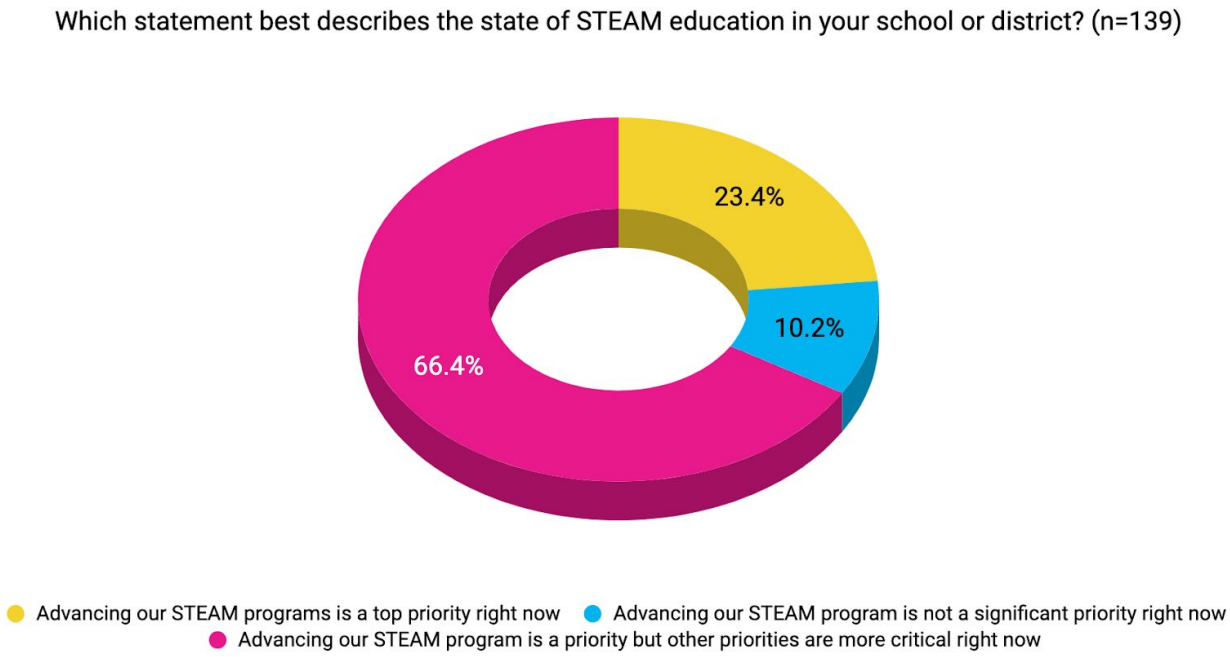
4 Ways to Make Coding & STEAM Playful

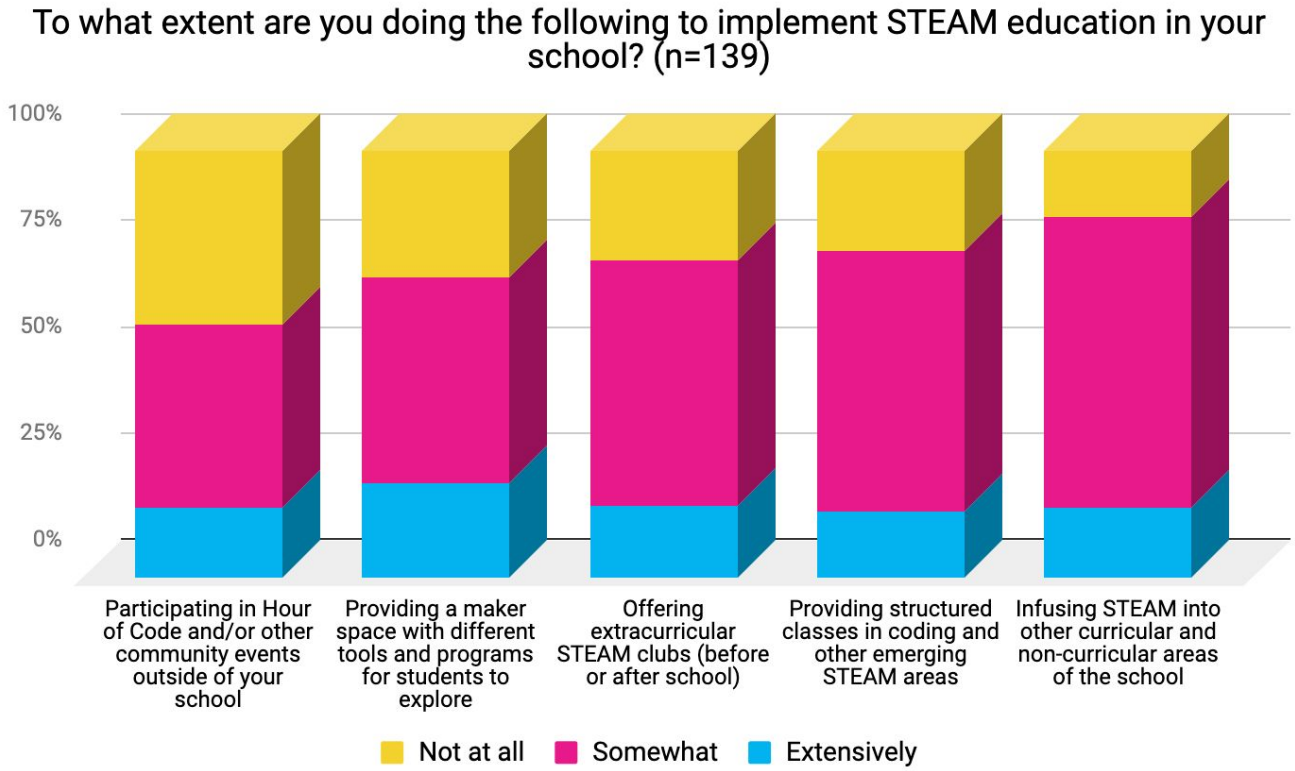
The Current State of STEAM in Schools

It's no secret that job opportunities in STEAM fields are growing and it is important to introduce STEAM and coding concepts to students at an early age. Unruly Studios, an EdTech company that focuses on coding for kids, surveyed over 139 elementary and middle school principals across the US to learn about STEAM needs in schools. 23% of respondents said that advancing STEAM education in their school was a top priority. 66% of them said it's a priority but other problems take precedence. The rest said that it was not a top priority at this time.



Although most principals in the survey agreed that STEAM is a priority, the challenge many cited is **how** and **when** they can fit STEAM lessons into the already busy school day.

Although dedicated STEAM classes are great, they are not realistic for every school and take time and resources to implement. To solve this challenge, schools are actively looking for creative ways to integrate STEAM into existing curriculum and afterschool programs. Unruly Studios asked the same group of principals how they are currently implementing STEAM in their schools and 84% of them said that they are infusing STEAM into other curricular and non-curricular areas of the school.



Introducing new STEAM and coding concepts into existing curriculum is challenging. One key to success is to make it approachable, fun, and playful for both teachers and students. Here are four strategies to make coding and STEAM more playful that can work for any subject and class setting.

1. Combining Play with Coding through Games

One way to mix coding with physical activity is to play active games. Students are naturally great at creating their own games on the playground and understanding the rules for games is the first step in computational thinking. After all, the rules for the games are the rules for the code!

A great way to introduce coding to students is through block coding. Block coding is a simplified programming language where the code is represented by easy to understand blocks that snap together to create rules. You can try block coding for free on a computer with the Unruly Block coding app, or with a block coding language called Scratch. Below is one example of how to use block coding to create a game.

