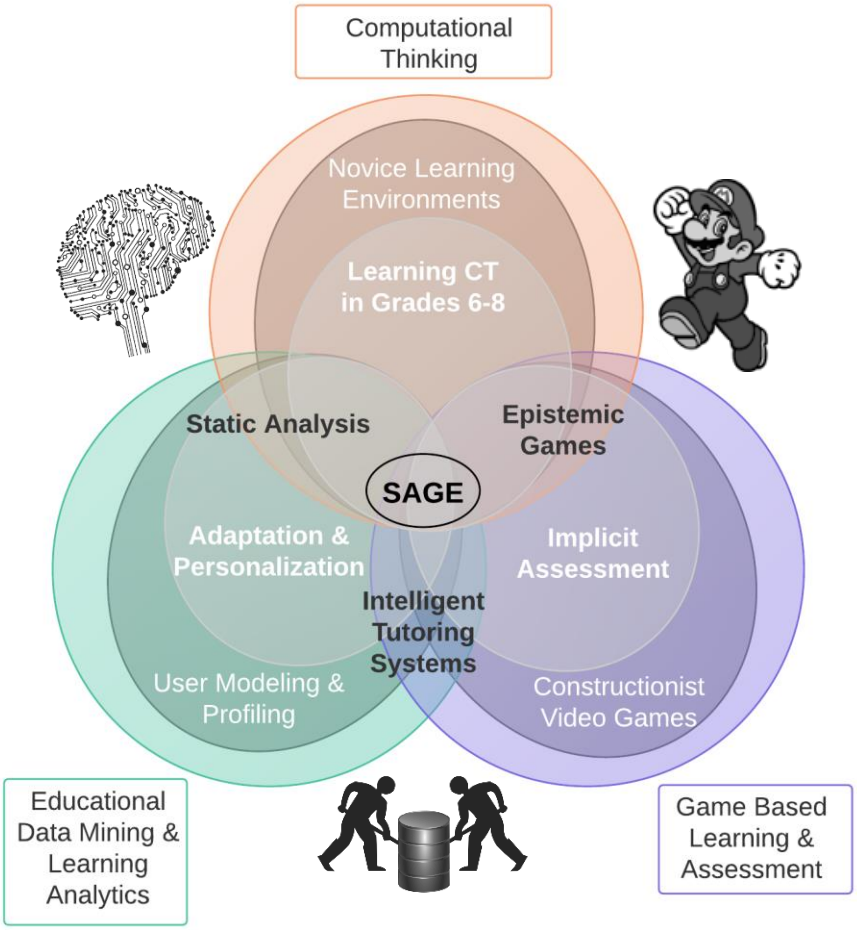


Social Addictive Gameful Engineering (SAGE)

