**Teacher Name : Computer Science Department Subject : Computer Science Start Date(s): 45 Days Grade Level (s): 8th Grade Building :**

|  |
| --- |
| **Unit Plan: - Problem Solving** |
| **Unit Title:**  **Unit 1 - Problem Solving**  The Problem Solving unit is a highly interactive and collaborative introduction to the field of computer science, as framed within the broader pursuit of solving problems. Through a series of puzzles, challenges, and real world scenarios, students are introduced to a problem solving process that they will return to repeatedly throughout the course. Students then learn how computers input, output, store, and process information to help humans solve problems. The unit concludes with students designing an application that helps solve a problem of their choosing.  **Essential Questions:**   1. What strategies and processes can I use to become a more effective problem solver? 2. How do computers help people to solve problems? 3. How do people and computers approach problems differently? 4. What does a computer need from people in order to solve problems effectively?   **Standards:**  **CSTA K-12 Computer Science Standards (2017)**  **AP - Algorithms & Programming**  3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.  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-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.  3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools  **Summative Unit Assessment:** Propose an App Activity   |  |  | | --- | --- | | **Summative Assessment Objective** | **Assessment Method (check one)** | | Propose an App Activity: The class evaluates various web applications to analyze the specific problems that they were designed to solve, the inputs that they need to work, and the outputs they provide to users. The class concludes with observations of these apps as well as a teacher led discussion about the impact of apps on society. | \_\_x\_\_ Rubric \_\_\_ Checklist \_\_\_\_ Unit Test \_\_\_\_ Group \_\_\_\_ Student Self-Assessment  \_\_\_\_ Other (explain) | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DAILY PLAN Week One: The Problem Solving Process** | | | | | | |
| **Day** | **Objective (s)** | **DOK LEVEL** | **Activities / Teaching Strategies** | **Grouping** | **Materials / Resources** | **Assessment of Objective (s)** |
| 1 | **Lesson 1: Intro to Problem Solving**  Students will be able to:-Communicate and collaborate with classmates in order to solve a problem -Iteratively improve a solution to a problem -Identify different strategies used to solve a problem |  | Students will create accounts on Code.Org with teacher’s access code  The class works in groups to design aluminum foil boats that will support as many pennies as possible. At the end of the lesson groups reflect on their experiences with the activity and make connections to the types of problem solving they will be doing for the rest of the course. | W, I | Unplugged Activity -Pennies, Aluminum Foil  -Student CSD Handout/Notebook  -TEACHER HANDBOOK | Building an Aluminum Boat  End of Class Discussion/Journal Writing |
| 2 | **Lesson 2: The Problem Solving Process**  Students will be able to: \*Given a problem, identify individual actions that would fall within each step of the problem-solving process \*Identify useful strategies within each step of the problem solving process |  | This lesson introduces the formal problem-solving process that the class will use over the course of the year, Define - Prepare - Try - Reflect.. At the end of the lesson the class collects a list of generally useful strategies for each step of the process to put on posters that will be used throughout the unit and year. | W, I | -Video: Problem Solving Process  -Poster Board to Create Problem Solving Poster  -TEACHER HANDBOOK -Student CSD Handout/Notebook | Students will create a classroom poster  On page 2 of the activity guide, check that students have written down reasonable actions for each of the four steps in the problem they are trying to solve. |
| 3 | **Lesson 3: Exploring Problem Solving**  Students will be able to:  Apply the problem solving process to approach a variety of problems  Assess how well-defined a problem is and use strategies to define the problem more precisely |  | In this lesson the class applies the problem solving process to three different problems: a word search, a seating arrangement for a birthday party, and planning a trip. The problems grow increasingly complex and poorly defined to highlight how the problem solving process is particularly helpful when tackling these types of problems.  Day 3: Have students complete both the Word Search Handout Activity and the Birthday Party Guest Handout |  | **1. Word Search Activity Sheet**  2. Birthday Party Guest Handout  -Markers | Word Search: Objective  Find and circle all 8 words as quickly as you can! Classroom Race  **Birthday Party Guest Problem** Find the best possible arrangement of guests at the party. Draw your solution in the space below. |
| Days  4 &  5 | Day 4 & 5: Plan a Trip Project and Reflection: Each member of the group will individually be developing a plan for a trip that follows criteria they'll develop as a team. (May take a period and a half for students to play their trip).  After students have planned their trip, have them work on the problem solving process Notes |  | 3. Plan a Trip Handout  -Google Maps and Computer | **Plan a Trip**: Students will need to work online for this problem to use Google Maps - Website or some other tool that will allow them to plan a trip. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Week Two: The Problem-Solving Process** | | | | | | | |
