Ancient Brain: A JavaScript coding platform for education with 3D graphics, Websockets, AI and support for teachers

Mark Humphrys
School of Computing, Dublin City University
Ireland
mark@ancientbrain.com

Abstract

This paper introduces a JavaScript coding site called "Ancient Brain" (at https://ancientbrain.com/) , which is designed for education with support for students and teachers. It has an extensive list of features to support coding almost anything that can be coded in JavaScript, including HTML-based pages, 2D graphics worlds, 3D graphics worlds, Internet-enabled Websockets apps, and AI apps. All are coded and run on the site in the browser with no install. Support for teaching includes "teacher" and "student" accounts where student code is hidden from other students but not from the teacher, who can run and even edit the code written by all their students. This site has been tested with several years of undergraduate and taught postgraduate students, though not written up in a paper until now. At time of writing there are 9,323 JavaScript creations on the site. We survey some of the extraordinary range of programs on the site, written by, at last count, 2,150 coders. The next step is to take the site into secondary schools. We will discuss how this could be done, with examples.

CCS Concepts

• **Computer science education, Software engineering education, K-12 education**;

Keywords

Coding sites, Education, JavaScript, WebGL, Websockets, AI in education

1 Introduction

The space of text-based programming (as opposed to visual programming) in the classroom is an interesting space. It seems clear that older secondary school students are ready to move beyond visual programming environments like Scratch and Snap! and try coding in regular text-based programming languages like Python and JavaScript. But how should they code? On what platforms? What can their code do? And how can teachers safely run untrusted student code to mark it?

A wide range of tools and technologies are in use in coding education at school (K-12) and university level. See [1] for a 2019 survey of programming technologies in K-12 worldwide. In their survey, visual programming dominates. The text programming languages they found in use (restricting to high school level) were Java, Python, Pascal and Smalltalk. JavaScript does not appear at all, except in one course which allowed students using the Snap! visual programming language to look at the JavaScript that it translates to underneath. Despite the survey, there are in fact a number of projects to teach JavaScript to students and we will look at them below.

We will first explain the logic that led us to choose JavaScript as a language for a shared coding environment for education. The proof of this choice will come when we explain some of the things students have created on this platform.

2 Principles for a shared coding environment

This work evolved out of an earlier project called the "World Wide Mind" [2] [3] [4] [5]. The initial focus of the World Wide Mind project was not on education but on AI research. The idea was to make a platform that allowed the building of "hybrid AI" systems from the work of multiple authors. Similar to "ensemble AI" or "mixture of experts" models, with the focus on bringing together the work of widely dispersed authors. The name "World Wide Mind" indicates that parts of the AI could exist at different servers around the world.

As the project developed, and as we built a platform for multiple authors to share and run code, it was soon realised that the platform could be used for teaching.

The following principles for a shared coding environment emerged from the World Wide Mind project:

1. Users (e.g. researchers in AI, or students in an educational setting) should be able to view collections of programs written by other users (e.g. researchers, teachers, 3rd parties, and possibly other students). Normal programs are public.
2. Users can run these programs without any install needed.
3. Users can run them securely, even though they are untrusted code (e.g. untrusted student code).
4. Users can edit their own programs without any install needed. They are saved for later use, and use by others.
5. For normal programs, other users can view the source code of the program.

6. For normal programs, other users can clone the program to get their own copy to edit.
7. The programs can do advanced graphics.
8. The system should scale to large numbers of users.

3 JavaScript solution

After some false starts experimenting with other technologies (notably Java), we eventually chose a JavaScript language based solution to implement these principles. The programs are stored on a server and run on the client side (in the browser). We address the principles as follows:

1. The programs will be stored on a server, where users may browse lists of what programs exist. Programs on the site are by default public. We have an option to set them to hidden (not discoverable).
2. Users can run the JavaScript without needing any software other than a browser.
3. Users can run the JavaScript securely, because the JavaScript security model is mature and stable [6]. (It is of course fundamental to the Web.)
4. Users can edit inside the browser using an editor JavaScript library, with no install of any software needed. The code is saved to their account for later use and (normally) use by others.
5. Users can "view source" to view any JavaScript delivered to the client side. This is part of the Web model. We will allow a program to opt out of this by being hidden (not discoverable). We will allow a program to partially opt out of this by having its JavaScript obfuscated.
6. Users can clone the programs if they register an account. They cannot edit other users' creations, but need to make their own copy to edit.
7. Since the emergence of WebGL, and JavaScript graphics libraries based on it such as Three.js [7] and P5 [8], JavaScript can do advanced 3D and 2D graphics, all on the client side.
8. Code runs on the client side, not the server side, so the system scales well to large numbers of users and runs.

