ACTIVITY TITLE: DESIGNING AN INTERACTIVE GAME WITH SCRATCH

Activity code: ncEFGZG01

DURATION	120 minutes
AGE RANGE	11-12
TOPICS	3D CREATIVITY DESIGNING

Description of the project

This activity introduces students to the fundamentals of game development through the creation of an interactive video game using Scratch — a visual programming language tailored for young learners. The primary goal is to empower students to shift from being passive consumers of games to active creators of digital content, fostering both creativity and technical fluency.

Working in pairs or small groups, students will engage in the full game design cycle: brainstorming, programming, and refining a basic game that includes characters, animations, scoring systems, and interactive mechanics.

Key Concepts they will explore include:

- Block-based programming with Scratch (sprites, stages, and scripts)
- Core programming structures: loops, conditionals, and variables
- Event-driven logic using broadcasts
- Interactive design with keyboard input and real-time feedback
- Fundamental game mechanics: rules, goals, challenge, and user experience

This activity also introduces students to the basics of game logic and systems thinking by encouraging them to analyze what makes games enjoyable and fair. Students will apply computational thinking and collaborate to build a playable, shareable game.

By the end of the session, each student group is expected to have developed a functional Scratch game prototype and be able to explain its logic and structure to others through a brief demo and reflection.

Objectives: What will I learn?

- **Create a simple interactive game in Scratch** by using sprites, motion blocks, and basic user inputs,

in order to understand how visual programming can be used to build functional digital experiences.

- **Collaborate in pairs or small teams to design and test their game ideas**, so they can develop communication and teamwork skills essential for real-world STEAM projects.
- **Present and explain the logic behind their games**, using correct terminology and code structures, to enhance their ability to articulate technical decisions and develop digital literacy.
- **Apply programming concepts** such as loops, conditionals, and variables through hands-on coding challenges, to build fluency in foundational computational thinking.
- **Develop problem-solving, logical reasoning, and debugging abilities** by troubleshooting and iterating on their game projects, equipping them with essential skills for creative and technical fields.
- **Foster creativity and collaboration** through the game design process, encouraging the exploration of original ideas and peer feedback, as a way to express themselves while learning technical skills

Materials: What do I need?

1. **Provided by the teacher/institution:**
   - Computers or laptops with Scratch installed
   - Projector or demo screen (optional)
   - Sample Scratch game for demonstration
   - Stable internet connection
   - Printed Scratch block cards or cheat sheets
   - Access to basic art supplies or presentation tools
2. **Provided by students:**
   - Drawing/sketching materials (pencils, erasers, markers)
3. **Downloadable resources**
   - Printed Scratch cheat sheets or block cards (for guidance)
   - Player Performance Tracking Template + Rubric

Previous preparation

Before beginning the activity, the following steps should be taken to ensure a smooth and impactful session:

Set up digital tools and software

- Ensure that Scratch is installed and tested on all student devices or confirm access for browser-based use.

- If using the online version, create student accounts or open a shared class folder for saving and sharing projects.
- Test the projector or screen-sharing tools in advance if a live demonstration is planned.

Prepare teaching and reference materials

- Prepare and print Scratch block reference cards or quick-start guides (optional but highly useful for beginners).
- Create and have ready a simple sample Scratch game, showcasing movement, scoring, and basic interactivity to serve as a demo during the introduction.

Organize student groups

- Arrange students into pairs or small teams (2–3 per device) to encourage collaboration and peer support.
- Assign rotating roles within each group (e.g., Coder, Tester, Designer, Debugger) to promote balanced participation and shared responsibility.

Review key programming concepts and gaming elements

- Briefly review the core elements of a game (rules, goals, feedback, challenge) with the class.
- Optionally, ask students to share their favorite games and identify what makes them enjoyable — this primes discussion for the design process.

RESEARCH

Have a look at these resources

Today's learners engage with video games daily — but how do games actually work, and how can they influence the world around us?