| **Day** | **Objective (s)** | **DOK LEVEL** | **Activities / Teaching Strategies** | | **Materials / Resources** | | **Assessment  of Objective (s)** | |
| 6 | Identify a computer as a machine that processes information and Provide a high level of description of the input, output, and storage process of a computer. | 1 | **Lesson 4: What is a computer?**  Students will work in groups to create posters of what is a computer and what is not a computer by putting specific objects into categories (Is a Computer/Not a Computer). | | Copies of activity guide | | Journal Entry, Class Discussion, Class Poster | |
| 7 | Students will identify the inputs and outputs of common computing devices |  | **Lesson 5: Input and Output**  In this class students consider a number of computing devices to determine what types of inputs and outputs they use. Groups are assigned to a computing device and based on a teacher-provided definition of input and output, list the inputs and outputs of their device. To conclude the lesson the class examines common activities they do on a computing device and select the inputs and outputs used for that activity from the chart. | | Video | | Identify the inputs and outputs of common computing devices | |
| 8 | -Define processing as the work done (possibly by a computer) to turn an input into an output  -Define an algorithm as the series of commands a computer uses to process information  -Develop and iteratively improve an algorithm for processing information based on given  Constraints |  | **Lesson 6: Processing**  This lesson dives deeper into the concept of processing that was introduced as part of the definition of a computer. Pairs work together to put a deck of cards in order, a form of processing information. In the end, the class discusses what processing means within the context of solving information problems. | | Playing Cards | | Define an algorithm as the series of commands a computer uses to process information | |
| 9 | Describe how information can be processed to solve a particular problem.  Identify a possible source of a given input.  Determine what information should be stored on a device for later. |  | **Lesson 7: Apps and Storage**  Unplugged: This lesson covers the input and output aspects of computers in a context that is relevant and familiar to students: apps. The class evaluates various web applications to analyze the  Specific problems that they were designed to solve, the inputs that they need to work, and the outputs they provide to users. The class concludes with observations of these apps as well as  a teacher led discussion about the impact of apps on society | | App Exploration - Activity Guide | | Determine what information should be stored on a device for later. | |
| 10 | Students will decode patterns learn more about binary code. |  | **Binary Code:** Data in computers is stored and transmitted as a series of zeros and ones. How can we represent words and numbers using just these two symbols? Students will experiment with binary code activities. | | <https://classic.csunplugged.org/binary-numbers/>  https://classic.csunplugged.org/wp-content/uploads/2014/12/unplugged-01-binary\_numbers.pdf | | Students will solve a series of puzzles using binary code clues in this fun unplugged activity. | |
| **Week Three: Apps and Storage** | | | | | | | |
| **Day** | **Daily Objective (s)/Goals** | **Activities / Teaching Strategies** | | **Materials / Resources** | | **Assessment of Objective (s)** | |
| 11 | This project combines the two major themes of Unit 1, problem solving, and the input-output-store-process model of a computer. This project ties both themes to a broader goal of identifying real world problems and finding ways to use technology to help solve them. | **This is a one week Activity for Students**  Propose an App Activity: The class evaluates various web applications to analyze the specific problems that they were designed to solve, the inputs that they need to work, and the outputs they provide to users. The class concludes with observations of these apps as well as a teacher led discussion about the impact of apps on society.  **Student will…**  ● Work with a partner  ● Define a real world problem  ● Brainstorm ways an app could be used to help solve that problem  ● Identify the inputs / outputs / storage / processing used by your app  ● Share your ideas with another group for peer feedback  ● Incorporate feedback to create a final version of the app  ● Create a poster of your app to share with the class  **Student will submit...**  ● This completed Project Guide  ● Completed Peer Review  ● A poster of your app | | **Primarily an**  **Unplugged activity!**  **Distribute:**  -Apps and Problem Solving Guide/In Notebook  - Peer Review Sheet | | * The Project Guide * Completed Peer Review * A poster of your app * Presentation | |
| 12 | Students will continue work on their packet. Students will remember that large, complex, and poorly-defined problems are much harder to solve. Students will clearly define their problem by recording responses to the questions in pack. |
| 13 | Students will sketch a version of their app and indicate what all the different outputs are. Students will need to complete a peer review. Students will trade projects with another group and complete the peer review. Students will improve their app based on reviews. |
| 14 | Finalize the APP Poster or PowerPoint Based on the results of your peer feedback make any additions or changes you need to make to how you defined your problem or how you describe your app. Then make a poster that presents the final version of your app. Your poster needs to include the following information. |