4 Ancient Brain

Our JavaScript coding site is called "Ancient Brain", and is live at https://ancientbrain.com/. See Figure 1. The reader is strongly urged to simply go there now and start running programs (which are called "Worlds"). You run Worlds by clicking on them in the browser, with no install needed. Worlds can be games, AI applications, general apps, coding challenges, maths demos, physics demos, quizzes, chat applications, virtual worlds, museum tours, and more.

Ancient Brain is in use in multiple teaching courses at Dublin City University, at both undergraduate and taught postgraduate level. At time of writing, it hosts 9,323 JavaScript Worlds written by 2,150 coders. Experiments have been run on bringing it into schools, and writing a course or textbook for schools is at the top of future work. We are seeking partners to work on this.

Figure 1 shows part of the home page of ancientbrain.com as at Feb 2024. This showcases a random four programs (called "Worlds"). Each can be run by clicking on it, with no install. The ones here

are: (1) One Cube World, an extremely simple World to draw a cube, suitable for use on day one of learning to code, (2) Landmark Museum, a World that calls OpenAI's GPT API and also Google Image Search API, and displays returned text and images on the walls of a 3D graphics scene, (3) Recognise any image, an AI World to recognise images using ML5 and MobileNet, and: (4) Game of Crowns, an advanced World with 3D graphics and 3D models. To find and run all these Worlds, go to the site and use the search box. At time of writing, there are 9,323 Worlds on the site.

5 Ancient Brain features

Ancient Brain now has many features, including:

1. Edit code in the browser, with no install, using the Ace JavaScript editor [9]. Editor syntax highlighting and syntax error checking. There is intelligent code help. You can select code and the editor will find a manual page / help page for that code.
2. On editor "Save", the code is live and can be run. No "build" or "publish" or other intermediate step. Runs can be started with a button in the editor. Or, since a run is a webpage, you can leave the run webpage up and just hit "Reload" to load in the new code.
3. An option to obfuscate your JavaScript. If chosen, you can see the plain text of your JavaScript, but other users (when they "view source") only see an obfuscated version. The site provides a further option to make the code hidden URL where other users cannot find it or run it. In an educational setting, we encourage students to choose "hidden URL" during the project, so no other student can see their work. After the project is over, the work can be made public.
4. Support for 2D and 3D graphics. Built-in graphics libraries and many sample Worlds with open source code. Support for Three.js, P5, Phaser and native WebGL coding. Other graphics libraries can be uploaded by users. For an overview of what graphics are possible, see "Editor's Choice Worlds" in the main menu of the site. We also allow the making of text-only Worlds, that are regular web pages with HTML elements.
5. Users can upload images, textures, music, sound effects, JavaScript libraries, JSON data files and 3D models to use in their Worlds. So students can customise their Worlds with their own images and music. Uploads are public. You can reference and use other users' uploads in your code.
6. Mouse re-definition support. Support for re-defining what mouse drag and click and scroll do. For example, mouse drag to move objects or drag the camera in a 3D world. Sample Worlds are provided. See "Mouse" in the Ancient Brain docs [10].
7. Worlds run on mobile. We imagine most editing will be done on desktop, but the site works on mobile, and Worlds can run on mobile and work through touch. From the educational viewpoint, this means students can develop Worlds on desktop machines in class and then show them off to parents and friends later on their mobile.
8. Touch re-definition support. Support for re-defining what touch drag and tap and pinch do. For example, touch drag
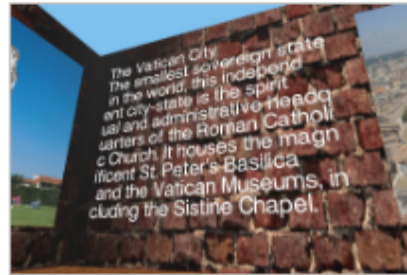
Ancient Brain
Code. Play. Teach. Learn. Research. Demo.

An online coding and hosting platform for everyone.

34 countries   **2,150 coders**   **9,318 worlds**   505,369 runs
3.2 million lines of code   **Update stats**
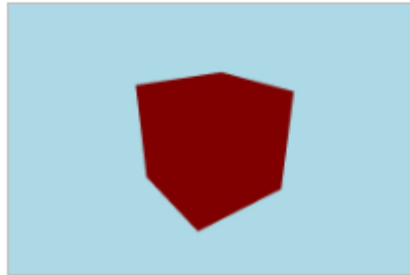
One Cube World (...
1112 runs ♦ 2 likes
By **Starter user**
Created: 11 Apr 2018
Modified: 17 Sep 2023
Simple starter World (Three.js version). Built-in camera control.