In this activity, students will explore how block-based programming with Scratch enables them to design interactive games that respond to player actions using code logic. They'll learn how user input (like arrow keys or clicks) controls game elements like movement, scoring, or events. These mechanics mirror real-world systems used by professional developers.

But games are more than just entertainment. They can teach, inspire, and raise awareness. Game development combines coding, storytelling, art, and sound — making it a powerful tool for communication and expression.

Real-World Connections

- Games like Foldit help solve protein folding puzzles for medical research.

- Minecraft is used in schools to teach history, math, and even environmental awareness.
- Designers like Brenda Romero and Kim Swift broke barriers and built games that explore ethics, inclusion, and emotion.
- Scratch itself has been used to build educational games about recycling, space science, and cyberbullying awareness.

Key Questions to Investigate

Encourage students to reflect, discuss, and explore:

- What are the most important elements that make a game fun or meaningful?
- Can games be designed to solve real problems or teach concepts? How?
- How are diverse cultures and people represented in popular games? Is there room for improvement?
- Who are some female or gender-diverse figures in the gaming world or tech fields that inspire you?
- How can I design a game that is inclusive and educational?

Optional Research Task:

Ask students to search for a real-world game that teaches something or brings attention to a global issue (e.g., climate, inclusion, health). Have them summarize the game's mechanics and purpose before starting their own design.

CREATE

Some things you need before beginning

Why does this matter?

Have you ever faced a challenge and had to come up with a clever way to solve it? That's exactly what happens in this activity, but through a video game that you will design!

In this project, you'll create a game in Scratch where a character finds different challenges and uses special tools to overcome them. While doing this, you'll be learning how to solve problems, think creatively, and use coding to make ideas come to life. These are important skills, not just for video games, but for real life too!

Interesting facts and why this is important for you

- Coding is everywhere, even when you don't see it. Every time you use a phone, play a game, watch a video, or scan a bus card, coding is working in the background. Learning to code helps you understand how the digital world around you really works.

- Coding helps you think better. When you code, you break big problems into small steps, test ideas, and fix mistakes. These are the same skills you use when studying for a test, solving a puzzle, or planning a trip with your family.
- The apps and games you love are made with code. From TikTok to Minecraft, all your favorite digital tools and games were created by people who learned how to code. With coding skills, you can go from user to creator.
- Coding can solve real problems in the world. People use code to design robots that clean oceans, apps that help people learn new languages, or tools that help farmers grow food. Coding gives you the power to help others and take action on issues you care about.
- Anyone can be a coder, not just adults or computer experts. Many young people your age are already creating their own games, animations, and apps. Some even share their projects online and inspire others.
- Understanding code helps you stay safe online. When you know how digital tools are built, it's easier to spot fake websites, understand how your data is used, and protect your privacy.

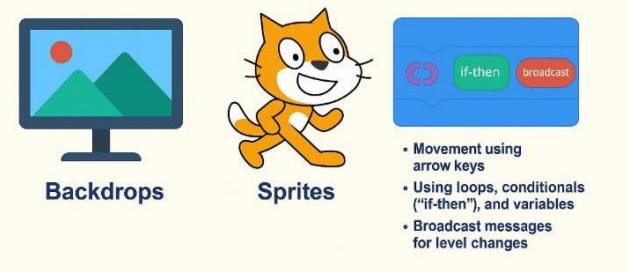
Now, follow these steps

Introduction and Demo

- Explore the Scratch interface: sprites, backdrops, blocks, and stage.
- Introduce the activity goal: create a simple interactive game (e.g., collect stars, avoid objects, complete a mission).
- Show a basic game as an example: "Make it Fly" and explain how user input changes what happens on screen.

Highlight key programming concepts:

- Movement using arrow keys
- Control logic: loops, conditionals ("if-then"), and variables
- Use broadcast messages to trigger level changes or effects



- Once elementary elements are introduced discuss on: "What decisions did the creator make to make this game fun or meaningful?" "Can you think of a mission that reflects a real-world issue—like protecting wildlife, teaching history, or recycling that could be taught by playing this game?"