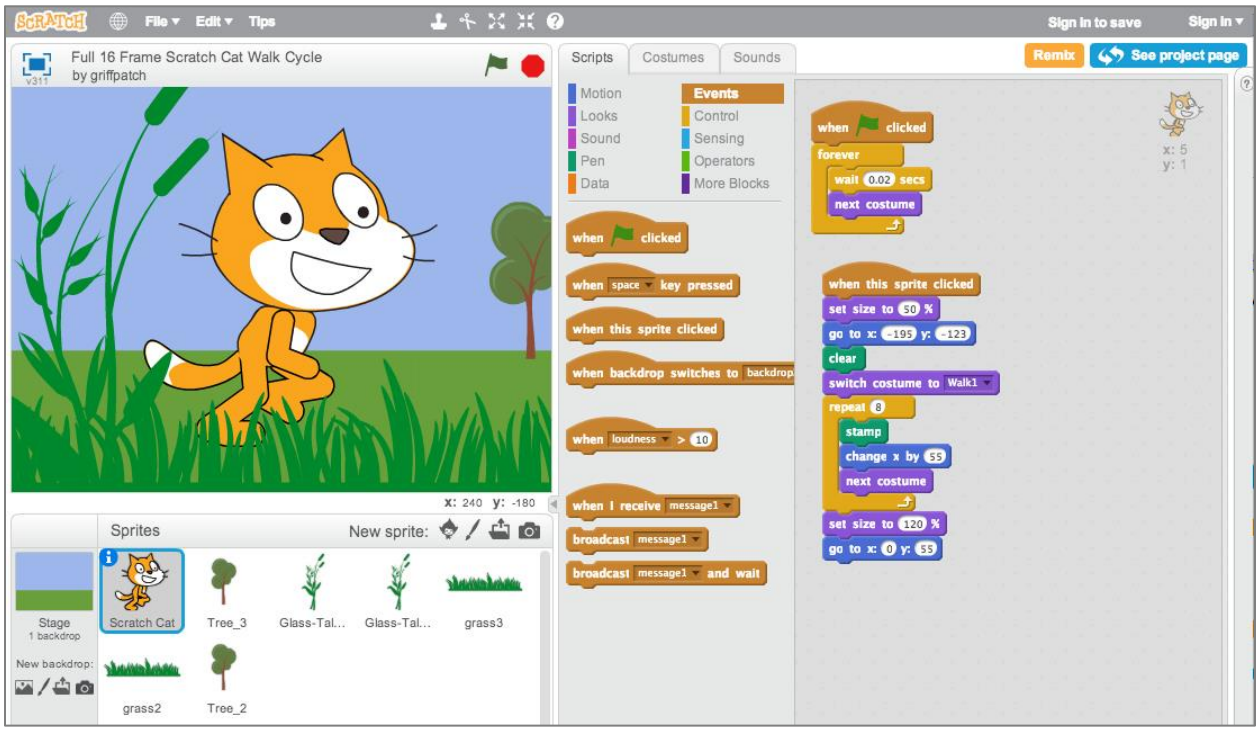
a collaborative game-based learning and assessment system which infuses computational thinking in grade 6-8 curricula

COLUMBIA UNIVERSITY
The Programming Systems Laboratory

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

What is Computational Thinking?

In 2006, writing as the head of the Computer Science Department at Carnegie Mellon University, Jeanette Wing articulated a call for educational action that releases her subject from the constraints of its scholarly seclusion. Stipulating that ubiquitous computing is to today as computational thinking (CT) is to tomorrow, she frames CT as a "universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use." Although a consensus definition for the term has remained elusive, CT aims to describe how humans, not computers, think logically, algorithmically, procedurally, analytically, recursively, and creatively, with an engineering-attunement. This hybrid thinking helps us devise models and representations which enable problem solving, data transformation, and system automation. The primacy of conceptualization, not programming, of ideas, not artifacts, frees computer science fundamentals from their constituent levees, allowing strong mental models to flood diverse disciplines both in the sciences and humanities, as well as in in their applied counterparts, medicine, law, and journalism. Of course, computers retain their central role: "Abstractions are the 'mental' tools of computing. The power of our 'mental' tools is amplified by the power of our 'metal' tools. Computing is the automation of our abstractions." The emphasis on semantics rather than syntax, however, engenders a common language in which to explore the possibilities of computing, and a means by which to confront the dwindling wonderment experienced by digital natives commonly capable of using enveloping technologies, but devoid of digital fluency.