Landmark Museum
115 runs ♦ 0 likes
By **Christopher Dobey**
Created: 4 Dec 2023
Modified: 4 Feb 2024
A walkthrough tour of the most significant landmarks in any given country. Three.js 3D World wit...
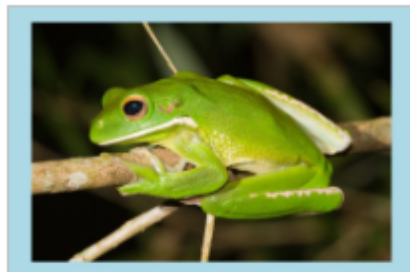
Recognise any image
267 runs ♦ 0 likes
By **Starter user**
Created: 10 Jun 2021
Modified: 17 Sep 2023
*Try to recognise any image with ML5 and MobileNet. Enter any image URL at runtime.*
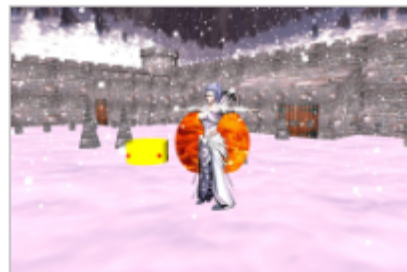
Game of Crowns s...
209 runs ♦ 0 likes
By **Enhanced**
Created: 21 Jun 2018
Modified: 17 Sep 2023
Enhanced version of "Game of Crowns" by Niamh Byrne with better graphics and controls

Figure 1: Part of the home page of ancientbrain.com as at Feb 2024.

to move objects or drag the camera in a 3D world. Sample Worlds are provided. See "Touch" in the Ancient Brain docs [10].

9. Support for 3D models. Built-in libraries for embedding 3D models (e.g. houses, people) in Worlds. Sample Worlds are provided. See "3D models" in the Ancient Brain docs [10].
10. Support for physics in 3D Worlds. Built-in physics libraries with gravity, momentum, etc. For example, ammo.js [11].
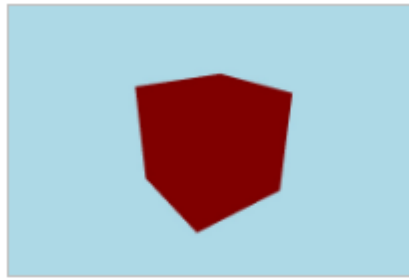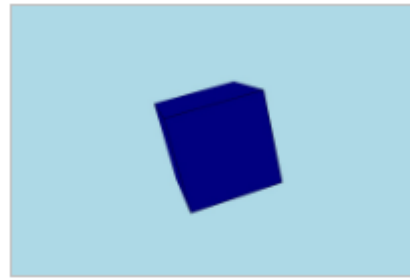
One Cube World (...
1112 runs ♦ 2 likes
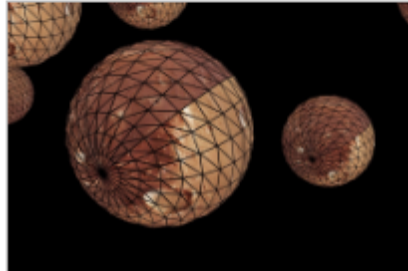By **Starter user** 🇮🇪
Created: 11 Apr 2018
Modified: 17 Sep 2023
Simple starter World (Three.js version). Built-in camera control.

One Cube World (P5)
1845 runs ♦ 1 like
By **Starter user** 🇮🇪
Created: 10 Nov 2018
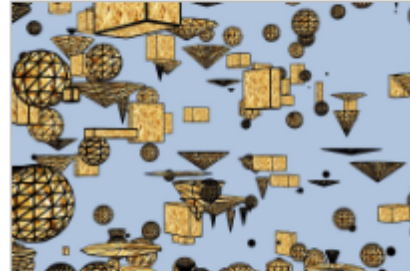Modified: 17 Sep 2023
Simple starter World (P5 version). No camera control. So cube rotates to show 3D.

My first world
412 runs ♦ 0 likes
By **Sagnik Chakraborty**
Created: 19 Sep 2022
Modified: 17 Sep 2023
hopefully will be wonderful, but its scary for now lol

Spaghett
321 runs ♦ 3 likes
By **Theo Delettre** 🇫🇷
Created: 30 Sep 2021
Modified: 17 Sep 2023
Clone of "One Cube World (P5)" by Starter user

Figure 2: The two versions of "One Cube World", plus two variants that were quickly made by students once they saw how the site worked.

11. Support for AI. Support for the ML5 AI library [12]. You can upload other AI JS libraries and use them.
12. Websockets support for all Worlds. This allows real-time communication between users running your World. Users can write multi-user Web games and have real-time chat between users of the World. See "Websockets" in the Ancient Brain docs [10].
13. Almost any JS can be injected into the run window. JS can include other JS at run-time (either local JS or remote JS). JS that makes server calls may not be portable, but almost any other JS should be portable to the site. There are projects on the site to port all three.js examples [13] and all P5 editor built-in sketches [14].

14. Search engine for Worlds, users and uploads. Code search to search the JS code of all public programs.
15. Can embed Worlds in external third-party sites. Like YouTube embeds.

