and make informed decisions using appropriate technological tools and resources. * 2.4.HS.1: Plan and manage activities to develop a solution or complete a project.
* 2.4.HS.2: Collect and analyze data to identify solutions and/or make informed decisions.
 |
| **Mathematics: Common Core** | HSA-CED.A.2: Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales. HSF-IF.A.1: Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. HSF-BF.A.1: Write a function that describes a relationship between two quantities. HSG-MG.A.1: Use geometric shapes, their measures, and their properties to describe objects. HSG-CO.A.2: Represent transformations in the plane using, e.g., transparencies and geometry software; describe transformations as functions that take points in the plane as inputs and give other points as outputs. HSS-ID.A.1: Represent data with plots on the real number line (dot plots, histograms, and box plots). |

|  |  |
| --- | --- |
| **Unit 6:** Mastering 2D Game Programming in Unity | **Total Learning Hours for Unit: 30**  |
| **Unit Summary**: Emphasis is placed on increasing individual proficiency and creativity in various aspects of 2D game development using Unity. Students learn to design and implement complex game mechanics, manage physics-based interactions, create advanced character controllers, and develop tile-based level designs. Students increase knowledge of game state management, optimize performance, and integrate advanced UI and audio systems into their 2D games.  |
| **Performance Assessments:****Week 1: Advanced 2D Game Mechanics** Objective 1: Design and implement complex 2D game mechanics Topics: Vector math for 2D movement Implementing gravity and jump mechanics Creating power-ups and special abilities Developing enemy AI patterns Activities: Enhance a basic platformer with double jump and wall jump abilities Create a power-up system with temporary effects Implement a simple enemy AI with patrolling and chasing behaviors Performance Assessment: Individual project: Develop a unique game mechanic and implement it in Unity **Week 2: 2D Physics and Interactions** Objective 2: Create and manage 2D physics-based interactions Topics: Rigidbody2D and Collider2D components Raycasting in 2D Joints and hinges for 2D objects Particle systems for 2D effects Activities: Create a physics-based puzzle game element (e.g., balance beam, swinging platforms) Implement a grappling hook mechanic using physics joints Develop a destruction system with physics and particle effects Performance Assessment: Individual or Group Project: Design and implement a level featuring multiple physics-based interactions**Week 3: Advanced 2D Character Controllers** Objective 3: Develop advanced 2D character controllers Topics: State machines for character behavior Animation blending and transitions Implementing character abilities (dash, glide, etc.) Handling various movement types (platformer, top-down, isometric) Activities: Create a state machine for a character with multiple animations Implement advanced movement abilities (air dash, wall slide, etc.) Develop a character that can switch between movement types Performance Assessment: Individual assignment: Design and implement a unique character controller with multiple abilities **Week 4: Tile-based Level Design** Objective 4: Implement tile-based level design Topics: Using Unity's Tilemap system Creating and using Rule Tiles Procedural level generation basics Optimizing large 2D levels Activities: Design a set of tiles and create a small level using Tilemaps Implement Rule Tiles for auto-tiling Create a simple procedural level generator Performance Assessment: Group project: Design and implement a large level using Tilemaps, including some procedurally generated elements **Week 5: Game State and Progression** Objective 5: Create and manage game state and progression Topics: Implementing save/load systems Creating an inventory system Designing progression mechanics (XP, leveling, skill trees) Managing multiple scenes and level transitions Activities: Implement a save/load system using PlayerPrefs and JSON Create a simple inventory system with item usage Develop a basic skill tree with unlockable abilities Performance Assessment: Individual project: Design and implement a progression system for a game concept **Week 6: Optimization, UI, and Audio** Objectives 6 and 7: Optimize 2D game performance and integrate advanced UI and audio systems Topics: Profiling and optimizing 2D games Object pooling for performance Creating responsive UI for various screen sizes Advanced audio management (mixer, spatial audio in 2D) Activities: Use the Unity Profiler to identify and fix performance issues Implement object pooling for frequently spawned objects Create a dynamic UI that adapts to different resolutions Set up a mixer with multiple audio layers and effects Final Project (Performance Assessment): Students will work in small teams to: 1. Develop a complete 2D game incorporating all learned elements
2. Implement advanced mechanics, physics interactions, and character control
3. Design levels using the Tilemap system, including some procedural elements
4. Create a progression system with save/load functionality
5. Optimize the game for performance
6. Implement polished UI and audio systems
7. Explain the advanced concepts used in their individual journals
 |
| **Leadership Alignment**: 8.A Manage Goals and Time: Students will work independently to complete a 2D game while managing their time and demonstrating understanding of concepts and self-advocacy by achieving planned, individual goals. 8.B Work Independently: Students will demonstrate self-advocacy skills by achieving planned, individual goals while working on implementing specified new elements into their projects 9.A.1 & 9.A.2 During the team project phase, students will demonstrate their ability to know when it is appropriate to listen and when to speak and conduct themselves in a respectful, professional manner |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.2. Digital Citizen:* 1.2.a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.
* 1.2.d. manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.

