PLANNED INSTRUCTION

A PLANNED COURSE FOR:

Video Game Design

Curriculum writing committee: Audrey Dennis

Grade Level: 9-12

Date of Board Approval: _____2020_____

Course Weighting: Video Game Design

Tests	**30%**
Projects	**30%**
Quizzes/Homework	**25%**
Participation/Classwork/Soft Skills	**15%**
Total	**100%**

Curriculum Map

Overview:

Time/Credit for the Course: 1 Semester, ½ Credit

Course Description:
Video Game Design is a fun and interesting course that encompasses a student's curiosity or knowledge of programming and game design. Games are created using the object-oriented game design engine of ClickTeam, Unity, or other software program. The course includes game-theory reading with game building application lessons by integrating cross-curriculum and STEM activities. Topics include but are not limited to scene construction, interactive design, digital art, parent child objects, launching, graphing game coordinates, spawning, software ratings, game critique exposition, global variables, two-dimensional game art, gravity, C#, and the binary number system.

Goals:

Marking Period 1 Understanding of:

Game Design Vocabulary
Software Basics
Click Ball
Quality Assurance
Scene Construction
Iterative Design and the Scientific Method
Digital Art
Parent and Child Objects
Graphing Game Coordinates
Spawning
Player Movement
Basic Game Play
Animation, Sound, & Effects

Marking Period 2 Understanding of:

Gameplay Mechanics
User Interface
Software Ratings
Game Critique Exposition
Proof of Concept: Launching
Beta Build: Launching
Quality Assurance
Global Variables
Sensory Detection and Navigation
Challenges Using Variables
Two-Dimensional Game Art
Gravity and Ballistics
Game Mod: Particle Physics
Quality Assurance
Binary Number System

Big Ideas:

Big Idea # 1:

Computer and Information Technologies:

Computer technology is a data management and communication tool essential for business and personal productivity, problem solving, and decision making in the global world.

Textbook and Supplemental Resources:

Video Game Design and Programming Concepts 1st Edition. D. Michael Ploor. Goodheart-Wilcox Publisher, 2020.

Unity Learn. https://learn.unity.com/ February 13, 2020.

Computing & Game Design with ClickTeam Fusion. Daniel Block. First Printing, 2017.

Curriculum Plan

45 Days Per Unit

Unit 1

Standards:

PA Business, Computer Information Technology Academic Standards

15.4.12.A: 15.4.12.B: 15.4.12.C: 15.4.12.D: 15.4.12.E: 15.4.12.F: 15.4.12.G: 15.4.12.H: 15.4.12.I: 15.4.12.J: 15.4.12.K: 15.4.12.L: 15.4.12.M: 15.3.12.B: 15.3.12.C: 15.3.12.D: 15.3.12.F: 15.3.12.G: 15.3.12.H: 15.3.12.I: 15.3.12.J: 15.3.12.N: 15.3.12.P: 15.3.12.R: 15.3.12.S: 15.3.12.T: 15.3.12.W: 15.3.12.X: 3.4.12.A1, 3.4.12.A2, 3.4.12.A3

Common Core Standards

CC.2.3.HS.A.8, CC.2.3.HS.A.9, CC.2.3.HS.A.13, CC.2.3.HS.A.3, CC.2.3.HS.A.13 CC.2.3.HS.A.11, CC.2.3.HS.A.13, CC.3.6.9-10.A, CC.3.6.9-10.B, CC.3.5.9-10.A, CC.3.5.9-10.B, CC.3.5.9-10.C, CC.3.5.9-10.D, CC.3.5.9-10.E, CC.3.5.9-10.F, CC.3.5.9-10.G

Anchors:

R11.B.3, G.1.1.1, G.1.2.1, G.2.1.2, G.2.3.2

Eligible Content:

Game Design Vocabulary, Software Basics, Click Ball, Quality Assurance, Scene Construction, Iterative Design and the Scientific Method, Digital Art, Parent and Child Objects, Graphing Game Coordinates, Start Your 3D Engines, Pedal to the Metal, High Speed Chase, Step into the Driver's Seat, Player Positioning, Food Flight Collision Decisions, Jump Force, Make the World Whiz By, Particles and Sound Effects

Objectives:

Unit 1:

Lessons 1-9 Click Team

- Define common game design vocabulary. DOK 1
- List computer languages used in game design. DOK 1
- Compare and contrast fusion 2.5 with other common software applications. DOK 2
- Describe the user interface of the ClickTeam software. DOK 3
- Identify command, buttons, toolbars, panels, tabs, and other user interface structures. DOK 1
- Activate help text to identify command buttons and hot key commands. DOK 4
- Use game design tools to create a simple game. DOK 1
- Set background and add objects to enhance a scene. DOK 4
- Program objects to react to the game rules. DOK 4
- Evaluate the quality of your own work and the work of others. DOK 3

- Assess positive aspects of the playability and functionality of a game. DOK 3
- Provide constructive criticism to peers by suggesting possible solutions to problems. DOK 3
- Insert active and background objects to a game frame. DOK 4
- Program object movement to simulate natural movement. DOK 4
- Debug errors. DOK 4
- Refine interactions. DOK 4
- Perform authentic research on a selected topic. DOK 3
- Test a game or simulation to meet performance requirements. DOK 4
- Predict outcomes and evaluate deviations from expected results. DOK 4
- Perform research and evaluate source information. DOK 3
- Analyze data using the scientific method. DOK 4
- Create custom pixel art graphics for icons and logos. DOK 4
- Differentiate between raster and vector images. DOK 3
- Apply hexadecimal and rgb color models to graphic design. DOK 4
- Calculate pixel dimensions. DOK 1
- Differentiate relative and absolute referencing for plotting points on a game frame. DOK 3
- Position game objects in specified locations within a coordinate system. DOK 4
- Explain the parent-child object relationship. DOK 3
- Use ordered pair notation. DOK 1
- Plot two-dimensional points within an x and y axis system. DOK 4
- Explain path movement and nodes. DOK 3
- Use ordered pair notation. DOK 1
- Apply algebraic thinking skills to solve for relative locations. DOK 4
- Create a path. DOK 4

Lessons 1-3 Unity

- Use Unity Hub to install and manage versions of Unity on your computer. DOK 1
- Create a new Unity ID to be able to access of all Unity's services. DOK 4
- Create a new project through Unity Hub. DOK 4
- Navigate 3D space and the Unity Editor comfortably. DOK 2
- Add and manipulate objects in the scene to position them where you want. DOK 3
- Position a camera in an ideal spot for your game. DOK 4
- Control the layout of Unity Editor to suit your needs. DOK 3
- Create C# scripts and apply them to objects. DOK 4
- Use Visual Studio and a few of its basic features. DOK 1
- Write comments to make your code more readable. DOK 4
- Utilize fundamental C# methods and classes like transform.Translate and Vector3. DOK 4
- Add Rigidbody and Collider components to allow objects to collide realistically. DOK 4
- Duplicate objects in the hierarchy to populate your scene. DOK 4