| 15 | Present the Project to Class: The last step of this process is to present your app to your classmates. This may be done as a gallery walk or a full-class presentation. As you present your app make sure you’re ready to talk to your classmates the following points. |

**Teacher Name : Computer Science Department Subject : Computer Science Start Date(s): 45 Days Grade Level (s): 8th Grade Building :**

|  |
| --- |
| **Unit Plan: Interactive Animations and Games** |
| **Unit Title: Interactive Animations and Games**  In the Interactive Games and Animations unit, students build on their coding experience as they create programmatic images, animations, interactive art, and games. Starting off with simple, primitive shapes and building up to more sophisticated sprite-based games, students become familiar with the programming concepts and the design process computer scientists use daily. They then learn how these simpler constructs can be combined to create more complex programs. In the final project, students develop a personalized, interactive program. Along the way, they practice design, testing, and iteration, as they come to see that failure and debugging are an expected and valuable part of the programming process.  **Essential Questions:**  What is a computer program?  What are the core features of most programming languages?  How does programming enable creativity and individual expression?  What practices and strategies will help me as I write programs?  How do software developers manage complexity and scale?  How can programs be organized so that common problems only need to be solved once?  How can I build on previous solutions to create even more complex behavior?  **Standards: PA Core Standards, PA Academic Standards/Anchors (based on subject)**  3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.  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-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.  3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools  **Summative Unit Assessment:** Create a Computer Game Project   |  |  | | --- | --- | | **Summative Assessment Objective** | **Assessment Method (check one)** | | In this cumulative project for Chapter 1, students plan for and develop an interactive greeting card using all of the programming techniques they've learned to this point.  This lesson is the culmination of Unit 3 and provides students an opportunity to build a Game Lab project of their own from the ground up. | \_\_x\_\_ Rubric \_X\_\_ Checklist \_\_\_\_ Unit Test \_\_\_\_ Group \_\_X\_\_ Student Self-Assessment  \_\_\_\_ Other (explain) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week Four: Images and Animations** | | | | |
| **Day** | **Objective (s)** | **Activities / Teaching Strategies** | **Materials / Resources** | **Assessment of Objective (s)** |
| 16 | Identify how computer science is used in a field of entertainment | **Lesson One:** **Programming for Entertainment** (Unplugged The class is asked to consider the "problems" of boredom and self-expression, and to reflect on how they approach those problems in their own lives. From there, they will explore how Computer Science in general, and programming specifically, plays a role in either a specific form of entertainment or as a vehicle for self-expression.) | Review the research resources linked in  Code Studio  Print a copy of the activity guide for each  group of three students | In the activity guide, look at the "Interesting Fact or Use" section of the second page, make sure students have identified a use of Computer science in their chosen fields. |
| 17 | Reason about locations on the Game Lab coordinate grid  Communicate how to draw an image in Game Lab, accounting for shape position, color, and order | **Lesson Two**: **Plotting Shapes** (Unplugged: This lesson explores the challenges of communicating how to draw with shapes and use a tool that introduces how this problem is approached in Game Lab. The class uses a Game Lab tool to interactively place shapes on Game Lab's 400 by 400 grid. Partners then take turns instructing each other how to draw a hidden image using this tool, accounting for many of the challenges of programming in Game Lab.) | For the Teacher  Drawing Shapes - Exemplar  Sample Shape Drawing - Exemplar  For the Students  Drawing Shapes– Activity Guide  Drawing Shapes Activity Guide | Communicate how to draw an image in Game Lab,  accounting for shape position, color, and order |
| 18 | Use a coordinate system to place elements on the screen.  Sequence code correctly to overlay shapes. | **Lesson Three:** **Drawing in the Game Lab**  The class is introduced to Game Lab, the programming environment for this unit, and begins to use it to position shapes on the screen. The lesson covers the basics of sequencing and debugging, as well as a few simple commands. At the end of the lesson, the class creates an online version of the image they designed in the previous lesson. | Prepare projector or other means of  showing videos if you wish to watch as a  class | Use a coordinate system to place elements on the screen  Sequence code correctly to overlay shapes. |
