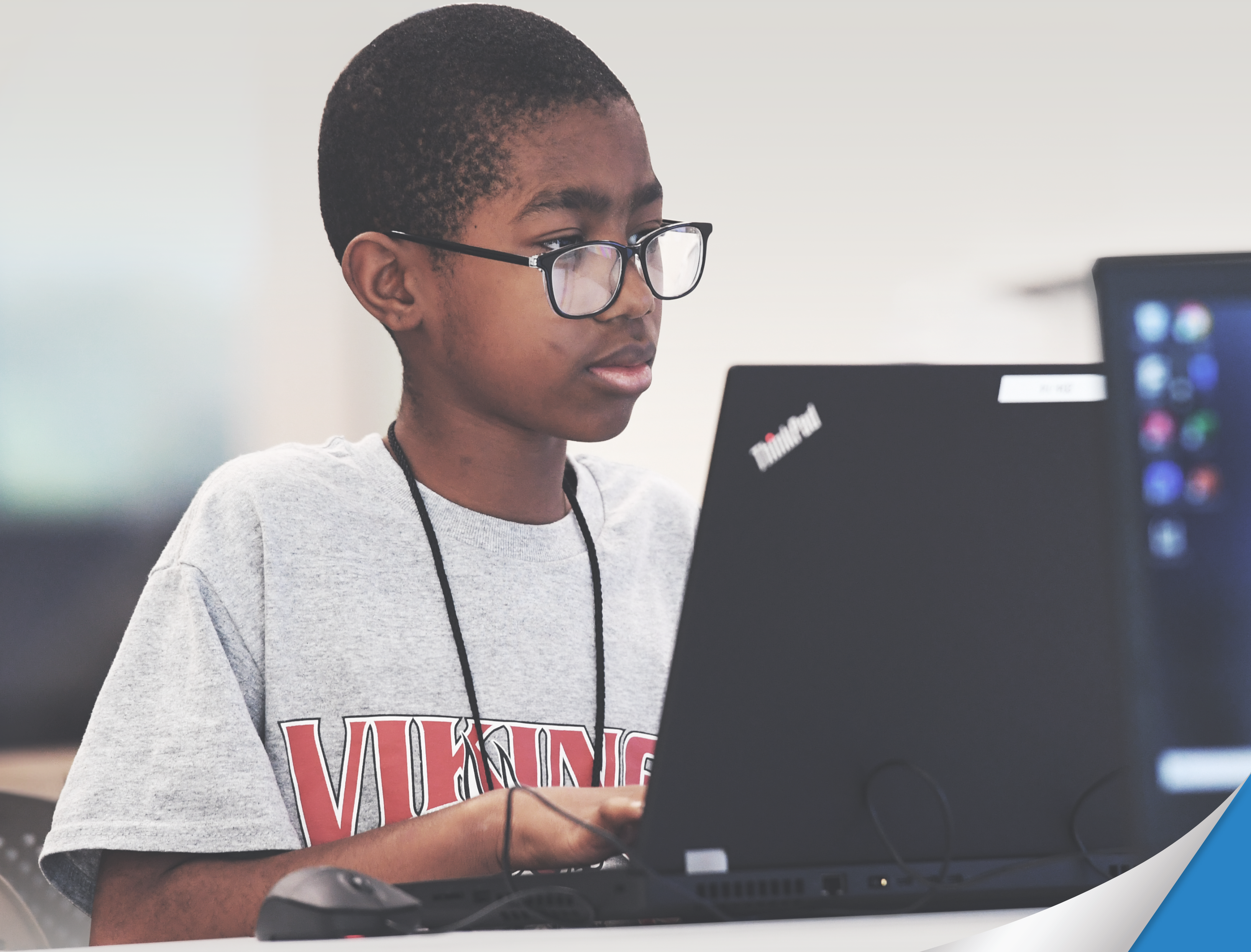
VIDEO GAME DESIGN



TGR FOUNDATION

MIDDLE SCHOOL

TGR FOUNDATION

VIDEO GAME DESIGN

LESSON OVERVIEW

The worlds of science, engineering, math, technology, and art come together in video game design. The gaming industry is vast and growing. It includes many careers, from marketing and public relations to music composition and animation. This 4-day lesson plan introduces students to the fundamental principles of game design through a series of fun and interactive sessions. Students will learn the history of video games and the value that these games bring to our society. They will then learn how logic and reason are used to create enriching and fun games through a host of exciting activities. The lesson culminates with a team-based project module, where students will have a hands-on opportunity to storyboard and practice building their own game.