- Declare variables properly and understand that variables can be different data types (float, Vector3, GameObject). DOK 3
- Initialize/assign variables through code or through the inspector to set them with appropriate values. DOK 3
- Use appropriate access modifiers (public/private) for your variables in order to make them easier to change in the inspector. DOK 1
- Analyze and gain user input with Input.GetAxis, allowing the player to move in different ways. DOK 4
- Use the Rotate function to rotate an object around an axis. DOK 1
- Clean and organize your hierarchy with Empty objects. DOK 3
- Adjust the scale of an object proportionally in order to get it to the size you want. DOK 2
- More comfortably use the GetInput function in order to use user input to control an object. DOK 3
- Create an if-then statement in order to implement basic logic in your project, including the use of greater than (>) and less than (<). DOK 4
- Transform a game object into a prefab that can be used as a template. DOK 4
- Instantiate Prefabs to spawn them into the scene. DOK 4
- Override Prefabs to update and save their characteristics. DOK 4
- Get user input with GetKey and KeyCode to test for specific keyboard presses. DOK 3
- Apply components to multiple objects at once to work as efficiently as possible. DOK 4
- Create an empty object with a script attached. DOK 4
- Use arrays to create an accessible list of objects or values. DOK 1
- Use integer variables to determine an array index. DOK 1
- Randomly generate values with Random.Range in order to randomize objects in arrays and spawn positions. DOK 4
- Change the camera's perspective to better suit your game. DOK 2
- Repeat functions on a timer with InvokeRepeating. DOK 4
- Write custom functions to make your code more readable. DOK 4
- Edit Box Colliders to fit your objects properly. DOK 2
- Detect collisions and destroy objects that collide with each other. DOK 4
- Display messages in the console with Debug Log. DOK 3
- Use GetComponent to manipulate the components of GameObjects. DOK 1
- Influence physics of game objects with ForceMode.Impulse. DOK 4
- Tweak the gravity of your project with Physics.gravity. DOK 3
- Utilize new operators and variables like &&. DOK 4
- Use Bool variables to control the number of times something can be done. DOK 4
- Constrain the RigidBody component to halt movement on certain axes. DOK 4
- Use tags to label game objects and call them in the code. DOK 4
- Use script communication to access the methods and variables of other scripts. DOK 4
- Manage basic animation states in the Animator Controller. DOK 3

- Adjust the speed of animations to suit the character or the game. DOK 2
- Set a default animation and trigger others with anim.SetTrigger. DOK 4
- Set a permanent state for “Game Over” with anim.SetBool. DOK 4
- Attach particle effects as children to game objects. DOK 4
- Stop and play particle effects to correspond with character animation states. DOK 4
- Work with Audio Sources and Listeners to play background music. DOK 3
- Add sound effects to add polish to your project. DOK 4

Core Activities and Corresponding Instructional Methods:

- Identify and define terminology related to game design.

- Software: Unity, Unity Hub, ClickTeam.

- Students will perform student activities for lessons 1-9 at www.m.g-wlearning.com which include chapter activity files.

- Students will complete the following in the Video Game Design and Programming Concepts textbook; review questions, higher order thinking strategies, word hunt, crossword puzzle, integration activity, game build, working in teams, and G-W Learning mobile website problems for each chapter.

- Handouts
  - Concept Map
  - KWL Chart
  - Notetaking
  - Venn Diagram

- Rubrics
  - Group Participation
  - Individual Participation
  - Individual Reports
  - Soft Skills

- Students will view videos and complete game builds/projects with Unity Lessons 1-3 https://learn.unity.com/

  Example Lesson 1 – Start your 3D engines: https://learn.unity.com/course/create-with-code-teacher-training/?tab=materials

  Step 1: Make a course folder and new project, Step 2: Import assets and open Prototype 1, Step 3: Add your vehicle to the scene Step 4: Add an obstacle and reposition it Step 5: Locate your camera and run the game Step 6: Move the camera behind the vehicle Step 7: Customize the interface layout

- https://learn.unity.com/course/create-with-code-live

  Seven weeks of tutorials and online video instruction from professionally certified Unity instructors.