| 19 | Use and reason about drawing commands with multiple parameters   Generate and use random numbers in a program | **Lesson Four: Shapes and Randomization**  Game Lab This lesson extends the drawing skills to include width and height and introduces the concept of random number generation. The class learns to draw with versions of ellipse () and rect() that include width and height parameters and to use the background() block to fill the screen with color. At the end of the progression the class is introduced to the random Number () block and uses the new blocks to draw a randomized rainbow snake. | Review the level sequence in Code Studio | Use and reason about drawing commands with multiple parameters  Generate and use random numbers in a program |
| 20 | Identify a variable as a way to label and reference a value in a program Use variables in a program to store a piece of information that is used multiple times | **Lesson Five: Variables** This lesson introduces variables as a way to label a number in a program or save a randomly generated value. The class begins the lesson with a very basic description of the purpose of a variable and practices using the new blocks. Afterwards, the class uses variables to save a random number, allowing the programs to use the same random number multiple times. |  | Identify a variable as a way to label and reference a value in a program  Use variables in a program to store a piece of information that is used multiple times |

**Teacher Name : Computer Science Department Subject : Computer Science Start Date(s): 45 Days Grade Level (s): 8th Grade Building :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Week Five: Images and Animations** | | | | | | | | | | | | | | |
| **Day** | **Objective (s)** | | **DOK LEVEL** | | | **Activities / Teaching Strategies** | | | **Materials / Resources** | | **Assessment of Objective (s)** | | | |
| 21 | Create and use a sprite  Use dot notation to update a sprite's properties | |  | | | **Lesson Six: Sprites** In order to create more interesting and detailed images, the class is introduced to the sprite object. Every sprite can be assigned an image to show, and sprites also keep track of multiple values about themselves, which will prove useful down the road when making animations. At the end of the lesson, everyone creates a scene using sprites. | | | Copies of activity guides for each student | | Create and use a sprite  Use dot notation to update a sprite's properties | | | |
| 22 | Explain how the draw loop allows for the creation of animations in Game Lab  Use the draw loop in combination with therandomNumber() command, shapes, and sprites to make simple animations | |  | | | **Lesson Seven: Draw Loops** Game lab This lesson introduces the draw loop, one of the core programming paradigms in Game Lab. The class combines the draw loop with random numbers to manipulate some simple animations with dots and then with sprites. Afterwards, everyone uses what they learned to update the sprite scene from the previous lesson. | | | Print and assemble the manipulatives. Prepare the video. | | Use the draw loop in combination with the randomNumber() command, shapes, and sprites to make simple animations | | | |
| 23 | Describe the connection between updating a  Sprite’s location properties and sprite movement on the screen. | |  | | | **Lesson Eight: Counter Pattern Unplugged**  This unplugged lesson explores the underlying behavior of variables. Using notecards and string to simulate variables within a program, the class implements a few short programs. Once comfortable with this syntax, the class uses the same process with sprite properties, tracking a sprite's progress across the screen | | | Prepare materials for Labels and Values:  index cards, post-its, or scraps of paper (2 in. by 2 in.) etc. (~ 50 per pair) | | Describe the connection between updating a sprite's location properties and sprite movement on the screen | | | |
| 24 | Identify which sprite properties need to be  changed, and in what way, to achieve a specific movement | |  | | | **Lesson Nine: Sprite Movement Game Lab**  By combining the Draw Loop and the Counter Pattern, the class writes programs that move sprites across the screen, as well as animate other sprite properties. | | | **Animating Sprites Video** | | Use the counter pattern to increment or decrement sprite  Properties & Identify which sprite properties need to be changed, and in what way, to achieve a specific movement | | | |
| 25 | Use the value of a Boolean statement to determine whether a command should be executed. | |  | | | **Lesson Ten: Booleans Unplugged** This lesson introduces boolean values and logic, as well as conditional statements. The class starts by playing a simple game of Stand Up, Sit Down in which the boolean (true/false) statements describe personal properties (hair or eye color, clothing type, age, etc). The class  then groups objects based on increasingly complex boolean statements, then looks at how conditionals can impact the flow of a program. | | | Copies of activity guides for each student | | Use the value of a Boolean statement to determine whether a command should be executed. | | | |
| **Week Six: Images and Animations** | | | | | | | | | | | | | | |
| **Day** | **Objective (s)** | | | | **DOK LEVEL** | | **Activities / Teaching Strategies** | | | **Materials / Resources** | | **Assessment of Objective (s)** | | |
| 26 | Use conditionals to react to changes in variables and sprite properties | | | |  | | **Lesson 11: Boolean s and Conditionals** Game Lab The class starts by using Booleans to compare the current value of a sprite property with a target value, using that comparison to determine when a sprite has reached a point on the screen, grown to a given size, or otherwise reached a value using the counter pattern. After using Booleans directly to investigate the values or sprite properties, the class adds conditional if statements to write code that responds to those Boolean comparisons. | | | Code Studio Access | | Use conditionals to react to changes in variables and sprite properties | | |