These lessons are aligned to the following Common Core English Language Arts Standards for Science and Technical Subjects, Common Core Math Practices, National Educational Technology (ISTE), and Computer Science Teachers Association (CSTA) Standards:

Content Areas: Computer Science, Technology, Art, and Design

Activity Duration: 4 class sessions (45 minutes each)

Grade Level: Secondary

Essential Questions:

1. How have video games evolved over time? Why is gaming important in our society?
2. How are logic and reason used to create video games?
3. What careers exist in gaming? How do teams of professionals work together to build games?
4. What are the building blocks of video games? How is software used to build games?

Materials:

- Projector and screen
- Computer with internet capabilities (one per student, or one per student group)
- Access to the Internet
- Whiteboard or chalkboard
- Paper and pencil/pen
- Worksheets included in the Educator Guide—one of each per student
- Large post-it paper and markers (optional)

Objectives:

1. Identify the basic elements of video games and how games have evolved over time.
2. Understand the value of video games—what they add to society and how gaming forms a vital part of our economy.
3. Comprehend how algorithms use logical expressions like if-then statements and loops to build games.
4. Map out a game by creating concepts and implement those concepts using code.

BACKGROUND INFO

Computer gaming design is a large and lucrative field. According to the Entertainment Software Association (ESA), 63% of U.S. households have at least one member who spends three or more hours per week playing video games.3 Games and gaming systems continue to evolve, and as they do, we find new uses for video games. Evidence shows that games are a great way to learn, and new data suggests that video games could have a powerful impact on mental and physical rehabilitation.[4] As the video game design field continues to expand, it is essential that students learn marketable skills that will prepare them for roles in this thriving industry. Furthermore, it is urgent that educators work to provide students with a framework for understanding how video games impact their lives and our greater society.

Game design is rooted in computational thinking, a powerful problem-solving approach that utilizes logic to break a complex problem or idea into smaller, manageable components. Some of the most common patterns in game design are control statements such as if-then conditions (ex: if I click a button, then a character will jump) and loops (a series of code statements that repeat). Algorithms use those patterns of control statements to provide a clear set of instructions that create the framework of a program. These logic patterns appear in most every coding language and form the backbone of most gaming design.[5] By contextualizing these logical statements in the real world and then coding them in an engaging, hands-on digital tool, students can effectively grasp computational thinking and approach coding and game design with a sense of confidence.[5]

Scratch is an online resource that is maintained by the Media Lab at the Massachusetts Institute of Technology (MIT). By utilizing basic elements of code, users can program their own interactive games and stories. Scratch was specifically designed for use with students ages 8–16, and is well suited for use in the classroom.6

PROCEDURE

DAY 1

Engage (Slides 1–7)

Overview: Students will learn the history of video games and how they have evolved over time. They will also receive context for why games and gaming are important to our society. They will learn about careers in game creation and identify necessary skills for these careers. Students will finish the lesson by learning the basic components of video games and work together to identify those components in the games they know and enjoy.

Slides 3–6 prompt students to capture information in the "Video Games and Society" worksheet.

Slide 7 will prompt students to complete an activity using the "Deconstruct a Game" worksheet.

Note: computers with internet access will NOT be needed for Day 1.

Slide 1

- Invite students to reflect on their current ideas and perceptions of video game design.
  Ask: When you think of video games, what comes to mind?
- Clarify the difference between video games and computer games—designed for different formats (TV/console versus computer/smartphone, keyboard and mouse versus controllers).

Slide 2

- Review lesson objectives with students.
- Express that students will have the opportunity to use computers in days 2 and 3 of this unit.

