CODECAST: An Innovative Technology to Facilitate Teaching and Learning Computer Programming in a C Language Online Course

Rémi Sharrock, Hamonic Ella, Mathias Hiron, Sebastien Carlier

CODECAST: An Innovative Technology to Facilitate Teaching and Learning Computer Programming in a C Language Online Course

Rémi Sharrock
Ella Hamonic
Telecom ParisTech
LTCI, IMT, Paris Saclay
Paris, France
first.last@imt.fr

Mathias Hiron
France-IOI
Paris, France
mathias.hiron@gmail.com

Sebastien Carlier
Epixode
Paris, France
s.carlier@epixode.fr

ABSTRACT
This paper introduces the CODECAST tool: an in-browser C language interpreter, paired with an event and voice recorder and player that facilitates teaching and learning to program by synchronizing audio with source code edition, visualization, step by step execution and testing.

Author Keywords
teaching; programming; code edition; audio; code visualization; code execution; code testing; mooc; online learning.

INTRODUCTION
The CODECAST tool offers an innovative approach to teaching and learning C programming using an online code editor and interpreter that runs in the browser and doesn't require any software installation. The teacher is able to orally explain the entire code creation process while his interaction with a code editor and interpreter is recorded. He can also explain different aspects of the coding process like testing, running, debugging and optimising, with the help of several data and algorithms visualization modules. The learner can play back the explanation and can take control over it anytime he wishes. This means the learner can interact directly with the code and try different ideas he may have while listening to the teacher's explanation: making his own changes to the code, testing with his own inputs, running the code step-by-step to better understand its behaviour, visualizing other parts of the algorithm or the data. CODECAST lets learners go back and forth between the teacher's explanation and their own ideas at any point during the explanation. This facilitates the understanding by easily switching from the teacher's contexts and examples to the learner's ones.

L@S 2017, April 20-21, 2017, Cambridge, MA, USA.
ACM 978-1-4503-4450-0/17/04.
http://dx.doi.org/10.1145/3051457.3053970

CODECAST: MOOC USE CASES
This tool has been created for MOOCs with the help of France-ioi, a french non-profit that offers online contents and tools to teach programming to young students and prepares students for the international Olympiads in Informatics. It has been successfully used with more than 30.000 learners within two MOOCs (Massive Online Open Courses) dedicated to the C Language, available in French on the FUN platform (France Université Numérique) in 2016. Designed for early beginners in computer programming, the *C Language ABC* MOOC made it easier for learners to get onboard with computer programming. It is worth noting that this MOOC had a remarkable completion rate: 16,7% of enrolled learners got an "Honor Certificate". Through a facilitated process of self-appropriation of code uses logic and autonomous testing, the CODECAST tool may be a significant explanatory factor.

CODECAST: THE FEATURES
While developing this tool, the team aimed to combine a "tutorial style video" and an Integrated Development Environment. If online tutorials are traditionally made with screencast technologies, video content has many limitations for the students (e.g.impossibility to copy-paste parts of the code). CODECAST enables online tutorials to be fully interactive.

The CODECAST tool consists of a C language interpreter, integrated with visualization tools, a code editor and an event and voice recorder for the teacher and a player for the learner, all directly accessible on the web within a browser.

The recorder: recording audio and coding events simultaneously.
The recorder is accessible on a single web page that displays, in its most simple configuration, a record/stop button and a source code editor as seen in Figure 1. Other configurations may include a display of some specific programming teaching modules like algorithm and data structure visualizations, a variable inspector, a memory monitor, compilation and step-by-step execution controls, a program input editor and program output display. The default code editor supports source code color syntax

highlighting and is also configurable (text font, color, background, line numbering…)

Figure 1. The CODECAST tool in its most simple configurable interface: a recording button and a code editor area.

Clicking the record button starts the capture of audio as well as keyboard and mouse interactions with the active modules, including key presses, mouse clicks, drag-and-drops and text selection. The teacher is able to explain a complete "coding from scratch" process by starting the recording with a blank source code editor and typing the code one letter at a time. He may also copy/paste some existing code then modify it, and orally explain his changes while he is performing them. Another approach is for the teacher to pre-fill the code before starting the recording, then focus his explanation on the step-by-step execution of this code. Portions of the code may be highlighted to point out some of its aspects, and multiple visualization modules may be activated to help the teacher introduce specific concepts, algorithms or other relevant runtime information.

Once the teacher stops the recording, the recorded files are compressed and uploaded to a cloud service. A unique URL link is then generated for the recording to be shared or integrated in online learning platforms.

The player: playing audio and coding events and interacting with code.

The player is accessed or displayed integrated in a website, using the URL link generated during the recording process. It presents exactly the same graphical user interface as was configured during the recording (enabled modules, colors) except that the record/stop button becomes a play/pause button and a timeline appears right next to the play button. When the play button is pressed, the playback starts and the audio is played in synchronisation with all the keyboard and mouse events. The CODECAST player shows every action performed by the teacher during the recording, and displays the same screen. By dragging the timeline cursor, the learner may rapidly seek a relevant part of the recording; the state of the interface is updated instantaneously while the cursor is moved. The learner may also pause and take control of CODECAST at any time, then modify the code, test it with his own use cases, run it step-by-step, change the inputs or the code itself or use all the modules in the same conditions as the teacher. When the learner clicks the play button again, all the modifications made are discarded and the screen goes back to the state it was in when the pause button was previously clicked.

CODECAST: TECHNICAL DETAILS

The CODECAST tool is developed mostly in Javascript. The code is open source and available on the github platform. It has to be installed on a web server and a cloud service has to be configured to upload the recordings (such as Amazon Web Services). When compiling a C program, a server-side service uses clang to generate an abstract tree, which is then used for the in-browser interpretation of the code. During the recording process, both the audio and the events are stored in the browser memory. The audio is recorded using specific browser audio APIs (Application Programming Interfaces) that interfaces with the system audio and enables to use the computer's external or integrated microphones. When the recording is finished, the audio is compressed to the MP3 format inside the browser, and uploaded to a server. All the events are stored using a JSON based format [1] and uploaded to a server. The player only needs a unique identifier to download the MP3 and the corresponding JSON file from the server before playing then in sync.

CODECAST: THE DEMO

The audience will be able to test the CODECAST tool, directly on their own computer, loading both the recorder and the player within a browser. To test the recorder, the computer has to have a microphone. Many existing "CODECASTS" were created for MOOCs and will be directly accessible from a page of collected URLs. A website will be available to centralize the demonstrations.

ACKNOWLEDGMENTS

We thank France-ioi and Epixode for helping us designing and developing the CODECAST tool. We acknowledge the Patrick and Linda Drahi foundation for providing the funding.

REFERENCES

1. France-IOI website Retrieved January, 2, 2017 from http://www.france-ioi.org/