5.1 Ancient Brain for teachers

Ancient Brain has some built-in support for teachers. See "Teaching" and "Register a class" in the main menu. When you register a class, you get a special "Teacher" account and a number (you decide) of "Student" accounts which you will hand out to your students. These have certain properties:

```
 1
 2   const objectsize    = 200;       // size of object
 3
 4   const anglechange   = 0.01;      // how much the rotate angle changes eac
 5
 6   var angle = 0;                   // rotate angle starts at 0
 7
 8
 9   function setup()        // "setup" is called once at start of run
10 ▾ {
11     createCanvas ( ABWorld.fullwidth(), ABWorld.fullheight(),  WEBGL );
12   }
13
14   function draw()         // "draw" is called every timestep during run
15 ▾ {
16       background("lightblue");    // background color
17       fill("navy");               // paint box with this color
18
19       rotateX(angle);             // set each dimension rotation angle to
20       rotateY(angle);
21       rotateZ(angle);
22
23       box(objectsize);            // draw a cube of this size
24
25       angle = angle + anglechange ;       // change angle each step to get
26   }
27
```

Figure 3: The code for the P5 version of "One Cube World", shown here in the Ancient Brain editor.

- In a "Student" account, all code is obfuscated and at hidden URLs. Other students cannot see it, find it or run it. It is excluded from search and the URL is not guessable.
- In a "Teacher" account, you can find and see and run all code written by your students. You can view the plain text source code. And you can edit the code. It is like having "root access" to their accounts.

Running and marking student programs have some powerful features. First note that students do not email you their programs or put them into a file space that you have to manage. Students upload them to the server and you as the teacher do not have to manage any of that. Second, you can safely run untrusted student code. A recurring issue with teaching coding is how can the teacher run the code safely [15]. What if the code tries to access files or cause damage? The JavaScript model means that, even though the teacher is running untrusted code, it cannot damage their machine or browser or account.

5.2 Teachers edit student code

In addition to safely running the submitted student code, as the teacher you can edit and re-run any student code. You can make small (or large) changes to student code to see what happens. Test if it is robust with different parameters. You can even fix bugs and run it again.

This is one of those features you don't know you are missing until you have it. The author of this paper marked student projects for many years but never edited or bug-fixed them, until this system made it easy. The most useful edit is probably just to insert console.log statements at various points to see what is going on. Other useful small edits have included fixing resources and dependencies that existed when students submitted but later vanished. When marking code, it is rare enough for teachers to edit a student program and run it again. The reason being that it is usually not safe or not practical. Here it is both safe and practical, and even pleasurable.

6   Starter tutorial: One Cube World

We will now look at some of the 9,323 Worlds that exist on the site, to illustrate what can be written.

The site has two "starter tutorials" for the beginner user, one using P5 [16] and one using Three.js [17]. This is where to get started coding on the site. Each tutorial has a World to draw a cube in 3 dimensions. The P5 version of "One Cube World" has 13 lines of code. The Three.js version of "One Cube World" has 11 lines of code. (But arguably Three.js is more complex for the beginner coder.) Note in Figure 2 how the students quickly learned to make multiple objects, of multiple shapes, and put textures on the objects. To find these Worlds, use the search box on the site.
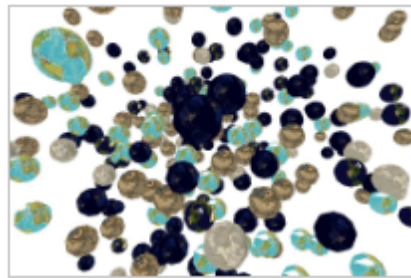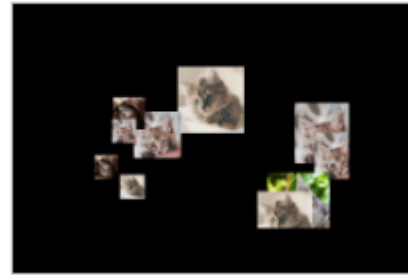
Blank Three.js World
1497 runs ♦ 2 likes
By **Starter user** 🇮🇪
Created: 20 Nov 2016
Modified: 17 Sep 2023
A simple starter World. An Array of spheres. Painted with textures. Random motion.

Tutorial 7.8
107 runs ♦ 1 like
By **"Coding Train" p...** 🇺🇸
Created: 5 Sep 2018
Modified: 17 Sep 2023
Objects and Images. On Ancient Brain you can upload your own images. Try it!

Simple World
1643 runs ♦ 2 likes
By **Starter user** 🇮🇪
Created: 1 Oct 2016
Modified: 17 Sep 2023
Simple World with a Mind-controlled agent, randomly-moving enemy, paint blocks with texture.

