SmartGurlz
teaching girls to code

Teaching Girls to Code

With SmartGurlz Products

TEACHING GUIDE

Introduction

a. Overview

This guide is for teachers, parents, and instructors who plan to teach coding with SmartGurlz curriculum. It contains materials to help teachers implement the curriculum in their classrooms, including connections to Computer Science Teachers Association (CSTA) K-12 Standards and 21st Century Standards.

A degree in computer science or prior coding experience is not a requirement to teach coding with SmartGurlz coding curriculum. Everything is explained in simple terms to enable easy facilitation and an enjoyable learning experience.

b. Purpose

The teacher's guide was created to help educators succeed in using SmartGurlz curriculum. The curriculum is designed to help students learn and apply the computer science concepts. It engages learners by allowing them safely and easily explore coding in a fun way.

c. Curriculum Structure

The curriculum is divided into 12 Units, and each unit is divided into 3 sections: concept introduction, hands-on activity, and assessment. Since teachers know their students more than the creators of the curriculum, teachers are encouraged to use other available tools and resources that can enrich the learning experience of students, such as posters and videos.

Each Unit is designed to be taught in 50 minutes (including the hands-on activities) and the unit content can be delivered as follows:

- **Concept Introduction (10 min):**
  Introduce new a new coding concept and explain the unit's hands-on activities.

- **Activities (30 min):**

  First, divide students into groups or have them work individually. Next, provide them with lesson materials (listed in each lesson unit). Afterwards, ask students to complete the hands- on activity as it's described in the curriculum.

- **Assessment Discussion (10 min):**
  - Students discuss their observations and comment on the hands-on activity
  - The teacher summarizes the topic and assesses the students with reflective questions. The teacher should reinforce the science and coding concepts the students should have observed during the activity.

As a general rule each unit (or module) is dependent on the previous units. In each unit you will find a set of learning objectives which will identify computer science skills and other concepts addressed in that unit.

In addition, the curriculum includes a lot of small challenges, sometimes referred to as missions. They are designed to teach students problem-solving and programming skills in the context of gaming and story-telling.

d. Materials:

- Tablets (iOS or Android) with SmartGulz's SugarCoded app installed
- SmartGurlz's Siggy robots
- 9 Volts battery for each robot (rechargeable batteries are recommended)
- Supplies, such as solid objects to serve as obstacles or a tape to mark out a maze field on the floor.

Getting Started

A. What does "SmartGurlz Curriculum" teach?

- Robot Math
    i. Distance, speed, angles
- Coding skills
    i. Sequence of coding commands
    ii. Wait commands
    iii. Decision-making structure: repeat
    iv. Code troubleshooting
- Soft skills
    i. Problem-solving
    ii. Collaboration and teamwork
    iii. Communication and presentation

B. Setup and class preparation

Though "SmartGurlz Curriculum" could be used to teach students one-on-one, it is originally designed for teachers with students working together in small groups.

Ideally, each pair of students works with one tablet and one Siggy robot. Students can also be grouped in teams of 3 or 4 and take different roles, such as coding, getting the robot ready, setting up the challenge field, etc.

Set up example:

- Fully-charged tablets with SugarCoded app installed on it
- Fully-charged Siggy robots
- Supplies (depends on the lesson unit)

In addition to setting up students' workstations, the teacher may occasionally need to set up teaching aids, such as PowerPoint slides, short videos, etc.

C. SmartGurlz products:

- Hardware: Siggy robots
    i. Each Siggy robot needs a 9 volts battery. Since students will be using the robot a lot during the course, it is recommended to invest in rechargeable batteries.
    ii. Tablets. They can be either iOS or Android tablets.

- Software: SugarCoded Application
    i. Tablets must have the SugarCoded app downloaded and installed on them. The application is available on both Apple's App Store and Google Play Store free of charge.

        ⇨ To make sure everything goes smoothly, it is recommended for teachers to download the app on each tablet, enable Bluetooth, and test the tablet's connection to the robots.

D. Final Project

1. Project requirements:

- Based on a specific topic of interest, each group of students has to design an obstacle course or a challenge field as their final project. They will also have to code their robot to complete that mission.

- In order to foster creativity and imagination, missions can be based either on a real-world problem or a fiction story.

2. Project presentation:

- The team will prepare a short presentation to explain to the teacher and the rest of the class: 1) their mission, 2) decisions they had to make as a team, 3) challenges they met and how they solved them, and 4) their most exciting moments during the course.

- Finally, the team must do a demonstration and show off their robot completing the mission.

- The robot must move autonomously. No remote control allowed.

During the final project class periods, the teacher's role is to guide students and facilitate friendly interactions. Project activities should be completed by students to experience and take charge of the projects.

Students should be free to choose the best way to present their projects. They can use posters, PowerPoint slides, or a simple explanation.

Standards Alignment

CSTA K-12 Computer Science Standards, Revised 2017	SmartGurlz Curriculum
1A-CS-01 K-2 Computing Systems Select and operate appropriate software to perform a variety of tasks. Recognize that users have different needs and preferences for the technology they use.	Students learn how to find and use the SugarCoded app. By the end of the course, students know different uses of the app. They also know the intended purpose of each app's functionality.
1A-CS-03 K-2 Computing Systems Describe basic hardware and software problems using accurate terminology.	Throughout the course, students participate in trial and error challenges. They learn how to use the robot's capabilities to troubleshoot their codes, such as using sound and light feedback to test the code.
1A-NI-04 K-2 Networks and Internet (Cybersecurity) Explain what passwords are and why we use them. Use strong passwords to protect devices and information from unauthorized access.	Teachers who use this curriculum are encouraged to protect students' tablets with passwords. They are encouraged to explain to students why passwords are important and why they need to be used.
1A-DA-05 K-2 Data and Analysis Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.	The curriculum teaches students how to save their codes through the SugarCoded application, and retrieve the information at a later time for modification or deletion.
1A-DA-07 K-2 Data and Analysis Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.	Through SugarCoded application's test mode, the curriculum teaches students to draw path lines and patterns for the robot to follow, run the simulation to visualize how the robot will behave, and then run the code on the robot to observe if it behaved as predicted.