Red Light, Green Light Game Example

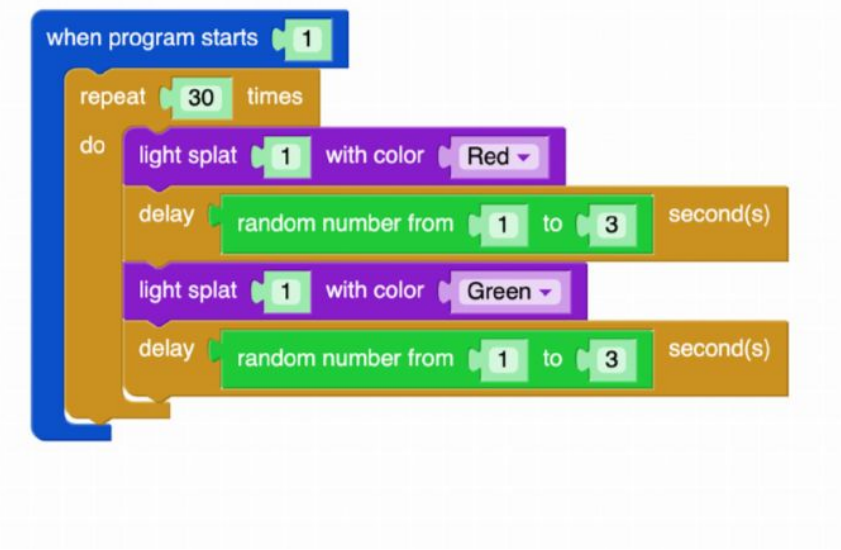
It's easy to get started with a game most people are familiar with, like red light, green light. The rules are:

1. Run when the light is green from one side of the room to another
2. Stop when the light is red
3. If you run when the light is red, you're out!
4. First person to reach the other side wins!

Tools required to play and code:

1. 1 iPad or Chromebook with web connection or any compatible device.
2. Download the Unruly Splats app on your ipad or visit our web app through a web browser

Unruly Splats are programmable floor buttons that students can code to make them light up, make sounds, and react when they are stomped on! For this game, you do not need physical Splats or additional hardware. You can play this all by using the "virtual Splats" on your device to indicate whether it's green or red. Here is what the code looks like:

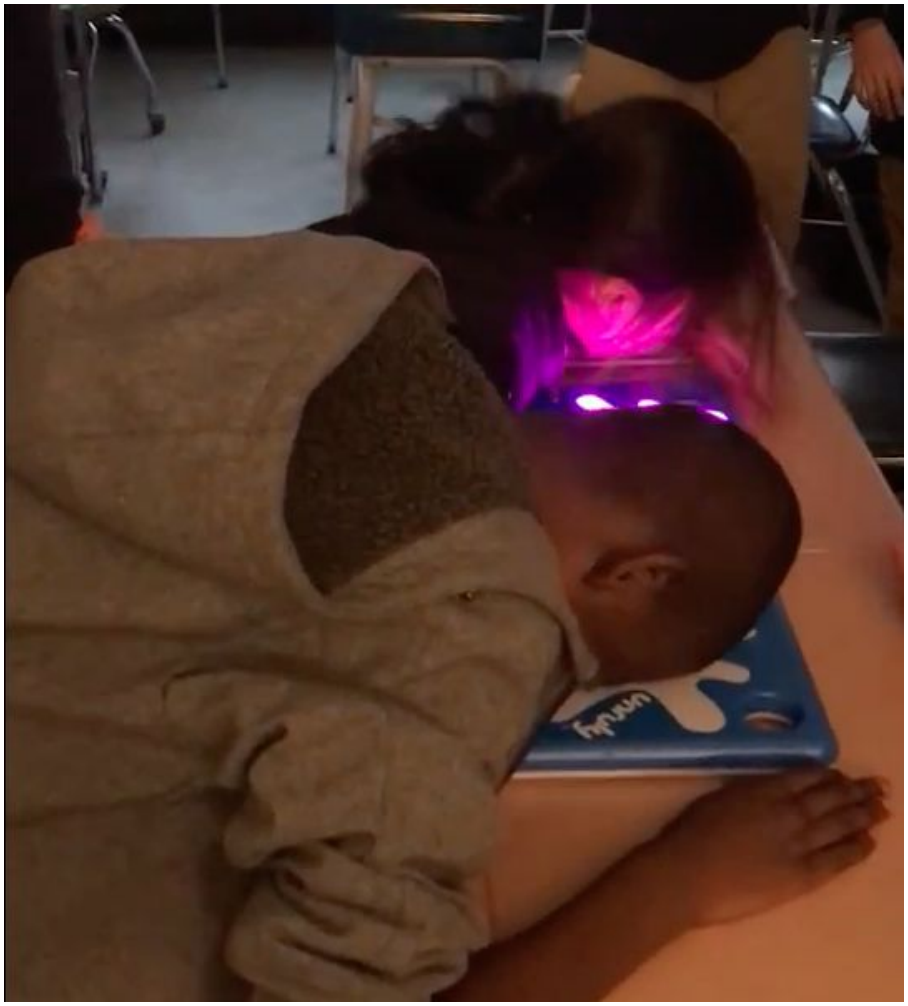


This program introduces nesting and the **Repeat** block. The rules for changing the Splat from red to green to red are nested inside the Repeat block. Within a Repeat block, blocks are repeated in a loop. This program repeats 30 times, meaning that the color changes from red to green 30 times. The color of the Splat is changed using the **Delay** and **Random Number** blocks. The Splat is lit up red and then turns green for random intervals between 1 and 3 seconds.

Modifying and Expanding Games

This is a simple way to introduce how you can transfer game rules into code rules. To go even further, you could ask the students to get into small groups and modify the rules for the same game to add some complexity. They could add a stopwatch to make it a speed competition or they add sounds when it turns from red to green, or they code a celebration sound and light show when someone gets to the other side. Get students to use their creativity!

This is just one example of how you can mix coding and physical activity with games, and there are lots of other great resources that support this. For more lesson plan ideas, check out our educator page. If you're looking for ideas to teach computer science concepts without any devices, CS Unplugged is another great resource.



Pictured above: *Students modified the code for Whack-a-Mole to create their own version of Bobbing for Apples using their heads instead of their feet or hands to activate the Unruly Splats*

Get More Coding Lesson Plans

2. Keep it Open-Ended

The real fun begins when students get to make their own rules and create their own games. This requires giving students more control over their own learning, also known as student-centered learning. It also means that students' work may look very different, and that's great since students learn and think differently.

Benefits of Student-Centered Learning

Student-centered learning enables students to take an active role in their own learning and focuses on giving students collaboration and problem solving skills. Research from Stanford University indicates that implementing a student-centered learning approach in schools can lead to higher grade averages, especially in schools with underserved students.

Let Your Students Make the Rules

At first, it may be overwhelming to have a completely open-ended lesson. A great way to get started initially is by setting some boundaries to an open-ended activity. The boundaries can be as narrow or wide as you feel comfortable with. Then, once your students feel confident, they can take it even farther! One example of how to do this is by prompting students to create a game that includes jumping and keeping score. Students can start off brainstorming different game ideas on pen and paper or in small groups. Once they have decided what they want their game to be, they can start breaking down the rules and coding them. At the end of the class, you might have 30 different games that the students can all play together or in stations.

Unruly Studios has seen students make hundreds of games, including zombie tag, radioactive whack-a-mole, bobbing for apples, and karate chop challenge. Giving students the freedom to make their own rules not only deepens their learning but also gives them a sense of ownership and pride - they have made something all on their own!

***Pictured above**: A student at an end-of-year showcase showing their parents how to play a game they invented in class called Karate Chop Challenge.*

Celebrating Missteps

An essential part of STEAM is being able to problem solve. Mistakes will happen and not everything will come easy. Celebrating missteps and using it as an opportunity for learning is a great way to empower your students. If a student has a bug in their code and just can't seem to figure out what's going on, a great way to celebrate is to use it as a class challenge. Crushing bugs is fun! Get the class together and work in small groups to see who can find the bug. Some teachers like to celebrate missteps by offering bug stickers to any students who find bugs in their code. It's one way to celebrate bugs as a normal and expected part of the problem-solving process.