1.6. Creative Communicator:* 1.6.d. publish or present content that customizes the message and medium for their intended audiences.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests. 3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables. 3A-AP-15: Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made. 3A-AP-16: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. 3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. 3A-DA-12: Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.  |
| **Educational Technology** | * 1. Empowered Learner

EL.1.a: Articulate personal learning goals, select and manage appropriate technologies to achieve them, and reflect on their successes and areas of improvement in working toward their goals. EL.1.c: Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. 1. Digital Citizen

DC.1.b: Manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online. 1. Knowledge Constructor

KC.1.a: Plan and employ effective research strategies to locate information and other resources for their intellectual or creative pursuits. KC.1.c: Curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections or conclusions. 1. Innovative Designer

ID.1.a: Know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. ID.1.b: Select and use digital tools to plan and manage a design process that considers design constraints and calculated risks. ID.1.c: Develop, test and refine prototypes as part of a cyclical design process. 1. Computational Thinker

CT.1.a: Formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. CT.1.b: Collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. CT.1.c: Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. 1. Creative Communicator

CC.1.b: Create original works or responsibly repurpose or remix digital resources into new creations. CC.1.d: Publish or present content that customizes the message and medium for their intended audiences.  |
| **Mathematics: Common Core** | Number and Quantity (N) N.VM.A.1: Recognize vector quantities as having both magnitude and direction. Represent vector quantities by directed line segments, and use appropriate symbols for vectors and their magnitudes. N.VM.A.2: Find the components of a vector by subtracting the coordinates of an initial point from the coordinates of a terminal point. N.VM.A.3: Solve problems involving velocity and other quantities that can be represented by vectors. Algebra (A) A.CED.A.2: Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales. A.REI.D.10: Understand that the graph of an equation in two variables is the set of all its solutions plotted in the coordinate plane. Functions (F) F.IF.B.4: For a function that models a relationship between two quantities, interpret key features of graphs and tables in terms of the quantities, and sketch graphs showing key features. F.BF.A.1: Write a function that describes a relationship between two quantities. Geometry (G) G.SRT.C.8: Use trigonometric ratios and the Pythagorean Theorem to solve right triangles in applied problems. G.GPE.B.4: Use coordinates to prove simple geometric theorems algebraically. G.GPE.B.6: Find the point on a directed line segment between two given points that partitions the segment in a given ratio. Statistics and Probability (S) S.MD.B.6: Use probabilities to make fair decisions. S.MD.B.7: Analyze decisions and strategies using probability concepts |

|  |  |
| --- | --- |
| **Unit 7:** Beta Testing and Game Optimization  | **Total Learning Hours for Unit:** 20 |
| **Unit Summary**: This unit provides a comprehensive approach to beta testing, data analysis, optimization, and deployment, while also emphasizing the importance of reflection in the game development process. It builds upon the technical skills developed in previous units, applying them in a real-world testing and optimization context. The unit structure allows for both individual skill development and collaborative project work, culminating in a polished, tested, and deployable game project. At the end of this unit students should be able to design and conduct effective beta tests for their games; collect, analyze, and interpret player feedback, and game data; optimize game performance based on testing results; prepare games for deployment on various platforms; and reflect on the game development process and outcomes. |
| **Performance Assessments**:Throughout the Unit: * Daily reflection journals: Students spend 5-10 minutes at the end of each class reflecting on their learning and challenges
* Weekly progress reports: Teams submit brief reports on their project status, including data insights and optimization plans.