Slide 3

- Explain that the history of video games begins with the earliest computers. EDSAC (Electronic delay storage automatic computer), built in 1949, was an early computer made in Britain. It was able to do just one task and followed a very limited set of instructions.
- Click to reveal each video game evolution on the timeline. Invite students to give a thumbs up if they have heard of each game as it is presented.
- Express how the early video games like Pong and Pac-Man were less sophisticated than current games. They used simpler graphics and had less complex storylines. These were early video games and were only accessible at video arcades. Arcade games were large machines that were dedicated to one game only.
- Computer games then evolved into the home with consoles like Atari and Super Nintendo. There are many classic games that debuted on these consoles, including Super Mario Bros. These games were eventually modified for portable consoles, like Game Boy.
- In the '90s and '00s, consoles became more sophisticated and could process games with more complex graphics and storylines. This resulted in games like Final Fantasy and Halo, which rely heavily on plot and character development to keep players engaged.
- Ask students to share with a partner about how they use games today. Are they gaming on an Xbox? A Playstation? Their smartphone? A computer?
- Ask students to take a few minutes to write on their worksheets and identify the ways that games have evolved over time (machines have become smaller and more efficient, graphics and audio are more detailed, plots are more complicated and the advent of the internet has made updating the game and playing globally more possible). Ask students how they think gaming will evolve in the future.

Slide 4

- Explain to students that games have practical and fun applications.[5] Computer games are increasingly used in educational settings and have been shown to improve learning.[6] Ask students to think of a way that video games have helped them learn.
- Lead students to the "Video Games and Society" worksheet, where they will record ways that video games have improved their lives. Encourage 2–3 students to share their responses.

Slide 5

- Review the careers in gaming with students and lead them to the "Careers in Gaming" activity on their "Video Games in Society" worksheet.
- Write the careers on the board. Explain each career. A software engineer programs video games. Graphic designers design the look of the game universe: characters, settings, etc. Marketing and public relations professionals sell and promote games. They work to build excitement around games and engage people in playing them. Sound engineers design the music, character voices and other sound components of video games.
- Once the students have completed the activity sheet, encourage students to come up to the board and write relevant skills under the careers listed. Skills include time management, computer programming, collaboration, and writing and editing.

Slide 6

- Click to reveal the elements of video games.
    - If needed, use an example of a game to walk through the components. Example, in Super Mario Bros.:
        - Genre: Adventure game
        - Characters: Mario, Luigi, Princess Peach, Bowser
        - Storytelling: Mario and Luigi are the protagonists. Bowser is the antagonist.
        - Objective: Mario and Luigi are journeying through their world and defeating enemies with the goal of saving Princess Peach from the antagonist Bowser.
        - User Interface: Users press A to begin levels and jump, the forward button to walk forward, the Start button to pause the game, etc.
        - Audio: The game has a soundtrack with different music for different environments (water, underground). Each time a character makes an action, there is a related sound effect.

Slide 7

Ask students to take out their "Deconstruct a Game" worksheet. Assign students into groups of three or four. Have each group select a game to deconstruct by identifying the basic elements. This can be done on paper or if you have large post-it paper and markers accessible, you can have each student group deconstruct their game on the large paper and hang them around the room. Then, 1–2 representatives from each group can explain the group's thought processes in deconstructing their game.

DAY 2

Explore/Explain (Slides 8–12)

Overview: In this section, students will learn about why programming logic is important to the creation of video games. They will explore two types of commonly used control statements: if-then conditions and loops. Students will engage in a group activity to further their learning and understanding of algorithms. The lesson will culminate with an instructorled tutorial on Scratch, where students will create their own Pong game by following an algorithm of step-by-step instructions.

Slides 9 and 10 will direct students to record answers on the "It's All Logic!" worksheet. Slide 11 leads students to the algorithm activity on the same worksheet. Slide 12 begins the instructor-led tutorial for building a Pong game.

Note: computers with internet access WILL be needed for Day 2. Before you begin this lesson, ensure that you have a computer that is hooked up to a projector and capable of accessing the internet.

Slide 8

- Begin by explaining that logic is an important part of game design, just as it is important in our daily lives. When something is "logical," it is rational and makes sense. For example, it is logical that if I put a key in the ignition of a car and turn it, the engine will go on. In a video game, if a button is pressed, it is logical that a character will perform an action. Ask students to think of other examples of logic in the games they play. Collect 2–3 student responses. Ask students to take out their "It's All Logic!" worksheet.