Flickr World
210 runs ♦ 0 likes
By **Starter user** 🇮🇪
Created: 12 Aug 2018
Modified: 17 Sep 2023
Demo of "Web page" API. Just use JS to make a web page. This page gets images from Flickr using ...

Figure 4: Some of the Starter Worlds.

The code for the P5 version of "One Cube World" is shown in Figure 3. A "dark mode" can also be selected. The P5 Starter Tutorial introduces novice coders to "One Cube World" and gets them to clone and edit the code. We start them with editing simple things like object size and colour. By the end of the tutorial we show them how to upload their own image to the site, paint it on the cube, and make multiple cubes. Of course, for more advanced coders, Worlds can get a lot bigger than this. At time of writing, the largest World created on the site has 7081 lines.

In contrast to some coding sites, our code editor has only a JS pane, and no HTML or CSS pane. Default HTML and CSS is taken care of. Any custom HTML and CSS can be done from within JS, and we provide tools and examples for that. This is a pure JS environment.

The tutorials give the new coder a series of exercises to clone and edit the code of "One Cube World". These tutorials have been used many times with success with students who have never coded in any language before. They have been tested with secondary school students and even primary school students, who learnt some real JavaScript in an hour with them.

7 Some collections of worlds

7.1 Starter Worlds

The site provides a large number of readable, open source, "Starter Worlds" to show how to code different types of World. See "Starter Worlds" in the main menu. Some examples are in Figure 4. Shown here are Starter Worlds for: (1) an ultra-simple Three.js graphics

Game of Crowns
2255 runs ♦ 2 likes
By **Niamh Byrne**
Created: 16 Nov 2016
Modified: 17 Sep 2023

Find Woody's Hat
588 runs ♦ 1 like
By **Benjamin Olojo**
Created: 4 Dec 2022
Modified: 19 Oct 2023
Clone of "Capture the Hat" by Tuomas Bazzan

escape...
800 runs ♦ 2 likes
By **Jon Bryan Ortiz**
Created: 22 Nov 2022
Modified: 17 Sep 2023
There are 10 pages scattered around the map. First person to find all 10 pages escapes...the res...

Chess
5850 runs ♦ 2 likes
By **Eoin Daly**
Created: 19 Oct 2022
Modified: 17 Sep 2023
A game of chess by the beach, on a beautiful day

**Figure 5: Some Worlds using 3D models on the site.**

World, (2) how to make a P5 graphics World, (3) a slightly more complex Three.js graphics World, and: (4) how to make a web page of HTML elements. To find these Worlds, go to "Starter Worlds" in the main menu.

Most of the Starter Worlds are designed so that the easiest parts of the code to change are in a section called the "tweaker's box" at the top. Essentially, changes that can be made without being a skilled programmer are in the "tweaker's box" (such as changing the values of some basic parameters) whereas the more difficult to approach code is lower down. So we encourage new users to clone the Starter Worlds and change things in the "tweaker's box" and see what happens.

7.2 Inserting 3D models

Pre-built 3D models (with pre-built covering materials) may be uploaded to Ancient Brain and inserted into 3D graphics Worlds. There are many sources of free and paid 3D models and materials, such as free3d.com and sketchfab.com. We provide a number of "Starter Worlds" with 3D models to show how to insert these kind of models into JavaScript Worlds on the site. The simplest is probably "Castle World" on that page, which uses a pre-built 3D model of a castle found at [18].

Mostly the 3D models on Ancient Brain are included into Three.js based Worlds. The JavaScript to include various types of 3D models into Three.js scenes is found in the "loaders" files in the three.js repository [19]. It is our experience that the help on how to insert models in JavaScript Worlds is scattered and thin, so the more open source examples we can get on the site the better. The Three.js
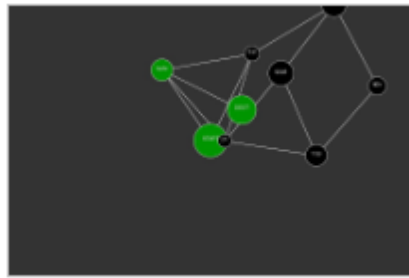
Breadth-first se...
2774 runs ♦ 0 likes
By **"Coding Train" p...** 🇺🇸
Created: 30 Apr 2019
Modified: 17 Sep 2023
Breadth-first search with a Force-directed graph.

GA (Finnegans Wake)
994 runs ♦ 0 likes
By **"Coding Train" p...** 🇺🇸
Created: 2 May 2019
Modified: 17 Sep 2023
Genetic Algorithm to write the first line of Finnegans Wake.