Instructional Methods: Direct instruction, Demonstration, Discussion, Questioning, Cooperative learning

www.m.g-wlearning.com
https://www.clickteam.com/clickteam-fusion-2-5
https://learn.unity.com/
http://www.stemchallenge.org/stem/#/home
https://www.yoyogames.com/
https://www.piskelapp.com/
https://github.com/
Teacher-prepared handouts
Guest speakers
http://gamestarmechanic.com/

Assessments:

Diagnostic:

Discussion of student's prior knowledge (pretest)
Terminology preview
Oral responses during class discussion
Responses to questions from the beginning of the chapter/lesson and projects
Responses from videos

Formative:

Checkpoint exercises throughout the chapter/project
Textbook review questions for each lesson
Vocabulary quiz for each lesson
Successful completion of homework/class work assignments:
Crosswords and Integration Activity

Summative:

Graded audit checks on homework
Objective tests covering theory and terminology
Graded video game build projects/challenges/labs

Unit 2

Standards: – Business, Computer and Information Technology

PA Business, Computer Information Technology Academic Standards

15.4.12.A: 15.4.12.B: 15.4.12.C: 15.4.12.D: 15.4.12.E: 15.4.12.F: 15.4.12.G: 15.4.12.H: 15.4.12.I: 15.4.12.J: 15.4.12.K: 15.4.12.L: 15.4.12.M: 15.3.12.B: 15.3.12.C: 15.3.12.D: 15.3.12.F: 15.3.12.G: 15.3.12.H: 15.3.12.I: 15.3.12.J: 15.3.12.N: 15.3.12.P: 15.3.12.R: 15.3.12.S: 15.3.12.T: 15.3.12.W: 15.3.12.X: 3.4.12.A1, 3.4.12.A2, 3.4.12.A3

Common Core Standards

CC.2.3.HS.A.8, CC.2.3.HS.A.9, CC.2.3.HS.A.13, CC.2.3.HS.A.3, CC.2.3.HS.A.13
CC.2.3.HS.A.11, CC.2.3.HS.A.13, CC.3.6.9-10.A, CC.3.6.9-10.B, CC.3.5.9-10.A, CC.3.5.9-10.B, CC.3.5.9-10.C, CC.3.5.9-10.D, CC.3.5.9-10.E, CC.3.5.9-10.F, CC.3.5.9-10.G

Anchors:

R11.B.3, G.1.1.1, G.1.2.1, G.2.1.2, G.2.3.2

Eligible Content:

Spawning, Software Ratings, Game Critique Exposition, Proof of Concept: Launching, Beta Build: Launching, Global Variables, Sensory Detection and Navigation, Challenges Using Variables, Two-Dimensional Game Art, Gravity and Ballistics, Game Mod: Particle Physics, Binary Number System, Watch Where You're Going, Follow the Player, Power Up and Count Down, For Loops for Waves, Clicky Mouse, Keeping Score, Game Over, What's the Difficulty, Project Optimization, Research and Troubleshooting, Sharing your Projects

Unit 2:

Lessons 10-21, 23 Click Team

- Explain how object movement synchronization is achieved in a game. DOK 3
- Describe animated sprites and moving targets. DOK 3
- Create custom sprite art and animation direction nodes. DOK 4
- Program object spawning. DOK 4
- Evaluate the quality of their own work and the work of others. DOK 4
- Assess positive aspects of the playability and functionality of a game. DOK 3
- Provide constructive criticism to peers by suggesting possible solutions to problems. DOK 4
- Evaluate a project to introduce a precise claim. DOK 4
- Cite specific examples to support a claim and refute a counterclaim. DOK 3
- Integrate vocabulary into formal writing. DOK 3
- Construct claim, evidence, and commentary in a single paragraph format. DOK 3
- Explain the iterative process of game design. DOK 3
- Compare and contrast input and output devices of a user interface. DOK 3
- Create frame-by-frame animation using automated settings. DOK 4
- Program acceleration and deceleration. DOK 4
- Simulate conservation of matter in a video game. DOK 2
- Describe how momentum influences object motion. DOK 3
- Program emitters to create random objects. DOK 4
- Develop bonuses and perks to enhance gameplay. DOK 4
- Evaluate the quality of your own work and the work of others. DOK 3
- Assess positive aspects of the playability and functionality of a game. DOK 3
- Provide constructive criticism to peers by suggesting possible solutions to problems. DOK 4