Slide 9

- Explain how if-then statements are logical cause-and-effect equations. List the examples on the slide and then call on a couple students to think of real-life examples of if-then type statements (if I press the home button on my phone, the app screen appears, if I microwave my food, it gets warm, etc.).
- Click to reveal the action statement at the bottom of the slide. Have students identify 3 examples of if then statements that they have seen in video games. Encourage 2-3 students to share their responses.

Slide 10

- Explain that loops are sequences of code that are repeated until certain conditions are reached in a game.
- Ask students to think of 2–3 real-life examples of loops (running, meter in music, etc.).
- Click to reveal the question at the bottom of the slide and lead students to the "Loops" section of their "It's All Logic!" worksheet. Have them write down why they think loops are important to game design (appropriate answers include creating a consistent game environment and saving time in computer programming) and encourage 2–3 students to share their responses.

Slide 11

- Explain that algorithms are a set of instructions that create a game environment. Provide examples of algorithms in real-life: recipes, directions, etc. Ask 2-3 students to call out examples of algorithms they have experienced.
- Provide the analogy of an algorithm for checkers:
  - If-then statements—if a red checker jumps diagonally over a black checker, then remove the black checker.
  - Loops—While at least one red checker and one black checker is on the board, continue taking turns.
  - Together, these if-then statements and loops create an algorithm for a game environment.
- Ask students to take out their "Draw a Cake Using Algorithms" worksheet:
  - Pass out a piece of blank paper to each student.

- Ask students to begin by taking 2–3 minutes to draw a cake in the allotted space on their sheet.
- Then, ask students to take 5–7 minutes to write down step-by-step instructions on how to draw their exact cake (in 10 steps or less) in the allotted space.
- Have students tear their instructions off of their worksheet and hand them to a partner.
- Allot 3–5 minutes for students to follow the instructions given to them by their partner in order to draw a cake.
- Have students hand their cake drawing and instructions back to their partner.
- Allow students 3–5 minutes to compare the drawings and answer the prompts on the activity sheet.
- Ask 2–3 partner groups to answer the following questions:
    - Did the cakes end up looking similar or different?
    - Were the directions on how to draw the cake clear? How would you improve them?
- Express that instructions must be specific and in correct order if they are to create the intended result. If an algorithm or set of instructions is not correct in a game, the game will be illogical or will malfunction.

Slide 12

- Lead students through a step-by-step game design tutorial in Scratch. Note: if you have a limited number of computers available, students can work in group and take turns operating the computer while following along with the tutorial. It is preferable that each student has access to a computer for this exercise.
    - Ensure that your computer and projector are on so that the class can observe.
    - Ensure that each student or student group's computer is on and capable of accessing the internet.
    - Instruct students to open their browser and go to scratch.mit.edu.
    - Have students click on the green button that says "Try it Out," as shown on the slide.
    - On the right-hand side of the screen, there will be a list of options. Click the blue bar that says "Create a Pong Game" and ask students to do the same.
    - Once everyone is in the "Create a Pong Game" tutorial, direct the students to follow along as you complete the Pong game tutorial step by step. Be sure to allow ample time for students to complete each step and check for comprehension.
    - Follow the tutorial as it guides you through the game build process.
    - Have students save their games for use in Days 3 and 4.

DAY 3

Elaborate (Slides 13–14)

Overview: In this section, students will expand on their knowledge of game elements and control-flow statements by creating a plan to remix the Pong game they built. Students will take the if-then statements they have designed and code them in Scratch. They will then loop the necessary statements and work to actively create their game concept.

Slides 13–17 will ask students to record information on the "Remix a Game" worksheet.

Note: computers with internet access WILL be needed for this lesson.

Slide 13

- Explain to students that this lesson will focus on customizing the Pong game they built. They will later be provided the opportunity to storyboard their own game.