Character recogn...
1795 runs ♦ 0 likes
By **"Coding Train" p...** 🇺🇸
Created: 5 Jul 2019
Modified: 17 Sep 2023
Neural network to do character recognition from image database and from handwriting in real-time.
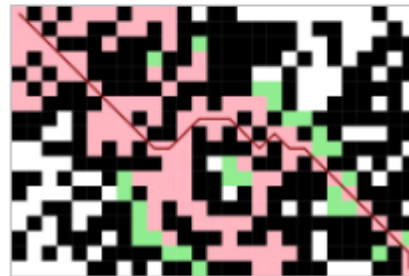
A star
3941 runs ♦ 1 like
By **"Coding Train" p...** 🇺🇸
Created: 1 May 2019
Modified: 17 Sep 2023
"A Star" algorithm to find shortest path through maze.

Figure 6: Some of the Worlds used in the AI course on Ancient Brain.

site has many impressive examples [20] but they tend not to be explained to the learner.

Users of Ancient Brain have uploaded many further examples. See Figure 5. All of the Worlds in this figure were created by students. "Game of Crowns" is based on the Starter World "Castle World", with the addition of a new model of a queen. Plus the student made it snow. "escape" is an extraordinary World with a 3D model of an entire city that you move inside of. "Chess" is a full 3D game of chess with rotating camera. To find these Worlds, use the search box on the site.

7.3 Showcases of student code

Student Worlds (and all Worlds) on the site may be public for the world to see and run (and also clone and edit). This is optional. Worlds may be set to hidden to opt out of this. At Dublin City University, we have public "Showcase Worlds" pages of the best student Worlds [21], with student consent of course. Students can opt out, but they generally love this feature.

As teachers well know, student projects tend to run once, be marked, and never be run again. Even if stored somewhere, they tend to have dependencies (like libraries and resources) that mean they will not work in the future. A video might be taken, but the running project is not seen again. Ancient Brain is committed to keeping old Worlds and dependencies intact so they will run indefinitely. This should be possible if dependencies are uploaded to Ancient Brain. If remote dependencies are used, this plan will not work.

At time of writing, the Showcase includes many advanced games and 3D environments written by students. It also includes many AI Worlds, including the following:
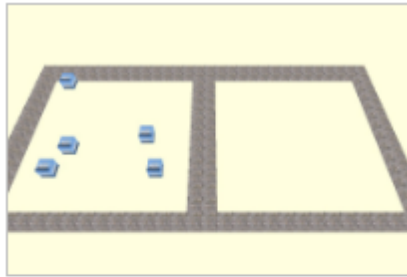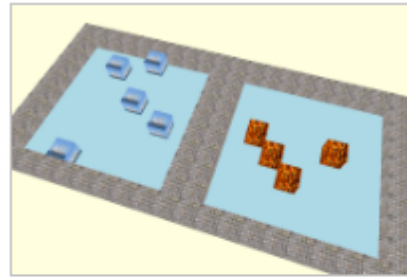
Ch.4. Two grids.
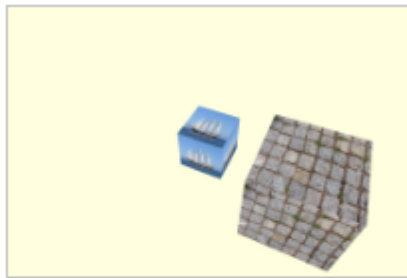71 runs ♦ 0 likes
Created: 9 Jul 2023
Modified: 24 Aug 2023
Two grids. My grid of ships, and my estimate of opponent's grid.

Ch.5. Click on l...
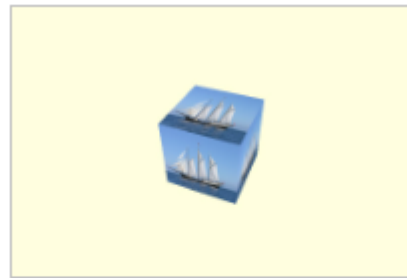51 runs ♦ 0 likes
Created: 9 Jul 2023
Modified: 24 Aug 2023
Click on location to attempt to sink a ship. Click in 2D screen space selects a location in 3D T...

Ch.2. Draw two b...
26 runs ♦ 0 likes
Created: 8 Jul 2023
Modified: 24 Aug 2023
Draw two boxes. Two textures. The "if" statement.

Ch.1. Draw a box.
59 runs ♦ 0 likes
Created: 7 Jul 2023
Modified: 24 Aug 2023
Draw a box in Three.js space. Click to drag camera.

Figure 7: Some of the Worlds already coded for each chapter of a textbook.

- A series of Worlds to do character recognition from images with neural networks, written by taught postgraduates.
- A series of Worlds to implement the A-star pathfinding algorithm in a 3D graphics environment.
- A series of Worlds that call remote AI APIs. These call a range of amazing AI APIs, including chatbots like GPT, image recognisers, image generators, text translation programs, and more. These AI API Worlds are detailed in a separate paper [22].