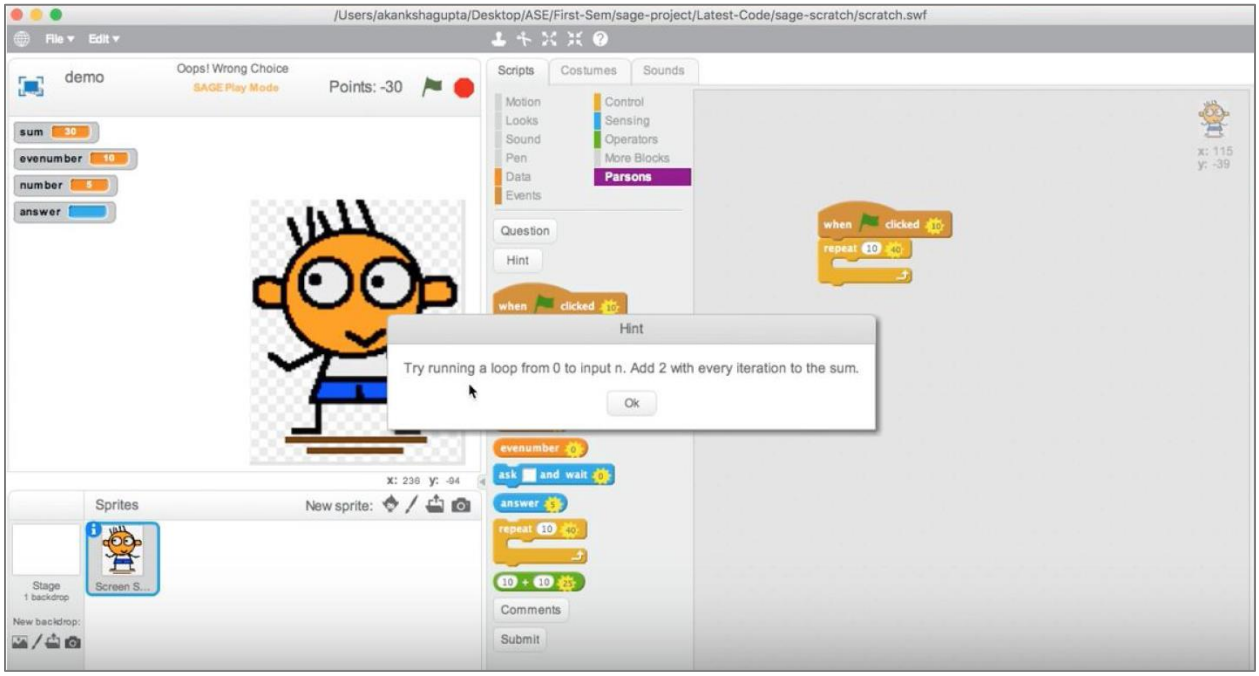


Scratch, the preeminent novice learning environment for grades 6-8

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

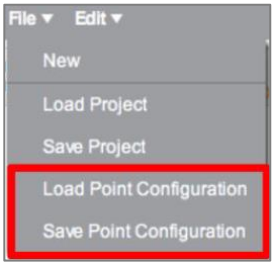
Why Now?

At an unrivaled and enduring pace, computing has transformed the world. The resulting global economic circuitry demands from its operators more than reading, writing, and arithmetic: a fourth foundation, computational thinking. This new baseline need arises in an era of incommensurate education reform focused more on standardized testing than durable learning. Meanwhile, computing's proliferation has empowered students to explore digital worlds interactively, enabling passion-driven discovery that engages the minds that schools too frequently fail to attract or satiate. Educators increasingly acknowledge that game-based learning (GBL) activates student motivation and improves outcomes, but teachers and game designers typically train in divergent dialects.



SAGE transforms Scratch into a gameful intelligent tutoring system. Scratch-embedded Parsons Programming Puzzles and a dynamic scoring system provide students with a continuous stream of feedback and instruction.

SAGE design mode enables teachers to configure points, or load, modify, and save presets

