blackbird

CURRICULUM DESCRIPTION

OVERVIEW

'**Games and Animations'** - A project-oriented programming curriculum that introduces new programmers to the skills required to build fun animations and games on the HTML canvas using JavaScript. Students develop real and applicable programming proficiencies like: declaring and using variables, conditional statements, data types, objects, loops, arrays and functions through engaging lessons and projects.

The Games and Animations curriculum introduces these skills on a platform that includes an easy-to-use code editor, helpful error messages and robust documentation that lowers the learning curve for users. Students are quickly able to manipulate the canvas with graphics like shapes, images, and colors. Before long, students learn to build animations, develop logic statements and complex data structures within their programs. Students combine these skills to develop fun games in guided projects at the end of each instructional segment.

ORGANIZATION

'**Units** - The 'Games and Animations' curriculum is divided into two large units of study (soon to be three!). Unit 1 focuses on the basic skills required to build simple games. Unit 2 builds on the skills developed in Unit 1 by introducing more abstract concepts like for loops and complex conditional statements.

Stages - Within a unit, students progress through 5 thematic stages. Each stage introduces a handful of fundamental programming skills. Stages conclude with a ‘Challenge’. Students must demonstrate mastery of the skills introduced by passing the challenge to move on to the next stage.

Lessons – Each stage is composed of:

- 5 'Teaching lessons' that introduce new programming skills.
- 5 'Practice lessons' that provide extra practice.
- 1 'Review lesson' that allows students to review material in a condensed format.
- 1 'Vocabulary quiz' - test if students have retained important vocabulary.
- 1 'Challenge lesson' that tests what students have learned in the stage.

Lessons vary in length but grow in complexity as the student progresses through a Stage. Every lesson (except Challenge lessons) include a ‘Show me’ feature that provides the correct line of code so students are never stuck on a task.

Guided Projects - The transition from curated lessons to a sandbox environment can be intimidating for students. Each stage has three self-directed 'Guided projects' that ask students to use the skills they’ve learned to create a complex computer program in the Workshop.

Workshop – Blackbird School has a robust sandbox environment (the Workshop) where students can develop their own code using what they’ve learned in the lessons. Students can explore model programs in the Nest, share code with friends and classmates on the platform, and receive feedback on projects from instructors.

Educator Dashboard – The platform includes a full-featured learning management system (LMS) where teachers can create classes and invite students. Teachers get detailed information about student progress, like where a student is in the curriculum and how much time they’ve spent actively engaged with lessons and projects. Teachers can view student’s work in the Workshop and can comment on their code allowing for quick and responsive feedback.

Teacher Resources - In addition to a supportive onboarding process, teachers have access to a growing library of videos and documents that help them get the most out of Blackbird in their classroom.

CURRICULUM OUTLINE

UNIT 1 - SIMPLE GAMES

- **Stage 1 – The canvas**: Students learn about the HTML canvas and how to place points, lines and text on it.
- **Stage 2 – Shapes and colors**: Students learn how to use functions to create colorful shapes and images on the canvas.
- **Stage 3 – Animation**: Students learn how to use the animate function to create engaging programs on the canvas.
- **Stage 4 – Variables and objects**: Students learn how about the data structures variables and objects.
- **Stage 5 – Simple games**: Students learn how to incorporate the skills they've learned to program simple games.

UNIT 2 - MORE GAMES

- **Stage 6 – For Loops**: Students learn how to construct repeating sections of code called for loops.
- **Stage 7 – Conditional statements**: Students learn more about different types of logic statements.
- **Stage 8 – Arrays**: Students learn to organize data in a list called an array.
- **Stage 9 – Loops and arrays**: Students learn how to build lists using loops
- **Stage 10 – Advanced games**: Students learn how to incorporate the skills they've learned to program more complex games.

LOGISTICS

- **Time requirements** - One semester or about 16-18 weeks.
- **Technology requirements** – Computers with internet (desktop or laptop), classroom with projector is a plus.