- Compare and contrast global and local variables. DOK 3
- Evaluate multilevel gameplay navigation. DOK 3
- Analyze navigation for seamless gameplay. DOK 4
- Program custom movement for tile-based gameplay. DOK 4
- Design a sensory-detection system. DOK 4
- Implement collision detection within a sensory-detection system. DOK 4
- Perform maintenance on programming to include proper commenting and elegant composition. DOK 3
- Create modular programming components and structures. DOK 4
- Implement elegant programming solutions. DOK 3
- Evaluate existing programming for errors or refinement possibilities. DOK 3
- Build a playable video game. DOK 4
- Evaluate images using the elements of art and the principles of design. DOK 3
- Explain how visual perspective, lighting, and shadows help create the illusion of 3d space. DOK 3
- Create custom colors using the rgb color model. DOK 4
- Compose complex shapes from primitive shapes. DOK 3
- Generate custom gradients to create lighting, shadow, and illusion of depth. DOK 4
- Apply textures and pictures to object surfaces. DOK 4
- Simulate 3d objects using 2d shapes. DOK 4
- Rotate objects in 3d space around the x, y, and z axes. DOK 2
- Determine gravity and balancing forces. DOK 3
- Explain how vectors determine the direction of speed and forces. DOK 3
- Calculate acceleration, velocity, and distance. DOK 1
- Program using a physics engine. DOK 4
- Create realistic ballistic trajectories within a video game. DOK 4
- Explain particles and particle physics use in a game. DOK 3
- Set up particle physics to simulate an explosion. DOK 4
- Create particle generation from destroyed objects. DOK 4
- Compare and contrast the binary and base-10 number systems. DOK 3
- Explain programming subroutines. DOK 3
- Construct an app to convert binary numbers into decimal numbers. DOK 3
- Identify the purpose of programming modes. DOK 1

Lessons 4-7 Unity

- Apply Texture wraps to objects. DOK 4
- Attaching a camera to its focal point using parent-child relationships. DOK 3
- Transform objects based on local XYZ values. DOK 3
- Apply Physics Materials to make game objects bouncy. DOK 4
- Normalize vectors to point the enemy in the direction of the player. DOK 3

- Randomly spawn with Random.Range on two axes. DOK 3
- Write more advanced custom functions and variables to make your code clean and professional. DOK 4
- Write informative debug messages with Concatenation and variables. DOK 3
- Repeat functions with the power of ienumerator and Coroutines. DOK 3
- Use setactive to make game objects appear and disappear from the scene. DOK 4
- Use Repeat functions with For-loops. DOK 4
- Increment integer values in a loop with the ++ operator. DOK 4
- Target objects in a scene with findobjectsoftype. DOK 3
- Return the length of an array as an integer with. Length. DOK 3
- Program basic movement. DOK 4
- Program object collisions. DOK 4
- Spawn object prefabs on timed intervals. DOK 3
- Switch the game to 2D view for a different perspective. DOK 3
- Add torque to the force of an object. DOK 3
- Create a Game Manager object that controls game states as well as spawning. DOK 4
- Create a List of objects and return their length with Count. DOK 4
- Use While Loops to repeat code while something is true. DOK 4
- Use onmousedown to enable the player to click on things. DOK 4
- Create UI Elements in the Canvas. DOK 4
- Lock elements and objects into place with Anchors. DOK 3
- Use variables and script communication to update elements in the UI. DOK 4
- Make UI elements appear and disappear with .setactive. DOK 3
- Use Script Communication and Game states to have a working “Game Over” screen. DOK 4
- Restart the game using a UI button and Scene Management. DOK 3
- Store UI elements in a parent object to create Menus, UI, or HUD. DOK 3
- Add listeners to detect when a UI button has been clicked. DOK 3
- Set difficulty by passing parameters into game functions like spawnrate. DOK 3
- Recognize and use new variable attributes to keep values private, but still editable in the inspector. DOK 3
- Use the appropriate Unity Event Functions (e.g. Update vs. Fixedupdate vs. Lateupdate) to make your project run as smoothly as possible. DOK 4
- Understand the concept of Object Pooling, and appreciate when it can be used to optimize your project. DOK 2
- Use Unity Forums, Unity Answers, and the online Unity Scripting Documentation to implement new features and troubleshoot issues with your projects. DOK 4
- Add and manage export modules for your Unity installs so you can choose which platforms to build for. DOK 3
- Build your projects for Mac or PC so they can be played as standalone apps. DOK 3