Step. 1: Plan Your Game

- Let students choose a theme (e.g., space, jungle, recycling, history quiz, climate challenge) based on a traditional story from your region or family (e.g., a myth, proverb, or childhood tale): "How could this story inspire your main character's mission or world?"
- Let them define the game's goal: What should the player do to win?
- Research about a woman that has worked in this field. They will have to search for this information: What has she developed? What is the game about? What is the goal of the game? Why was it made? Does it contribute to raise awareness about a problem?
- Identify the main character of their own game (sprite): What is their role? What abilities do they have (abilities must be related with a sustainable world, therefore actions for a better world must be taken into account)
- Sketch or write your idea in 5 sentences (by answering all the questions). You are welcome to draw your main character and sketch the ideas in the drawing. This answer will have "Who is your main character? Could they be inspired by a real woman or cultural figure? How does your game reflect your identity, culture, or values? Could your game raise awareness about a problem or encourage players to take action?

Step. 2: Open Scratch and Set Up

- Go to scratch and click Create.
- Delete the default cat sprite if not needed.
- Add or design your own sprites and select a meaningful backdrop; take into account the following questions to help you design your sprite:
  - "Does your game's setting reflect a real-world place or environment (e.g., Amazon rainforest, urban park, ancient temple)?"
  - "Can your game show something from your culture or another one you like — like music, clothes, or a special place?"

Step 3: Make Your Sprite Move

- Select your sprite.
- Add blocks for movement:
  - when green flag clicked

- forever
- if key ____pressed → move ___steps

- Test it using arrow keys or spacebar.
- Extension "Could you include a timer or reaction tracker to show player progress or game impact?" "How does your character move? Does it move fast like it's in a hurry, or slowly like it's exploring?"

Step 4: Add Game Logic

Depending on your game goal:

🟢 **Collect items**

- Add items (e.g., coins, stars, trash)
- Use:

if touching ____→ change score by 1 → hide

🔴 **Avoid obstacles**

- Add enemies or traps
- Use:

if touching ____→ say "Game Over" → stop all

🎀 **Answer questions**

- Use:

ask ____and wait

if answer = ____→ say "Correct!"

< **Add sounds and events**

- Use:

play sound _____

broadcast "next level"

when I receive ___to change stage or difficulty

Reflection

"What's the message behind the gameplay? Could collecting items represent helping others or preserving nature?"

"Could your quiz teach social or ethical topics—like digital safety, rights, or empathy?"

Step 5: Create a Score System

- Go to Variables → Make a variable → name it "Score".

- Add these blocks to your sprite:

  set score to 0 (at game start)

  change score by 1 (when collecting items)

- Reflect on: "What does your score measure? Can it reflect real-life positive actions, like recycling or cooperation?" "How could you visualize the player's impact—through stars, lives saved, or actions completed?"

Step 6: Test and Improve Your Game

- Click the green flag to play.
- Test all actions and fix bugs: check if sprites respond correctly.
- Add extra features if needed:
  - Sound effects
  - Multiple levels
  - Timers
  - Animations
- If needed ask for feedback from peers. Ak them if they recognize diverse characters or values?, try modifying your game based on what matters to your players; challenge level, tone, message, and representation…

Step 7: Show and Explain Your Game

- Use your creativity to create a presentation about how, why you made it and how it is played. You can choose how you want to express it:
  - Make a small infographic, poster, or drawing
  - Create a slide, comic, or storyboard
  - Write a short description or record a short video/audio message

- **In the reflection, include ideas like:**
  - What traditional story from your region or family have you chosen?
  - What is your game about? What do players do?
  - What message or idea does your game share?
  - Who inspired your main character or story? What woman have you research about? Give information about what you found
  - Did you use anything from your culture, language, or community?
  - What programming blocks or features did you use — and why?
  - What did you find difficult, and how did you solve it?
  - What are you most proud of in your game?

Step 8: Analyze a player's experience with data

- **Track Game Results**

  Create a simple chart or table to record how players perform when playing your game. For each player, write down:

  - Did they complete the game?
  - How many times did they fail or restart?
  - How many points (score) did they get?