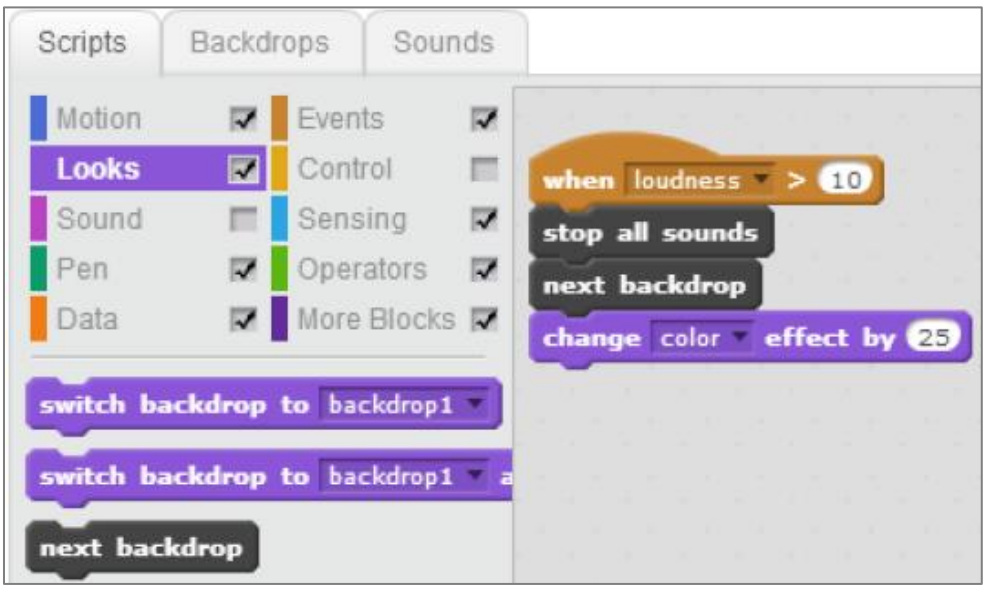


COLUMBIA UNIVERSITY
The Programming Systems Laboratory

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

What's Come Before?

Usually without a gameful approach, researchers have constructed numerous novice learning environments for programming during the last five decades. While most weren't designed with the aim of teaching CT, various CT curricula have leveraged these tools with moderate success. What has often made them effective is their low floors, high ceilings, and wide walls. Low floors allow easy starts. High ceilings offer opportunities to construct increasingly complex projects over time. Wide walls support a diversity of project types so that students with varying interests and learning styles become engaged. To varying degrees, these environments also assist in scaffolding learning, promote transfer to non-novice environments, and deliver toolsets by a systemically equitable and sustainable mechanism, for example, free on the web. However, since the late 1960's, Logo and its numerous counterparts have failed to produce a general populace of computational thinkers. Toy-like programming languages might provide short bursts of enthusiasm, but correct solutions to problems at lower code levels do not necessarily indicate conceptual traces to CT. Students need a CT environment that catalyzes the transfer of knowledge from one concrete instantiation to another, that fosters abstract formulations over a continuum, and that elicits iterative refinement culminating in complex productions.



SAGE provides palette and block restriction systems supportive of learning progressions within constructionist video games

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

Why SAGE?

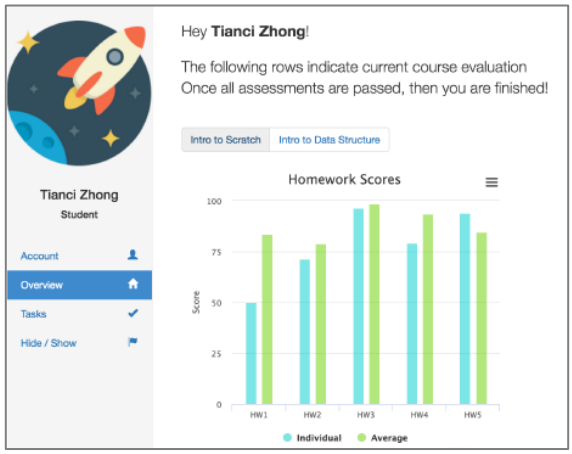
To infuse CT in grade 6-8 curricula, a learning system must simultaneously address the needs of schools, teachers, and students. These requirements are organized along SAGE's axes below. Each axis is additionally associated with a learning principle.

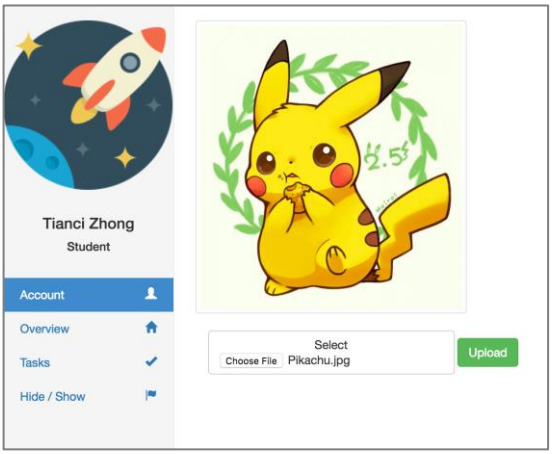
Social. Social learning aligns with the socio-cultural perspective in which individuals collectively construct knowledge though joint activities as they participate in cultural practices that shape cognition. Social interaction within the learning environment thus emerges not only as a motivator for engaging with educational content, but also as a vehicle in which to enact collaborative skills consistent with CT practices. Since modern problem-solving occurs in communities rather than in isolation, developing social agility while encountering CT concepts fosters a positive attitude to "failure" as learning unfolds, and encourages not only experimentation and collective construction of computational models, but also a culture of model critique that questions underlying assumptions and identifies pitfalls.

