

VIDEO GAME PROGRAMMING

Full-Year High School Course Syllabus | 2026–2027 School Year

New Market Skills Center | Instructor: Thomas Foster

Email: thomas.foster@tumwater.k12.wa.us

Phone: (360) 570-4447

Office Hours: 7:00–7:30 a.m. and 2:00–2:30 p.m.

Course Description

Video Game Programming is a project-based course that combines computer science, mathematics, visual art, and game design. Students begin with foundational C# programming in Visual Studio, computer systems, number systems, and art principles. They apply these skills in a console game before transitioning to the Unity game engine for interactive 2D game development. Second-year students may undertake approved 3D game projects. The course emphasizes computational thinking, creative problem-solving, debugging, teamwork, communication, and the complete game-development lifecycle.

Learning Outcomes

- Develop, test, and debug C# programs using variables, control flow, arrays, methods, and object-oriented design.
- Apply number systems, arithmetic, trigonometry, and vectors to interactive game behavior.
- Use visual-design fundamentals and animation principles to improve game assets and presentation.
- Build Unity games using sprites, input, collisions, 2D physics, scoring, and reusable code.
- Plan, prototype, test, revise, document, and present games individually and in teams.
- Demonstrate responsible file management, collaboration, professional communication, and meeting project milestones.

Year-at-a-Glance

Quarter	Instructional Emphasis	Major Deliverable
1	C# fundamentals; computer science; number systems; art foundations	BazeMaze C# console game
2	Unity fundamentals; 2D physics; hyper-casual design; beginning OOP	Original hyper-casual team game
3	Trigonometry; vectors; advanced C#; applied game mechanics	Math-focused Unity exercises and platformer
4	Integrated design, development, playtesting, and refinement	Final collaborative game project

Pacing is approximate. Topics and project work may overlap across quarters as instructional needs and team milestones require.

Quarter 1 — Foundations of Programming and Design

Programming: Hello World, comments, variables, C# type system, arithmetic, console input/output, if statements, switch statements, loops, arrays, methods, and introductory memory management. Students learn the anatomy of a C# console program and basic debugging.

Mathematics and computing: binary, decimal, and hexadecimal number systems; conversions; arithmetic; operator precedence; and computer hardware fundamentals.

Art and design: elements of art, studio logo design, introductory composition, and the Rule of Thirds.

Culminating assessment: BazeMaze, a C# console game demonstrating initialization, a main game loop, input, movement, collision, game-state updates, ASCII rendering, and orderly program termination.

Quarter 2 — Unity and Hyper-Casual Game Development

Students transition from console applications to the Unity Editor and 2D games. Instruction covers GameObjects, components, sprites, Rigidbody2D, colliders, physics materials, discrete and continuous collision detection, joints, and effectors. C# instruction expands into object-oriented concepts, enumerations, classes, information hiding, and properties.

Guided projects: SRCP (sprites, rigidbodies, colliders, physics materials); Cage (collision modes); Goldberg (2D joints and effectors); Click, Shoot, and Jump/Dodge hyper-casual games. A Flappy Bird-style exercise may be included as time permits.

Culminating assessment: Teams of two customize and develop an original hyper-casual game featuring a simple, understandable core gameplay loop.

Quarter 3 — Mathematics for Game Mechanics

Trigonometry: unit circle; sine, cosine, tangent; angles in degrees and radians; rotation and orbital movement. Guided projects include Orbit, Top-Down Vehicle, and Unit Circle.

Vectors: magnitude, normalization, angle, addition, subtraction, direction, and dot products. Guided projects include Vector, Vector Head to Tail, Enemy Follow, Vector Dot, Vector Dot Line, and Vector Dot Crosshair.

Advanced C#: static members, null references, object-oriented design, inheritance, and polymorphism. Students integrate these ideas in a 2D platformer and continue preparing collaborative game work.

Quarter 4 — Integrated Game Production

Students complete a final team game project with teams of two to six. Development includes concept, pitch, prototype, preproduction, production, Alpha, Beta, Gold, and final presentation. Students practice scope management, division of responsibilities, integration, debugging, playtesting, revision, documentation, and peer feedback. Second-year students may pursue approved 3D game designs.

Art and Design Across the Year

Visual instruction includes line, shape, form, space, texture, value, and color; the Rule of Thirds; color theory; principles of design; and principles of animation. Students apply these concepts to visual assets, user interfaces, composition, and game feedback.

Student Expectations and Employability

- Participate actively in demonstrations, exercises, critiques, and team meetings.
- Maintain organized project folders and clearly named files; back up work as directed.
- Submit original work, acknowledge outside resources, and follow school academic-integrity policies.
- Test and debug before submission, use feedback constructively, and document meaningful changes.
- Communicate respectfully, contribute equitably to team responsibilities, and meet agreed milestones.

Assessment and Communication

Student learning is evaluated through programming exercises, mathematics and art activities, demonstrations of understanding, working prototypes, game projects, and presentations. Assessment considers both the quality of completed work and evidence of the development process, including iteration and problem-solving. Specific grade

weighting, deadlines, late-work procedures, and make-up policies follow the instructor's published classroom requirements and applicable school policies.

Families and students should consult the school's designated learning platform for current assignments, feedback, due dates, and grades. Course sequence and optional activities may be adjusted for instructional pacing and available resources.

Appendix — Representative Quarter 1 Assignments

The following list records representative exercises and projects. Titles include activities adapted from or inspired by The C# Player's Guide, alongside course-specific work.

Number Sheet; Board Game; Hello World; ASCII Art; ReadLine; Triangle Farmer; Studio Logo; ASCII Name; Dominion Kings; Art Fundamentals; DataTypes; Four Sisters and Duckbear; The Defense of Consolas; PC Components; Repairing the Clock; Windows Files & Folders; Watchtower; Asterisk Patterns; Anatomy of a C# Console Program; Rule of Thirds; Buying Inventory; Replicator of D'To; Discounted Inventory; Laws of Freach; The Prototype; Magic Cannon; employability activities.