- **Create a Visual Chart**

  Use that information to create a bar graph, pie chart, or another simple visual representation of the data.

COMMUNICATE

- Each team gives a short live presentation to the class (or small groups). Something like the following can be said:
  - "Here's what our game is about…"
  - "We had trouble with ____but solved it by ____."
  - "We hope players feel ____when they play."

No need to repeat the presentation created in the step 7 — this is the time to summarize.

- Give Helpful Feedback; after each presentation, classmates can say:
  - "One thing I really liked was…"
  - "It would be even better if…"
  - "This game made me think about…"
- Encourage kindness, curiosity, and honesty — this is how real creators improve.
- Once feedback has been given:
  - Choose one thing to improve
  - Make a small update or bonus feature (e.g., fix a bug, add sound, adjust difficulty)

It is time to share!

Share your amazing work and inspire others!

#DesignscratchSTEAMbrace

- LinkedIn: https://www.linkedin.com/company/steambrace-project/posts/?feedView=all
- Instagram: https://www.instagram.com/steambrace_eu/
- X: https://www.instagram.com/steambrace_eu/

KEEP ON LEARNING

How can I make a similar project by myself?

- What made your game fun or engaging to play? *(Was it the challenge, the story, the characters, or something else that kept people interested?)*
- Did your original idea change while building the game? What did that teach you about designing or problem-solving? *(Did something work out differently than you planned — and was that a good thing?)*
- If you made a version 2.0 of your game, what new features would you add? *(A new level? Sound effects? A new goal or story twist?)*
- How could your game include more people or teach something important? *(Think about making it more accessible, fair, or focused on a meaningful message.)*
- Do you think your game could be used to explain a real idea or solve a real problem? How? *(What could someone learn from playing it? Could it make them think differently?)*
- Have you ever done a project like this before — something creative, coded, or made to teach others?
  (What made this project feel new or different for you?)

Try these next:

- Create a game for a younger student.
  Can you design a simple Scratch game to help younger kids learn something — like how to count, recycle, or stay safe online?
- Remix a famous game with your own twist.
  Take inspiration from a well-known game (like Pong, Pac-Man, or Mario) and reimagine it with new characters, settings, or rules that reflect your interests or culture.
- Make a story-based game with different endings.

What if your game had choices? Could the player decide what happens — like a choose-your-own-adventure?

- Design a character inspired by someone in real life. Turn a scientist, artist, athlete, or role model into a game hero. What powers or challenges would they face?

- Turn your game into a mini-exhibit. Create a poster, a comic strip, or a short video where you show your game and explain the meaning behind it — then share it at school or online.
- Explore game design careers. Look up what it's like to be a game designer, sound artist, animator, or accessibility expert. What part of game-making excites you most?

Which are other connected projects?

- Redesign your game for a different age group or theme (e.g., educational, environmental, historical).
- Add new features such as levels, sound effects, or a time limit to increase complexity.
- Explore advanced Scratch blocks, like sensing, clones, or timers.
- Connect with real-world roles: Research what professional game designers do and how they use similar concepts.
- Remix a classmate's game, adding your own twist while building on their ideas.

Connecting with the Arts: Theatre and Film

Scratch is not just a programming tool — it can also be a stage! Students can combine theatre or film arts with digital storytelling by creating interactive stories, script-based games, or short animated scenes that resemble plays or films.

Creative integration examples:

- Theatre: Students can program a game or animation where characters move across a "stage," deliver lines of dialogue, change backdrops (as stage scenery), and interact with the audience (e.g., using key presses).
    - Example: An interactive fairy tale where players choose what the characters say or do — students write dialogue, design costumes, and set lighting (backdrops).
- Film and Animation: Students can create a short interactive film with multiple scenes, transitions, and sound effects.

- Example: An animated mystery where a detective follows clues — the player controls the direction of the story like a director.

LINKS FOR FURTHER INFORMATION

Code.org Game Lab - Create simple games and interactive stories using block-based coding in a fun and visual way