**Week 1: Beta Test Design and Player Feedback** Objective 1: Design and conduct effective beta tests for their games Topics: Principles of beta testing in game development Creating test plans and scenarios Designing effective player surveys Ethics in player testing Activities: Develop a beta test plan for their game projects Create a player feedback survey Conduct an in-class beta test session Performance Assessments: Individual assignment: Design a comprehensive beta test plan Group activity: Peer review of beta test plans **Week 2: Data Collection and Analysis** Objective 2: Collect, analyze, and interpret player feedback and game data Topics: Implementing analytics in Unity games Basic statistical analysis of player data Visualizing game data (charts, heatmaps) Interpreting qualitative feedbackActivities: Implement basic analytics in their game projects Analyze provided sample game data using spreadsheet software Create visualizations of game data Performance Assessments: Individual project: Analyze and present findings from a provided dataset Group activity: Conduct and analyze results from peer beta tests **Week 3: Optimization and Performance Tuning** Objective 3: Optimize game performance based on testing results Topics: Common performance issues in Unity games Using the Unity Profiler Optimization techniques (object pooling, draw call reduction, etc.) Balancing game mechanics based on player data Activities: Profile their games using Unity Profiler Implement optimization techniques in their projects Adjust game mechanics based on beta test feedback Performance Assessments: Individual assignment: Optimize a provided sample scene for performance Group project: Implement and document optimizations in their game projects **Week 4: Deployment, Platforms, and Reflection** Objectives 4 and 5: Prepare games for deployment on various platforms and reflect on the game development process Topics: Building games for different platforms (WebGL, mobile, desktop) Platform-specific considerations and optimizations Submission processes for various app stores Reflective practice in game development Activities: Build their games for at least two different platforms Write a postmortem analysis of their game development process Peer review of game projects Performance Assessment:  Students will work individually or in small teams to: 1. Conduct a final round of beta testing on their games
2. Analyze the collected data and feedback
3. Implement final optimizations and adjustments
4. Prepare their game for deployment on at least two platforms
5. Write a comprehensive postmortem report
6. Present their game, beta test results, and reflections to the class
 |
| **Leadership Alignment**: 10.B.1 While beta testing their products and during the development of their beta test, students will demonstrate additional attributes associated with producing high quality products including the abilities to:  10.B.1.a Work positively and ethically  10.B.1.b Manage time and projects effectively  10.B.1.c Multi-task  10.B.1.d Participate actively, as well as be reliable and punctual  10.B.1.e Present oneself professionally and with proper etiquette  10.B.1.f Collaborate and cooperate effectively with teams  10.B.1.g Respect and appreciate team diversity   |
| **Industry Standards and/or Competencies**: ISTE Standards [ISTE | 1. Students](https://iste.org/standards/students) 1.7. Global Collaborator:* 1.7.a. use digital tools to connect with learners from a variety of backgrounds and cultures, engaging with them in ways that broaden mutual understanding and learning.
 |
| **Aligned Washington State Academic Standards** |
| **Computer Science** | 3A-AP-21: Evaluate and refine computational artifacts to make them more usable and accessible. 3A-DA-11: Create interactive data visualizations using software tools to help others better understand real-world phenomena. 3A-DA-12: Create computational models that represent the relationships among different elements of data collected from a phenomenon or process. 3B-DA-05: Use data analysis tools and techniques to identify patterns in data representing complex systems. 3B-AP-22: Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). 3B-AP-23: Evaluate key qualities of a program through a process such as a code review. If this unit is expanded to include bigger picture, career-related stuff: 3A-IC-26: Demonstrate ways a given algorithm applies to problems across disciplines. 3A-IC-27: Use tools and methods for collaboration on a project to increase connectivity of people in different cultures and career fields.   |
| **Educational Technology** | Empowered Learner 1.c: Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways. 1.d: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. Knowledge Constructor 3.d: Build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions. Innovative Designer 4.a: Know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems. 4.b: Select and use digital tools to plan and manage a design process that considers design constraints and calculated risks. 4.c: Develop, test and refine prototypes as part of a cyclical design process. 4.d: Exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems. Computational Thinker 5.a: Formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions. 5.b: Collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making. 5.c: Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. Creative Communicator 6.b: Create original works or responsibly repurpose or remix digital resources into new creations. 6.c: Communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. Global Collaborator 7.b: Use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints. 7.c: Contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal. |
| **Mathematics: Common Core** | Interpreting Categorical and Quantitative Data HSS.ID.A: Summarize, represent, and interpret data on a single count or measurement variable. HSS.ID.B: Summarize, represent, and interpret data on two categorical and quantitative variables. HSS.ID.C: Interpret linear models. Making Inferences and Justifying Conclusions HSS.IC.B: Make inferences and justify conclusions from sample surveys, experiments, and observational studies. Creating Equations HSA.CED.A: Create equations that describe numbers or relationships. Interpreting Functions HSF.IF.B: Interpret functions that arise in applications in terms of the context. HSF.IF.C: Analyze functions using different representations. Modeling with Geometry HSG.MG.A: Apply geometric concepts in modeling situations. Quantities HSN.Q.A: Reason quantitatively and use units to solve problems.  |