Slide 14

- Review if-then statements by referring to how if-then statements were used in the creation of the Pong game in Scratch.
- Distribute the "Remix a Game" worksheet to students.
- Provide students time to "code" 5 if-then statements to remix their Pong games. They will not program these into Scratch in this lesson, but will instead write out the if-then statements on their worksheet.

Slide 15

- Review loops by referencing how loops were used in the construction of the Pong game in Scratch. Ask students to think about which if-then statements they will want to loop when they remix their game.

Slide 16

- Review algorithms by referencing how loops were used in the construction of the Pong game in Scratch.
- Direct students to the "Remix a Game" worksheet and ask them to assess whether or not their if-then statements have created a clear algorithm.

Slide 17

- Review the objectives and direct students to scratch.mit.edu in order to code their if-then statements.
- Allow time for students to remix their Pong games.
- If students get stuck while coding, encourage them to review the "Create a Pong Game" tutorial.

Evaluate

Overview: In this section, students will evaluate the effectiveness of their code and assess what changes they would make to their game. Students will reflect on the importance of the game elements and contextualize the skills they have learned with the skills needed in the industry of game design.

Slide 18 will ask students to record information on the "Reflection Questions" worksheet.

Slide 18

- Once students have coded and looped their 5 if-then statements, distribute the "Reflection Questions" worksheet.
- Have students work individually, with partners or in groups to answer the questions on the worksheet.
- Lead students through the questions and invite 2–4 students or student groups to share-out their answers.
- Invite students to share out any next steps they may have for their game and any careers that interested them that they want to explore further.

DAY 4

Extension (Slide 19)

Overview: In this section, students will storyboard a new game concept that evokes all of the game elements they learned.

Slide 19 will prompt students to complete an activity using the "Imagine a Game" worksheet.

Note: Computers with internet access are OPTIONAL for Day 4. Students may complete this lesson individually, with partners, or in groups. If students are placed with partners or groups, invite each member to assume a career role(s) from slide 6 while creating the game.

Slide 19

- Explain to students that this lesson will focus on conceptualizing their own game.
- Distribute the "Imagine a Game" worksheet to students.
- Review the six elements of video games with students, using a game example like Donkey Kong or Tetris to add context.
- Provide students time to name their game and identify the six components in their own game.
- Invite students to review and provide feedback on at least two other game proposals.

WORKS CITED

[1] Common Core State Standards Initiative. 2017. English Language Arts Standards, Writing, Grade 8. http://www.corestandards.org/ELA-Literacy/W/8/#CCSS.ELA-Literacy.W.8.3.

[2] Common Core State Standards Initiative. 2017. English Language Arts Standards, Science and Technical Subjects, Grades 9 & 10. http://www.corestandards.org/ELA-Literacy/RST/9-10/

[3] Education Software Association. 2016. Essential Facts About the Software Gaming Industry: Sales, Demographics and Usage Data. http://essentialfacts.theesa.com/Essential-Facts-2016.pdf

[4] Preto, Sara. 2013. "Video Games for Rehabilitation." USC Institute for Creative Technologies. http://ict.usc.edu/news/video-games-for-rehabilitation/.

[5] Ioannidou, Andri; Bennett, Vicki; Repenning, Alexander; Koh, Kyu Han; Basawapatna, Ashok. 2011. Computational Thinking Patterns. Paper presented at the 2011 Annual Meeting of the American Educational Research Association (AERA) in the Division C - Learning and Instruction / Section 7: Technology Research symposium "Merging Human Creativity and the Power of Technology: Computational Thinking in the K-12 Classroom." http://files.eric.ed.gov/fulltext/ED520742.pdf.

[6] Scratch. Lifelong Kindergarten Group at the MIT Media Lab. https://scratch.mit.edu/about.

[7] East Carolina University. 2008. "ECU study shows casual video games relieve stress." http://www.ecu.edu/cs-admin/news/newsstory.cfm?ID=1344.

[8] Institute of Play. "Why Games and Learning." http://www.instituteofplay.org/about/context/why-games-learning/.

COMMON CORE STATE STANDARDS

ENGLISH LANGUAGE ARTS STANDARDS: WRITING (GRADES 6, 7 & 8)[1]

Text Types and Purposes (CCSS.ELA-LITERACY.W.6.3, CCSS.ELA-LITERACY.W.7.3 & CCSS.ELA-LITERACY.W.8.3)

- Write narratives to develop real or imagined experiences or events using effective technique, relevant descriptive details, and well-structured event sequences.