Nurturing Affinity Space Principle

SAGE affiliates students by enveloping them in shared endeavors that require cross-functional collaboration, thereby creating multiplicities of respected expertise.

SAGE presents a student dashboard that includes a socially motivating badge system, opportunities to compare progress among classmates, and personalized avatars





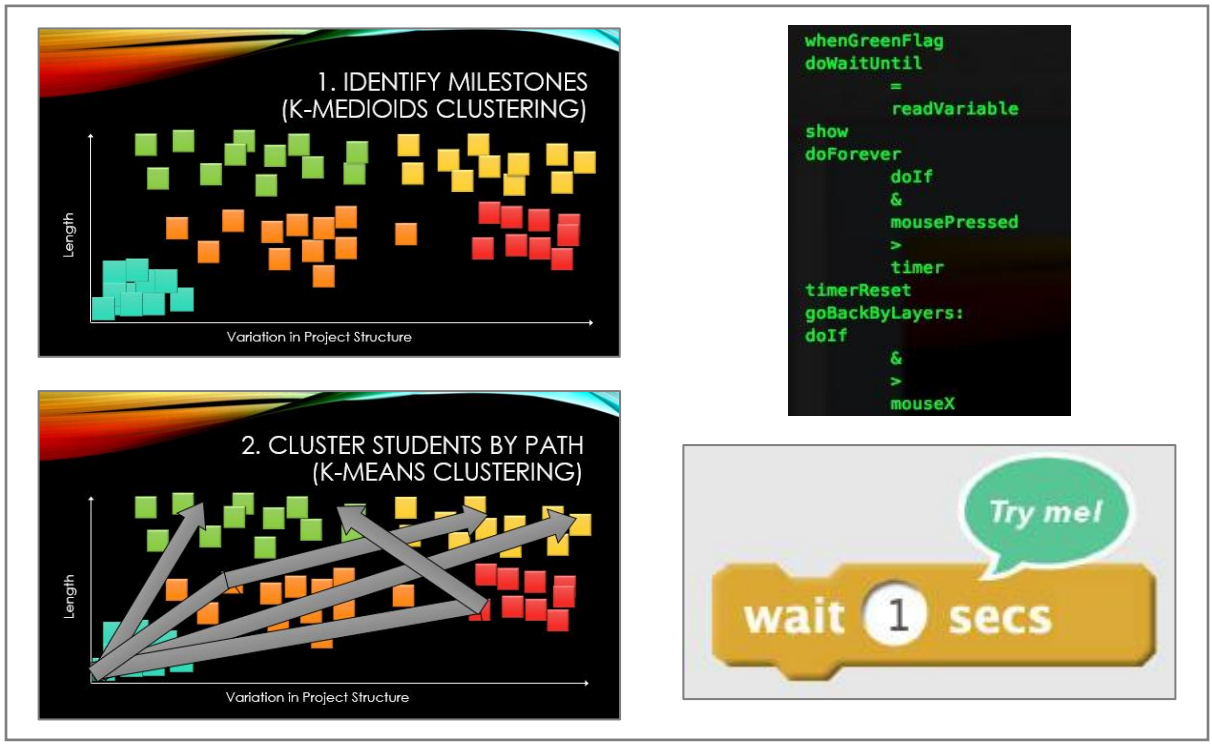
COLUMBIA UNIVERSITY
The Programming Systems Laboratory

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

Addictive. In the video game industry, the term addictive refers to engaged, repeated play, and harmonious rather than obsessive passion. In this context, addicting students to learning is a desirable goal, and the theory of flow might best support addictive learning in a GBL system. Characteristics of flow include: a clear sense of required action; intense concentration; continuous feedback; a sense of control; a capacity to act to overcome challenges; an accelerated distortion of time; and a sense of intrinsic reward. Flow enables students to practice CT with the attentiveness required to develop deep understandings, but studies indicate that fully immersive flow might inhibit metacognition, and therefore immersive flow provides better fit when aiming to proceduralize knowledge than when trying to acquire or reflect on it. Furthermore, since individuals cannot infinitely sustain flow, and students can meet many educational CT goals without achieving such a demanding psychological state, addictive learning must extend beyond flow, and capture longer-term student interest, motivation, and engagement.