| 27 | Use conditionals to react to keyboard input  Move sprites in response to keyboard input | | | |  | | **Lesson 12: Conditionals and User Input** Following the introduction to Booleans and if statements in the previous lesson, students are introduced to a new block called key Down() which returns a Boolean and can be used in conditionals statements to move sprites around the screen. By the end of this lesson students will have written programs that take keyboard input from the user to control sprites on the screen. | | | Boolean Expressions - Video | | Use conditionals to react to keyboard input in Code Studio  Move sprites in response to keyboard input in Code Studio | | |
| 28 | Use an if-else statement to control the flow of a program.  Respond to a variety of types of user input. | | | |  | | **Lesson 13: Other Forms of Input:**  The class continues to explore ways to use conditional statements to take user input. In addition to the simple keyDown() command learned in the previous lesson, the class learns about several other keyboard input commands as well as ways to take mouse input. | | | Code Studio Access | | Respond to a variety of types of user input.  Use an if-else statement to control the flow of a program. | | |
| 29 | Use conditionals to react to keyboard input or changes in variables / properties | | | |  | | **Lesson 14: Project Interactive Card** In this cumulative project for Chapter 1, the class plans for and develops an interactive greeting card using all of the programming techniques they've learned to this point. This end of chapter assessment is a good place for students to bring together all the pieces they have learned (drawing, variables, sprites, images, conditionals, user input) in one place. Students should still be working with code that is easily readable and doesn't involve very many high level abstractions | | | copies of the project guide for each student | | In this cumulative project for Chapter 1, students plan for and develop an interactive greeting card using all of the programming techniques they've learned to this point. | | |
| 30 | Sequence commands to draw in the proper order  Apply an iterator pattern to variables or properties in a loop | | | |  | |
| **Week Seven: Images and Animations** | | | | | | | | | | | | | | |
| **Day** | **Objective (s)** | **DOK LEVEL** | | **Activities / Teaching Strategies** | | | | **Materials / Resources** | | | | | **Assessment of Objective (s)** | |
| 31 | Use the velocity and rotation Speed blocks to create and change sprite movements |  | | **Lesson 15: Velocity**  After a brief review of how the counter pattern is used to move sprites, the class is introduced to the properties that set velocity and rotation speed directly. As they use these new properties in different ways, they build up the skills they need to create a basic side scroller game. | | | | Game Lab | | | | | Use the velocity and rotationSpeed blocks to create and change sprite movements  Describe the advantages of simplifying code by using higher level blocks | |
| 32 | Detect when sprites are touching or overlapping, and change the program in response. |  | | **Lesson 16: Collision Detection** (Game Lab) The class learns about collision detection on the computer. Pairs explore how a computer could use sprite location and size properties and math to detect whether two sprites are touching. The class then uses the isTouching() block to create different effects when sprites collide, including playing sounds. Last, they use their new skills to improve the sidescroller game that they started in the last lesson. | | | | Collision Detection Exemplar  Game Lab | | | | | Describe how abstractions help to manage the complexity of code | |
| 33 | Explain how individual programming  constructs can be combined to create more complex behavior |  | | **Lesson 17: Complex Sprite Movement**  The class learns to combine the velocity properties of sprites with the counter pattern to create more complex sprite movement, such as simulating gravity, making a sprite jump, and allowing a sprite to float left or right. In the final levels the class combine these movements to animate and control a single sprite and build a simple game in which a character flies around and collects coins. | | | | Game Lab | | | | | Use sprite velocity with the counter pattern to create different types of sprite movement | |
| 34 | Describe how abstractions can be built upon to develop even further abstractions |  | | **Lesson 18: Collisions (**Game Lab ) The class programs their sprites to interact in new ways. After a brief review of how they used the ***is Touching*** block, the class brainstorms other ways that two sprites could interact. They then use is Touching to make one sprite push another across the screen before practicing with the four collision blocks (collide, displace, bounce, and bounce Off). | | | | Game Lab | | | | | Model different types of interactions between sprites | |
| 35 | Create and use functions for blocks of code that perform a single high-level task within a program |  | | **Lesson 19: Functions (**Game Lab)  This lesson covers functions as a way to organize their code, make it more readable, and remove repeated blocks of code. The class learns that higher level or more abstract steps make it easier to understand and reason about steps, then begins to create functions in  Game Lab. At the end of the lesson the class uses these skills to organize and add functionality to the final version of their side scroller game. | | | | Game Lab | | | | | Explain how abstractions allow programmers to reason about a program at a higher level  Create and use functions for blocks of code that perform a single high-level task within a program | |