1A-AP-08 K-2: Algorithms and programming Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.	Students are given coding missions based on every day activities, such as going to a grocery store. Each mission requires students to write an algorithm that enables the robot to follow specific instructions in order to complete the mission.
1A-AP-09 K-2: Algorithms and programming Model the way programs store and manipulate data by using numbers or other symbols to represent information.	The curriculum teaches students to code with variables, which are symbolic names that store information referred to a value. The value can be a number, such as the speed of the robot.
1A-AP-10 K-2: algorithms and programming Develop programs with sequences and simple loops, to express ideas or address a problem.	Students develop unique programs to complete coding missions. Students code in sequence and sometimes use "repeat" commands for mission problems that require repeated codes.
1A-AP-11 K-2 Algorithms and Programming Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.	The curriculum challenges students with a final project. They have to understand the project requirements while they layout out steps needed to successfully finish the project.
1A-AP-12 K-2 Algorithms and Programming Develop plans that describe a program's sequence of events, goals, and expected outcomes.	Students plan a presentation and describe their codes to others. They are to explain the purpose of their program and do a demonstration by running the code on the robot.
1A-AP-14 K-2 Algorithms and Programming Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.	The curriculum's coding missions are designed to walk students through 'trial and error' process in order to learn how to find and fix errors in their codes; which contain sequence and loop commands.
1A-AP-15 K-2 Using correct terminology, describe steps taken and choices made during the iterative process of program development.	During the final project presentation, students describe their codes to others. They explain, using coding terms, choices and decision they had to make.
1B-CS-03 3-5: computing systems Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	As students discover problems during the testing phase, they have to find solutions in order to complete their missions.
1B-DA-07 3-5: Data and analysis Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.	Through SugarCoded application's test mode, students write codes, run the simulation to visualize how the robot will behave, then they run the code on the robot to observe if it behaves as predicted.
1B-AP-08 3-5: Algorithms and Programming Compare and refine multiple algorithms for the same task and determine which is the most	The curriculum is designed to enable students learn from each other. Challenges, such as missions and the final project, are the same for all students, but they

appropriate.	have to creatively come up with unique solutions. Students determine the most appropriate algorithms by observing each other's performances.
1B-AP-09 3-5 Algorithms and Programming Create programs that use variables to store and modify data.	Students use variables in their codes, such as the robot's speed variable.
1B-AP-10 3-5: Algorithms and Programming Create programs that include sequences, events, loops, and conditionals.	In order to complete missions, students write their code in sequence and use "repeat" commands for mission problems that require repeated codes.
1B-AP-11 3-5 Decompose (break down) problems into smaller, manageable sub problems to facilitate the program development process.	The curriculum's missions and final project teach students to break coding challenges into small tasks that facilitates development and teamwork.
1B-AP-13 3-5 Algorithms and Programming Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Students have to collaborate with peers during the final project. They work together to design and develop a solution that meets the project requirements.
1B-AP-15 3-5 Algorithms and Programming Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	The curriculum's coding missions are designed to walk students through 'trial and error' process in order to learn finding and fixing errors in their codes. They set robot performance goals, write codes, and test them on the robot. If robot's performance is not satisfying, they fix errors, and test again.
1B-AP-16 3-5 Algorithms and Programming Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	During the final project, students experience, with teacher's guidance, friendly collaboration with peers. They work together through solution brainstorming, design, implementation, and presentation.
1B-AP-17 3-5 Algorithms and Programming Describe choices made during program development using code comments, presentations, and demonstrations.	During the final project presentation, students describe their codes to others. They explain, using coding terms, choices and decision they had to make. They also do a demonstration by running the code on the robot.
1B-IC-20 3-5: Impacts of computing Seek diverse perspectives for the purpose of improving computational artifacts.	One of the objectives of the final project presentation is to allow students to learn from each other. The challenge is the same for all students, but they have to creatively come up with different solutions. During presentations, students gain different perspectives in solving coding challenges.

21st Century Skills Standards	SmartGurlz Curriculum
1: Core Academic Subject Mastery Students who can think critically and communicate effectively must build on a base of core academic subject knowledge. For this reason, core academic subjects are a bedrock component of the P21 Framework for 21st Century Learning. All 21st century skills can and should be taught in the context of core academic subjects.	Though the main focus of the curriculum is to teach students coding using robots, students apply math and physics concepts throughout the course. The curriculum touches on concepts such as distance, angles, speed, sound, and light.
2: 21st Century Skills Outcomes In addition to core subject mastery, are schools helping students become... ● Critical thinkers? ● Problem solvers? ● Good communicators? ● Good collaborators? ● Information and technology literate? ● Flexible and adaptable? ● Innovative and creative? ● Globally competent?	● The curriculum has plenty open-ended problems to encourage critical thinking among students. ● Unit missions and the final project are designed to better students at problem-solving. ● Students have to prepare and perform a presentation for a final project. This experience trains them to be better at communication. ● Students work together in teams and they learn friendly collaboration as they complete missions and the final project. ● Students use tablets, the SugarCoded application, and robots throughout the course. ● Students learn flexibility and adaptability through challenges in the trial and error process ● Though the curriculum has units in sequence with specific skills to learn at each step, there is room for innovative and creative ideas. Students find unique solutions for missions and the final project. ● Learning both coding and robotics skills equips students with the right skills to be globally competent.