- Build your projects for webgl so they can be uploaded and embedded online and shared with a single URL. DOK 3

Core Activities and Corresponding Instructional Methods:

- Identify and define terminology related to game design.
- Software: Unity, Unity Hub, ClickTeam.
- Students will perform student activities for lessons 10-21, 23 at www.m.g-wlearning.com which include chapter activity files.
- Students will complete the following in the Video Game Design and Programming Concepts textbook; review questions, higher order thinking strategies, word hunt, crossword puzzle, integration activity, game build, working in teams, and G-W Learning mobile website problems for each chapter.
- Handouts
  - Concept Map
  - KWL Chart
  - Notetaking
  - Venn Diagram
- Rubrics
  - Group Participation
  - Individual Participation
  - Individual Reports
  - Soft Skills
- Students will view videos and complete game builds/projects with Unity Lessons 4-7 https://learn.unity.com/

  Example Lesson 4 – Watch Where You're Going:
  https://learn.unity.com/course/create-with-code-teacher-training/?tab=materials
  Step 1: Create project and open scene, Step 2: Set up the player and add a texture, Step 3: Create a focal point for the camera, Step 4: Rotate the focal point by user input, Step 5: Add forward force to the player, Step 6: Move in direction of focal point
- https://learn.unity.com/course/create-with-code-live
  Seven weeks of tutorials and online video instruction from professionally certified Unity instructors.

Instructional Methods: Direct instruction, Demonstration, Discussion, Questioning, Cooperative learning

www.m.g-wlearning.com

https://www.clickteam.com/clickteam-fusion-2-5
https://learn.unity.com/
http://www.stemchallenge.org/stem/#/home
https://www.yoyogames.com/
https://www.piskelapp.com/
https://github.com/
Teacher-prepared handouts
Guest speakers
http://gamestarmechanic.com/

Assessments:

Diagnostic:

Discussion of student's prior knowledge (pretest)
Terminology preview
Oral responses during class discussion
Responses to questions from the beginning of the chapter
Responses from articles or videos

Formative:

Checkpoint exercises throughout the chapter/project
Textbook review questions for each lesson
Vocabulary quiz for each lesson
Successful completion of homework/class work assignments:
Crosswords and Integration Activity

Summative:

Graded audit checks on homework
Objective tests covering theory and terminology
Graded video game build projects/challenges/labs

Checklist to Complete and Submit:

(Scan and email)

__x___ Copy of the curriculum using the template entitled "Planned Instruction," available on the district website.

___x__ The primary textbook form(s).

__x___ The appropriate payment form, in compliance with the maximum curriculum writing hours noted on the first page of this document.

Each principal and/or department chair has a schedule of First and Second Readers/Reviewers. Each Reader/Reviewer must sign & date below.

First Reader/Reviewer Printed Name - Maura Angle

First Reader/Reviewer Signature - *Maura Angle* **Date: April 15, 2020**

Second Reader/Reviewer Printed Name_____

Second Reader/Reviewer Signature _____ Date_____

Please Go to Human Resources page on the Delaware Valley School District website for updated Payment form to be submitted.

https://pa01001022.schoolwires.net/site/handlers/filedownload.ashx?moduleinstanceid=7055&dataid=16708&FileName=AUTHORIZATION%20FOR%20PAYMENT%20-%20SECURED.pdf