Regime of Competence Principle

SAGE induces flow and cultivates engagement by presenting pleasantly frustrating challenges that stretch the competencies of each individual student.

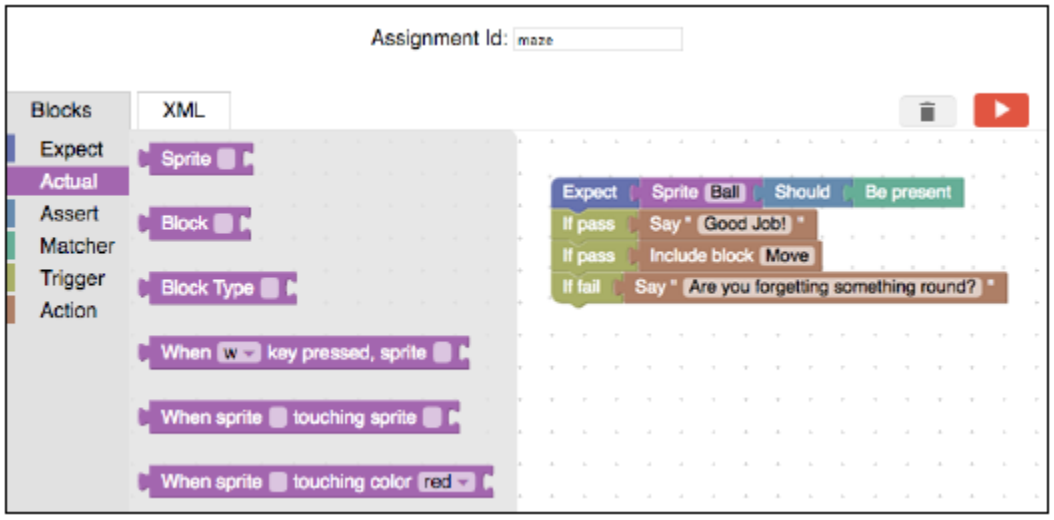


SAGE automatically identifies game milestones and student progress to guide the discovery of solutions

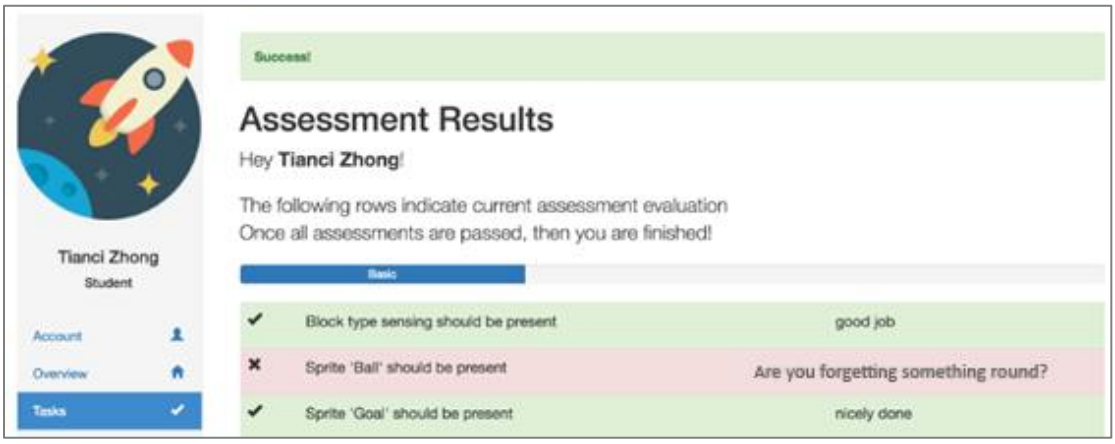
Gameful. When CT concepts, practices, and perspectives meld within game contexts, the learning becomes gameful: goal-oriented, curiosity-driven, failure-fearless, optimistic, and fun. Instead of overlaying the game cycle, CT content blends throughout the environment as an intrinsic component of game-play. Experiential learning theory emphasizes experiencing problem-solving strategies prior to deliberate cognition and reflective behavior during which abstract concepts subsequently develop. Similarly, learning in activity affords groups opportunities to work with authentic objects and technological activity systems that enable close inspections before expanding toward broader goals and generalizations. This active learning is supported by scaffolding, which provides structure by embedding guidance in context so that learners can perform expert-like tasks and develop understandings of CT and the relationship between its objectives and procedures.