MATH

The Standards for Mathematical Practice describe varieties of expertise that mathematics educators at all levels should seek to develop in their students

Practice MP1: Make sense of problems and persevere in solving them.

Practice MP5: Use appropriate tools strategically.

ISTE STANDARDS FOR STUDENTS

International Society for Technology in Education- The 2016 ISTE Standards for Students emphasize the skills and qualities that allow students to engage and thrive in a connected, digital world.

1. Empowered Learner: Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences.

    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.

    b. Students 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.

2. Innovative Designer: Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.
    a. Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
    b. Students select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
    c. Students develop, test and refine prototypes as part of a cyclical design process.
    d. Students exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
3. Computational Thinker: Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.
    a. Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
    d. Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
4. Creative Communicator: Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals.
    b. Students create original works or responsibly repurpose or remix digital resources into new creations.
    c. Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.
    d. Students publish or present content that customizes the message and medium for their intended audiences.
5. Global Communicator: Students use digital tools to broaden their perspectives and enrich their learning by collaborating with others and working effectively in teams locally and globally.
    b. Students use collaborative technologies to work with others, including peers, experts or community members, to examine issues and problems from multiple viewpoints.
    c. Students contribute constructively to project teams, assuming various roles and responsibilities to work effectively toward a common goal.

CSTA STANDARDS

Computer Science Teachers Association has developed a core set of learning standards designed to provide the foundation for a complete video science curriculum and its implementation at the K–12 level.

Level 2 (recommended for grades 6–9) Computer Science and Community: Middle school/junior high school students begin using computational thinking as a problem-solving tool. They begin to appreciate the ubiquity of computing and the ways in which computer science facilitates communication and collaboration. Students begin to experience computational thinking as a means of addressing issues relevant, not just to them, but to the world around them. The learning experiences created from these standards should be relevant to the students and should promote their perceptions of themselves as proactive and empowered problem solvers. They should be designed with a focus on active learning and exploration and can be taught within explicit computer science courses or embedded in other curricular areas such as social science, language arts, mathematics and science.

COMPUTATIONAL THINKING (CT)

1. Use the basic steps in algorithmic problem solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation).
2. Define an algorithm as a sequence of instructions can be processed by a computer.
3. Evaluate ways that different algorithms can be used to solve the same problem.

4. Act out searching and sorting algorithms.
5. Describe and analyze a sequence of instructions being followed (e.g., describe a character's behavior in a video game as driven by rules and algorithms).

COLLABORATION (CL)

1. Apply productivity/multimedia tools and peripherals to group collaboration and support learning throughout the curriculum.
2. Collaboratively design, develop, publish and present products (e.g., videos, podcasts, websites) using technology resources that demonstrate and communicate curriculum concepts.
3. Collaborate with peers, experts and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities.
4. Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.

COMPUTING PRACTICE AND PROGRAMMING (CPP)

1. Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems.
2. Demonstrate an understanding of algorithms and their practical applications.
3. Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables and functions.
4. Identify interdisciplinary careers that are enhanced by computer science.

VIDEO GAMES AND SOCIETY

PRINTOUT FOR DAY 1

You might know that video games are a fun way to relax, but did you know that they're also a great way to learn? Schools, hospitals and other organizations use video games to educate students, heal patients, and increase productivity.

The History of Computer Games
Computers and games have come a long way since the creation of EDSAC in 1949. From Pong to XBox and everything in between, gaming has developed better graphics, complex storylines, and new methods for players to connect. Can you describe how graphics, gaming consoles, storylines and video effects have changed over time?

____

____

____

____

Why Are Video Games Important?
Video games help us learn, heal, and have fun. They're all around us!

How have video games improved your life?

____

____

____

____

CAREERS IN GAMING

Because of the high demand for games, there are lots of careers for professionals looking to work in gaming.

Can you match the skills in the skills bank to the careers that you think would need those skills? You can use each skill more than once.