8 Teaching courses

8.1 AI course

Ancient Brain has been tested for several cycles now in the delivery of both online and in-person classes for undergraduates and taught postgraduates. In particular, a series of AI Worlds have been written to base AI courses around. These are visible at [23]. Shown in Figure 6 are JavaScript Worlds for breadth-first search, genetic algorithms ,

neural networks for character recognition, and A-star search. Each comes with student exercises.

These Worlds give an introduction to AI concepts, including A-star search, genetic algorithms, neural networks, backprop and image recognition. These are all JavaScript Worlds on Ancient Brain that can be run, cloned and edited. We have made student exercises for each. Many of the exercises follow the format: "clone" the World, make some edit (like change learning rate, population size, number of hidden nodes), run it again, and then try to explain what it does. Students are given working AI algorithms and can immediately make changes to explore around the edges of them and test their limits. All in the browser with no install and no risk.

8.2 Intro to Coding course

AI courses are for advanced coders. We believe a real future of this site is in "Introduction to Programming" courses for new coders,

probably in secondary school. We have such a course in experimental form. This is a course in how to code, using JavaScript and P5, based on the video series "Code! Programming with p5.js" by Daniel Shiffman [24]. The Shiffman code has been ported (with permission under MIT licence) to Ancient Brain "Worlds" that can be run, cloned and edited on Ancient Brain in the browser. See "Coding course" in the main menu. It is suitable for people who have never programmed before.

We have brought Ancient Brain into some secondary schools for experimental runs. We were able to teach some real JavaScript code to complete non-coders within one session. We even managed to teach some primary school children. The children's ability to customise and save and "own" their Worlds was motivating for many of them. What is clearly needed next is to make a course for secondary school students.

9 Related work

As noted above, JavaScript barely appeared in a 2019 survey of coding languages used in high schools [1]. However, there are many projects to teach JavaScript to young people. See [25] for a 2019 survey of JavaScript teaching environments. It compares a number of "code playgrounds" for teaching purposes. It proposes an environment called LearnJS, though this seems to be a local install and not a public server.

There are a number of JavaScript "sandboxes" and online tutorials such as at W3Schools [26]. These lack many of the features of Ancient Brain, notably in terms of user ability to own creations and uploaded files and share them with others publicly. See [27] for a JavaScript course for high school that uses an online editor at tutorialspoint.com [28].

Daniel Shiffman's site, The Coding Train [29], has extensive exercises and examples and puzzles for young people in JavaScript. He mostly directs them to P5 Editor [30] for their coding. P5 Editor works well, but Ancient Brain has many features that P5 Editor lacks, notably: World images, Websockets server, hide and obfuscate code, support for "teacher" and "student" accounts, search JS code of all Worlds, and support for Three.js and other libraries.

There are a number of JavaScript for kids books [31], [32], [33]. These struggle with what hosting site, if any, to use. Some use a localhost web server. Some just use local files and double-click on them. [32] has a simple online editor. If localhost or local files are used, how does a teacher see, run and mark student code? Ancient Brain solves this problem.

Replit Teams for Education [34] is closer to our work, though you cannot run user creations without logging in and forking them. CodePen [35] is also similar, though its default is a four-pane page: HTML, CSS, JS and Run panes. See [36] for use of CodePen in schools education.

10 Future work: Coding book for schools

We are actively seeking a partner to work on an introduction to coding book for secondary schools, using JavaScript on Ancient Brain. The Shiffman exercises above do not make use of Ancient Brain's unique functionality, so we are developing our own exercises. See the "Book chapters" account on the site [37].

Figure 7 shows draft Worlds built for these book chapters. This illustrates the kind of ambitious program we believe can be built in stages by new coders on Ancient Brain. A series of eight chapters leads to building a "Battleship" style Websocket game that can be played by two users over the Internet. It has 3D graphics with customisable textures, and mouse control to rotate and zoom camera. Each chapter builds on the previous, in introducing new concepts in coding. View the code (online) to see exercises for each chapter. We are seeking partners to help turn this into a book.

Ancient Brain has been tested in several cycles of teaching, but not written up in a paper until now. We believe that its features can now enable a new type of coding course for young people, introducing them to real text-based code, and yet allowing access to the advanced features they are used to in their regular computing life: 3D graphics, professional 3D models, music and sound effects, real-time networking and chat, and even AI. The aim of the site, fairly ambitiously, is both to make coding exciting for students, and to make running and marking a pleasure for teachers.

References

[1] Claudia Szabo *et al.* 2019. "Fifteen Years of Introductory Programming in Schools: A Global Overview of K-12 Initiatives", Koli Calling '19: Proceedings of the 19th Koli Calling International Conference on Computing Education Research, November 2019, Article No.8.

[2] Mark Humphrys. 2001. The World-Wide-Mind: Draft Proposal, Dublin City University, School of Computing, Technical Report no. CA-0301, Feb 2001.