Situated Meaning Principle

SAGE does not assemble CT concepts disconnected from gameplay, but instead envelopes students in gameful environments in which concept exploration emerges as narratives unfold.



SAGE offers a visual assessment editor that facilitates the scaffolding of student learning during gameplay

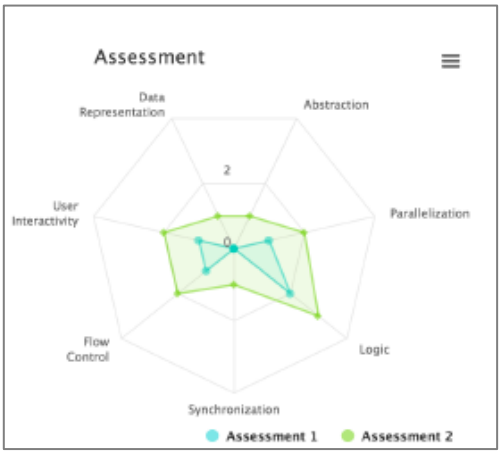


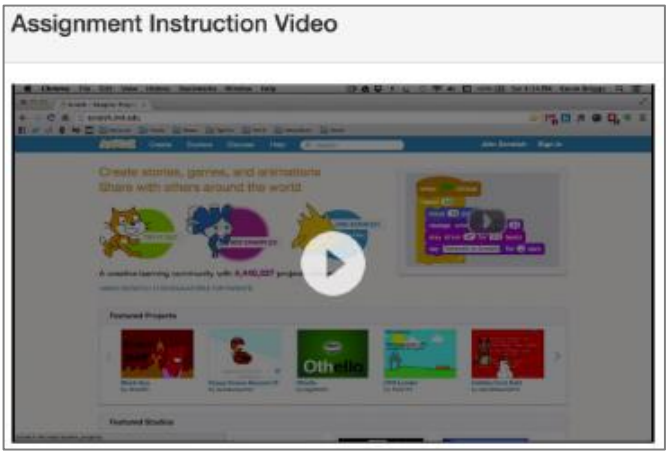
Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu

Engineering. Exploring multiple routes while solving problems engenders engineering habits of mind and mental processes that require important aspects of CT. This engineering problem-solving becomes fun in GBL since a game is, as game designer Jesse Schell reminds us, "a problem-solving activity, approached with a playful attitude." Problem-based learning situates students in real-world contexts so that they can construct principled arguments while collaborating to solve complex, ill-structure problems. This method enhances students' capacities to transfer knowledge to new problems and achieve more coherent understandings than traditional instruction because it consistently activates prior knowledge and demands cumulative reasoning, theory building, and conceptual change.

Explicit Information On-Demand and Just-In-Time Principle

SAGE explains directly only when students eagerly seek or desire information and have accrued the experience and motivation imperative to consume that information lucidly.





SAGE assesses CT concept competency and offers timely direct instruction

For more information and opportunities to become involved, please contact us!

https://github.com/cu-sage/

Social Addictive Gameful Engineering (SAGE)

a collaborative game-based learning and assessment system which infuses computational thinking in grade 6-8 curricula

COLUMBIA UNIVERSITY
The Programming Systems Laboratory

Jeff Bender, Gail Kaiser, Ryan Baker | jeffrey.bender@columbia.edu