3. Take on a Coaching Role

Educators do not always have all the answers, and they do not have to! Especially when schools are making efforts to integrate STEAM concepts into other classes. One great way to make learning STEAM more playful is by taking on a coaching role and giving students more responsibility over their learning.

Coaching vs. Teaching

The difference in this approach is a coach will guide the class through questions and discussions to get students to come up with the answers themselves. This is happening naturally in classes like PE or in sports, but it is being adopted successfully in other STEAM classroom environments. This gives students a more active role in their own learning.

Coaching in Computer Science

One great example of this is a computer science teacher in Hawaii named Michael Fricano, who teaches his K-2 students to code their own physically active games.

To start, Michael leads his class through a few coding tutorials. Once they are comfortable with the basic functions, they start brainstorming games as a group and working on them independently. This could be through an online programming environment like Scratch or through the Unruly App.

***Pictured above**: Students in Mr. Fricano's class collaborate together in small groups to code their Splats*

Students in his class pick up coding very quickly in this format. Michael recalled a moment where a first grade student asked him how to code a delay in her timer. This is something that they had not covered in the tutorials yet so Michael did not know how to do that on Unruly Splats. He told her he would try to figure it out after class but suggested she try to figure it out first.

Michael recalls, “Typically a student will just move on if they can’t figure it out, but five minutes later, I hear ‘Mr. Fricano, I think I figured it out!’. She showed the rest of the class how she did it and then a lot of the other students incorporated it into their programs. She was so proud that she did it herself”.

As Michael’s story illustrates, teachers do not have to have all of the answers. If you guide your students in the right direction and give them the opportunity to discover the answers on their own, it helps students develop problem solving skills. In many cases, students end up learning and retaining more.

> *"If you want to build a ship, don't drum up the men to gather wood, divide the work, and give orders. Instead, teach them to yearn for the vast and endless sea."*
>
> ***— Antoine de Saint-Exupéry***

4. Encourage Collaboration and Competition

According to a 1992 study by Johnson & Johnson, cooperative activities compared with individualistic efforts typically results in greater psychological health, social competence, and self-esteem. With so much emphasis on Social Emotional Learning (SEL) in schools today, having students work together to solve problems is a great way to teach students how to work together as they learn to code.

Working in Small Groups

Instead of having students sit in their own desks facing the front of the room, try organizing students in small groups so they can work together to solve a problem. However groups are organized, students should be talking with one another and collaborating on a task. This setup will facilitate students to share their ideas, ask questions, and play together.

Pictured above: *Students working in small groups of three to code and play games on Unruly Splats. One student is controlling the ipad and can make updates to the game in real-time.*

Using Competition for STEAM Learning

Creating competitions is another great way to combine coding with fun and activity. This could be a class competition to see who can get the highest score or who can find a bug the fastest or it could be in teams to see which class jumps more times in one day. For example, Unruly Studios ran a program with five schools across the US and at the end of the pilot, they completed a one-week event called the “Splat Olympics” with their classes. During the event, students played a series of games designed to get them moving. Students had fun and got silly, all while being introduced to fundamental STEAM concepts.

***Pictured above**: A group of students who participated in a PE pilot program where they integrated coding concepts with physical activity.*

Don’t Forget to Have Fun!

These four tips will make learning STEAM concepts more active and playful but it will also help with integrating coding into other classes. By integrating these concepts into existing curriculum, there is no need to take time away from other subjects or create dedicated STEAM or coding curriculum. STEAM should never feel like a chore. By making STEAM a fun addition to any classroom, you can pull interest from many diverse learners. For more information on how to integrate coding into your existing school in a fun, active way, visit www.unrulysplats.com.

Learn More about Active Code