[3] Ciarán O'Leary, Mark Humphrys and Ray Walshe. 2004. Constructing an animat mind using 505 sub-minds from 234 different authors, Proc. 8th Int. Conf. on Simulation of Adaptive Behavior (SAB-04), July 2004, Los Angeles, CA.

[4] John Pendlebury, Mark Humphrys and Ray Walshe. 2012. An Experimental System for Real-time Interaction Between Humans and Hybrid AI Agents, 6th IEEE International Conference on Intelligent Systems (IS'12), Sofia, Bulgaria, September 6-8, 2012.

[5] Oisín Mac Fhearaí. 2014. Massively multi-author hybrid artificial intelligence, PhD thesis, Dublin City University.

[6] Markus Vervier *et al.* 2017. Browser Security WhitePaper, X41 D-SEC GmbH, Aachen, Germany, 19 September 2017.

[7] Three.js JavaScript graphics library https://threejs.org [Retrieved 24 Feb 2024.]

[8] P5 JavaScript graphics library, https://p5js.org [Retrieved 24 Feb 2024.]

[9] Ace editor. https://ace.c9.io [Retrieved 24 Feb 2024.]

[10] Ancient Brain docs. https://ancientbrain.com/docs.php [Retrieved 24 Feb 2024.]

[11] ammo.js physics library. https://github.com/kripken/ammo.js [Retrieved 24 Feb 2024.]

[12] ML5 AI library. https://github.com/ml5js/ml5-library [Retrieved 24 Feb 2024.]

[13] threejs.org porting project. https://ancientbrain.com/user.php?userid=threejs [Retrieved 29 Feb 2024.]

[14] P5 Editor porting project. https://ancientbrain.com/user.php?userid=p5editor [Retrieved 29 Feb 2024.]

[15] Ryan W. Sims. 2012. "Secure Execution of Student Code", Technical report of Department of Computer Science, University of Maryland (15 May 2012).

[16] Starter Tutorial (P5). https://ancientbrain.com/p5.start.php [Retrieved 1 Mar 2024.]

[17] Starter Tutorial (Three.js). https://ancientbrain.com/three.start.php [Retrieved 1 Mar 2024.]

[18] "WebGL With Three.js - Lesson 6". Script Tutorials. https://script-tutorials.com/webgl-with-three-js-lesson-6 [Retrieved 1 Mar 2024.]

[19] Three.js loaders for 3D models. https://github.com/mrdoob/three.js/tree/master/examples/jsm/loaders [Retrieved 1 Mar 2024.]

[20] Three.js examples. https://threejs.org/examples [Retrieved 2 Mar 2024.]

[21] Showcase Worlds. Ancient Brain. https://ancientbrain.com/showcase.php [Retrieved 25 Feb 2024.]

[22] Mark Humphrys. 2024. "Bringing AI APIs into the classroom with a JavaScript coding site", to appear in 5th International Conference on Education and Artificial Intelligence Technologies (EAIT 2024), Oct 16, 2024 - Oct 18, 2024, London.

[23] AI programming exercises. https://ancientbrain.com/course.ai.php [Retrieved 26 Feb 2024.]

[24] "Code! Programming with p5.js" , The Coding Train. https://www.youtube.com/@TheCodingTrain [Retrieved 2 Mar 2024.]

[25] Ricardo Queiros. 2019. "Learning JavaScript in a Local Playground", 8th Symposium on Languages, Applications and Technologies (SLATE 2019).

[26] JavaScript tutorial at w3schools.com. https://www.w3schools.com/js/default.asp [Retrieved 5 Mar 2024.]

[27] Ching-Yu Huang and Mayra Bachrach. 2018. "A JavaScript Curriculum for High School Students to Explore Computer Science", *International Journal of Information and Education Technology*, Vol. 8, no. 12 (2018).

[28] Online Javascript Editor, at tutorialspoint.com. https://www.tutorialspoint.com/online_javascript_editor.php [Retrieved 5 Mar 2024.]

[29] The Coding Train. thecodingtrain.com [Retrieved 2 Mar 2024.]

[30] P5 editor. editor.p5js.org [Retrieved 26 Feb 2024.]

[31] Syed Omar Faruk Towaha. 2016. *JavaScript Projects for Kids*, Packt Publishing.

[32] Chris Strom. 2018. *3D Game Programming for Kids*, 2nd edn, O'Reilly.

[33] Siddharth Dalal. 2019. *Elementary JavaScript*, lulu.com.

[34] Replit Teams for Education. https://docs.replit.com/category/teams-for-education [Retrieved 10 Mar 2024.]

[35] CodePen. https://codepen.io/ [Retrieved 8 Mar 2024.]

[36] CodePen Education. https://codepen.io/features/education [Retrieved 10 Mar 2024.]

[37] "Book chapters". https://ancientbrain.com/user.php?userid=chapters [Retrieved 2 Mar 2024.]