SKILLS	SKILLS BANK
Graphic Design	Time management
Computer Programming	Problem-solving skills
Audio/Music Composition	Creative thinking
Marketing/Public Relations	Design skills
	Logical thinking
	Musical skill
	Attention to detail
	Collaboration
	Flexibility
	Programming skills

DECONSTRUCT A GAME

ACTIVITY FOR DAY 1

There are many components to the games we enjoy playing—sound, visuals and storylines to name a few. Think about a game you enjoy playing and map out the components in the spaces below.

My Favorite Game is: ____

Genre
What type of game is this?

____
____
____

Storytelling
What is the plot of your game?

____
____
____

Who are the heroes and villains?

____
____
____

User Interface
How does a user interact with the game?

____
____
____

Characters
Name some of the characters or objects in your game.

____
____
____

Objective/Goals
What is the objective of your game?

____
____
____

What do you have to do in your game to “win” or “lose”?

____
____
____

Audio
Describe the music, voices and sound effects in your game.

____
____
____

IT'S ALL LOGIC!

PRINTOUT FOR DAY 2

Coding is the language that computers use to operate. Coding is built on algorithms that contain logical statements—it's like a recipe that tells you how to bake a cake step-by-step. Each step and ingredient is important to the final result. Let's explore how we can connect logical statements and examples from the world around us with the logical statements used to build video games.

If-Then Statements

If-then statements are logical steps that affect how a game is played. Here are some if-then statements you might come across as you're playing checkers:

- If the red checker jumps diagonally over black checker, then remove the black checker
- If red checker reaches square at the opposite edge of the board, then add a second red checker on top
- If there are no black checkers left on the checkerboard, then the red player wins

Can you come up with three examples of if-then statements you've seen in video games?

1. ______________________________

2. ______________________________

3. ______________________________

Loops

Loops are used to repeat statements and allow for efficient programming. These are some loops that might be used to design a virtual checkerboard:

- Draw a black square and a red square to the right of the black square. Repeat this 4 times.
- Go to the next row and start with a red square. Put a black square next to it and repeat 4 times.
- Go to the next row and repeat the two sets of instructions above three more times.

Why do you think loops are important in video game design?

____

____

____

Algorithms

When we play checkers, there are rules and constraints on what we can and cannot do in the game. These rules and constraints are called an algorithm. They form the basis of the game environment and help determine things like who wins and loses.

DRAW A CAKE USING ALGORITHMS

ACTIVITY FOR DAY 2

In this activity, we'll use algorithms to help our classmates draw a cake. First, you'll start by drawing a cake yourself in the space below. Then, in ten steps or less, create a clear set of instructions on how to draw the cake. Tear off the instructions and hand them to your partner (make sure you hold onto your drawing of your cake—you don't want your partner to see it!). Your partner will hand their instruction list to you. On your blank sheet of paper, follow their instructions as closely as you can. When you have finished, hand your drawing and your partner's instructions back to your partner. Show each other the cakes you have drawn, and answer the following questions:

1. Does your cake look like the cake your partner drew? Why or why not?
2. Did your partner provide you with clear and easy to follow instructions on how to draw the cake? What changes would you make to the instructions to reach the intended result?

Draw your cake here:

Create your step-by-step instructions here. Tear these off and hand them to your partner:

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.

REMIX A GAME

PRINTOUT FOR DAY 3

If-Then Statements

Write out at least 5 if-then statements for your Pong game that you will code in Scratch:

1.

2.

3.

4.

5.

Predict:

What will your if-then statements add to the Pong game?

Will you want to loop any of the above if-then statements in your game? If so, which ones?

REFLECTION QUESTIONS

PRINTOUT FOR DAY 3

1. What does each of your if-then statements add to your Pong game?

2. Are the algorithms in your game clear? If not, do your if-then and loop statements need to be fixed? How would you improve them?

3. Which career would you choose in video game design and why?

IMAGINE A GAME

PRINTOUT FOR DAY 4

Individual/group name(s):

____

Name of your game:

____

Game Elements

Identify the following elements of your game

Objective:____

Genre: ____

Characters:____

Story:____

Objective:_ ____

User Interface:____

Audio: ____