|  |  |  | |  | | | |  | | | | |  | |
| **Week Eight: Building Games** | | | | | | | | | | | | | | |
| **Day** | **Objective (s)** | | | | **Activities / Teaching Strategies** | | | | **Materials / Resources** | | | | | **Assessment of Objective (s)** |
| 36 | Implement different features of a program by following a structured project guide | | | | **Lesson 20: The Game Design Process**  Game Lab This lesson introduces the process the class will use to design games for the remainder of the unit. The class walks through this process in a series of levels. As part of this lesson the class also briefly learn to use multi-frame animations in Game Lab. At the end of the lesson they have an opportunity to make improvements to the game to make it their own. | | | | Defender Game Project Guide | | | | | Implement different features of a program by following a structured project guide |
| 37 | Students will plan out their game Students should sample games, reviewing past work, and discussing as a group the type of game they'd like to build. If they want they can sketch ideas on scratch paper or in notebook. | | | | **Lesson 21: Using the Game Design Process** Game Lab In this multi-day lesson, the class uses the problem solving process from Unit 1 to create a platform jumper game. After looking at a sample game, the class defines what their games will look like and uses a structured process to build them. Finally, the class reflects on how the games could be improved, and implements those changes. | | | | Platform Guide Project Guide | | | | | Identify core programming constructs necessary to build different components of a game |
| 38 | Students are now ready to program their games on Code Studio through the next five days.  Create a plan for building a piece of software by describing its major components. | | | | **Lesson 22: Project - Design a Game**  Game Lab | Project  The class plans and builds original games using the project guide from the previous two lessons. Working individually or in pairs, the class plans, develops, and gives feedback on the games. After incorporating the peer feedback, the class shares out the completed games.  The scaffolding provided by the project guide and the practice  they have using it are intended to assist students in scoping their  projects and seeing their ideas through to completion. This project is  an opportunity to showcase technical skills, but they will also need to  collaborate with their partner, provide constructive peer feedback, and  repeatedly use the problem solving process as they encounter  obstacles along the way | | | | Use the project rubric attached to this lesson to assess student mastery of learning goals of this unit. | | | | | Independently scope the features of a piece of software |
| 39 | Implement a plan for creating a piece of software |
| 40 | Create a plan for building a piece of software |
|  | | | | | | | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week Nine: Building Games** | | | | |
| **Day** | **Objective (s)** | **Activities / Teaching Strategies** | **Materials / Resources** | **Assessment of Objective (s)** |
| 41 | Implement a plan for creating a piece of software | **Lesson 22: Project - Design a Game**  Game Lab | Project  Continue work on the Game: Students will define their sprites and variables in their game design packet. | Game Lab Studio | Students will be able to:  Independently scope the features of a piece of software |
| 42 | Create a plan for building a piece of software by describing its major components | **Lesson 22: Project - Design a Game**  Game Lab | Project  Define Functions: Functions: Your draw loop shouldn’t have a lot of complex code. Instead, break your program up into the major steps you’ll need for your game to work. The different behaviors you described for your sprites and variables should help you decide what these steps should be. Then describe what the code for that function should do. | Game Lab Studio | Create a plan for building a piece of software by describing its major components  Implement a plan for creating a piece of  software |
| 43 | Independently scope the features of a piece of software | **Lesson 22: Project - Design a Game**  LAST DAY TO WORK ON PROJECT | Game Lab Studio | Implement a plan for creating a piece of  software |
| 44 | Students will begin sharing their projects with the class | **Share PROJECT** : Give students a chance to share their games. If you choose to let students do a more formal presentation of their projects the project guide provides students a set of components to include in their presentations including: The original game they set out to build A description of the programming process including at least one challenge they faced and one new feature they decided to add A description of the most interesting or complex piece of code they wrote A live demonstration of the actual game |  | Use the project rubric attached to this lesson to assess student mastery of learning goals of this unit. You may also choose to assign the post-project test through code studio. |
| 45 | Students will finishing sharing their projects with class and fill out end of semester survey. |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |