CONSTRUIT 2017
International Conference
MAKING, THINKING AND LEARNING
IN THE DIGITAL AGE

Construit

~~Learning Empirical Modelling:~~ Constructionism or Computational Thinking?

Antony Harfield

Department of Computer Science & IT
Naresuan University, Thailand

Overview

- Revisit the construals we met during the conference
- Describe construals and the kinds of learning that might be achieved through making construals
- Relate to Constructionism (C) and Computational Thinking (CT)
- Reflect on how to position Construit in relation to C and CT?

Motivation

- 3 years of Construit and many more leading up to it
- Practical examples of making construals for educational purposes
- Research papers relating to C and CT
- Perhaps some confusion about making construals and its relationship to C and CT

Reflections: Day 1

- Matti Tedre: CT is not something that began in 2006… it has a rich history that we should be aware of

- Errol Thompson: Computational Reasoning… focus on reasoning about a problem using computational techniques

- Duncan Maidens: CT not focused on languages, but instead on fundamentals like decomposition

- Hans-Joachim Petsche: The first ever Empirical Modeller in Forth?

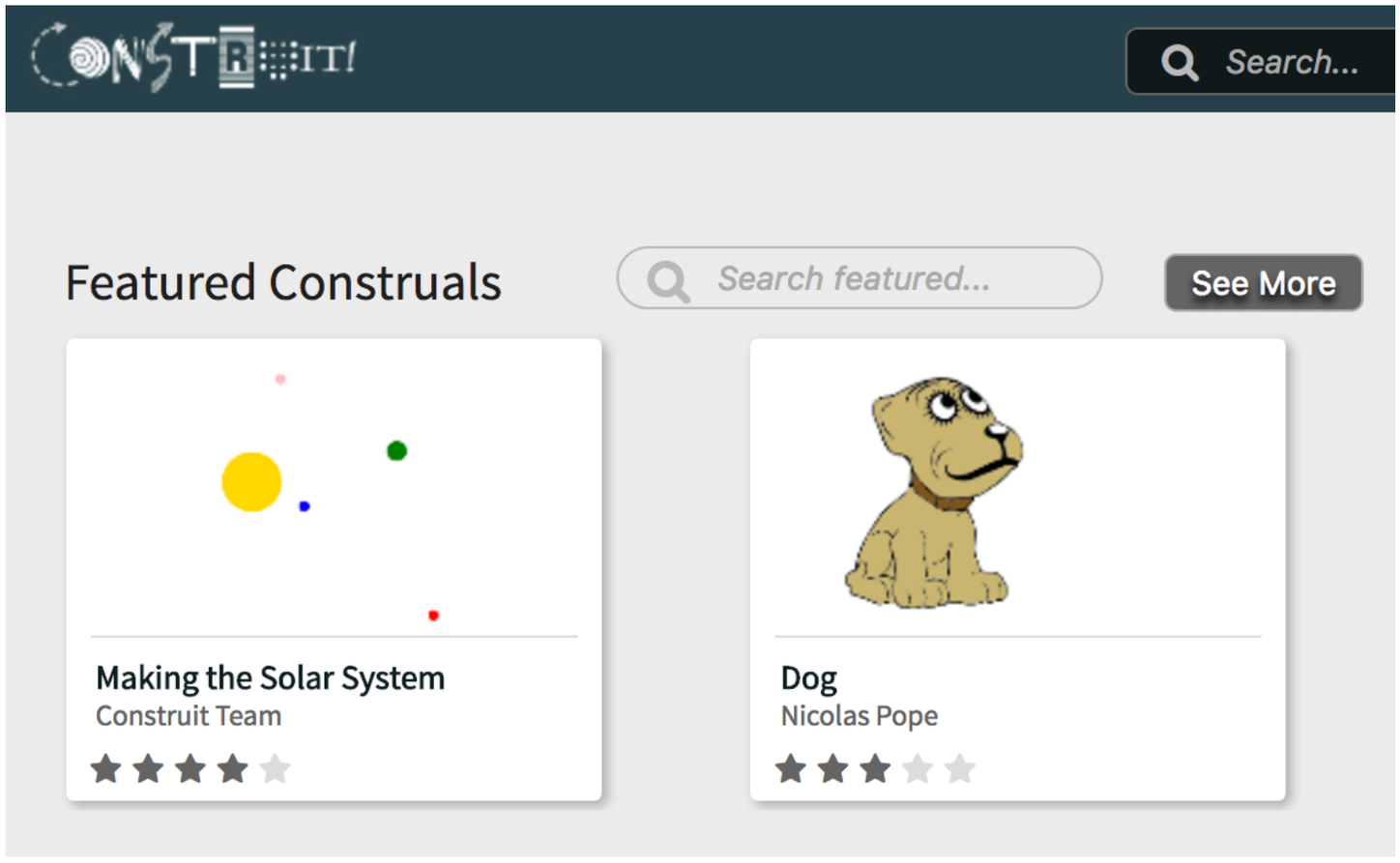
- Richard Cartwright: Modern software dev as creating dependencies/agency between micro-services (macro level)

Reflections: Day 2

Advert for “Piet’s papers”

Instead, let’s revisit some examples of making construals…

Solar System / Dog



→ learn how to “use” the tool (for beginners)

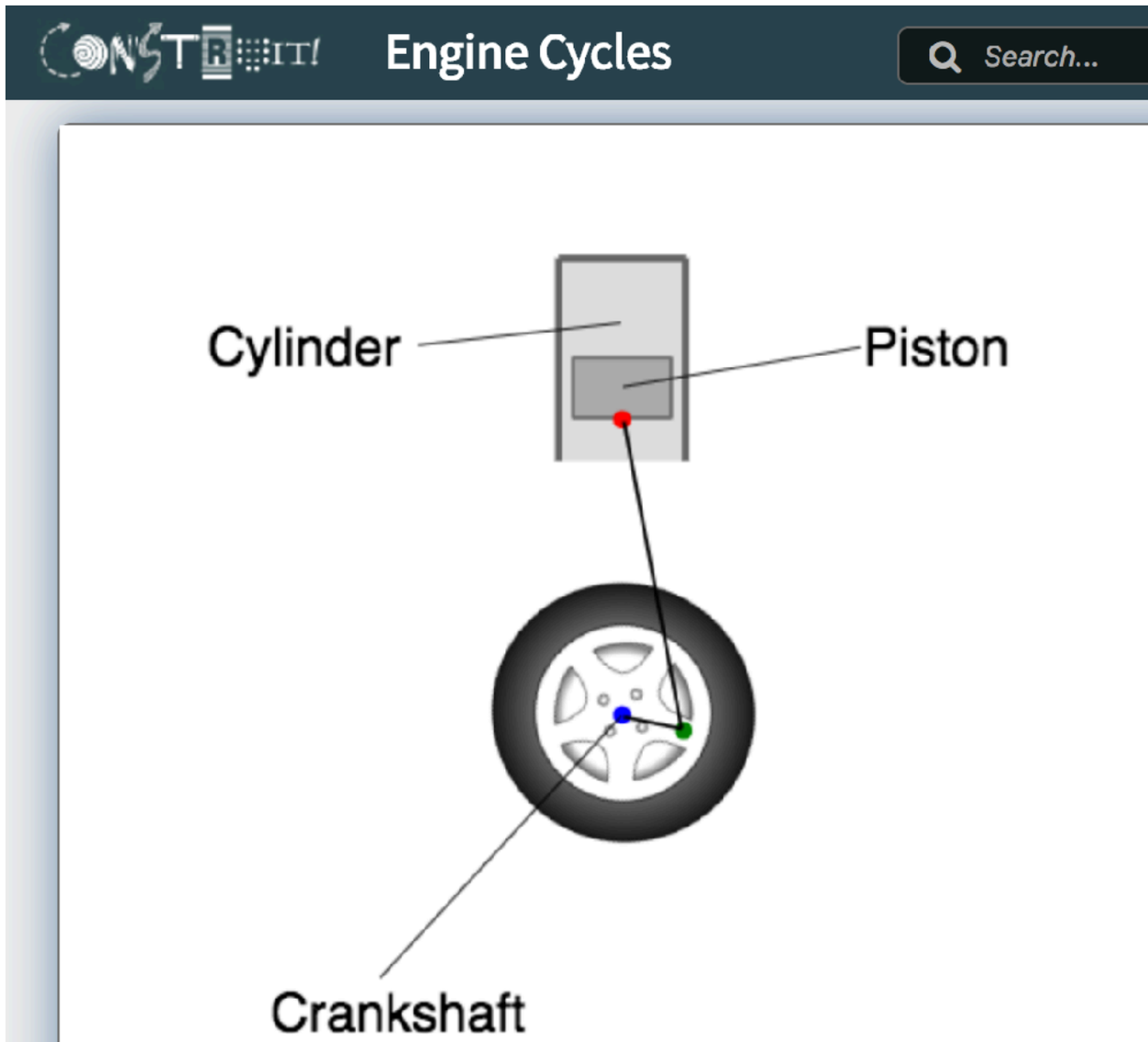
Learning method: Tutorial

Engine Cycles

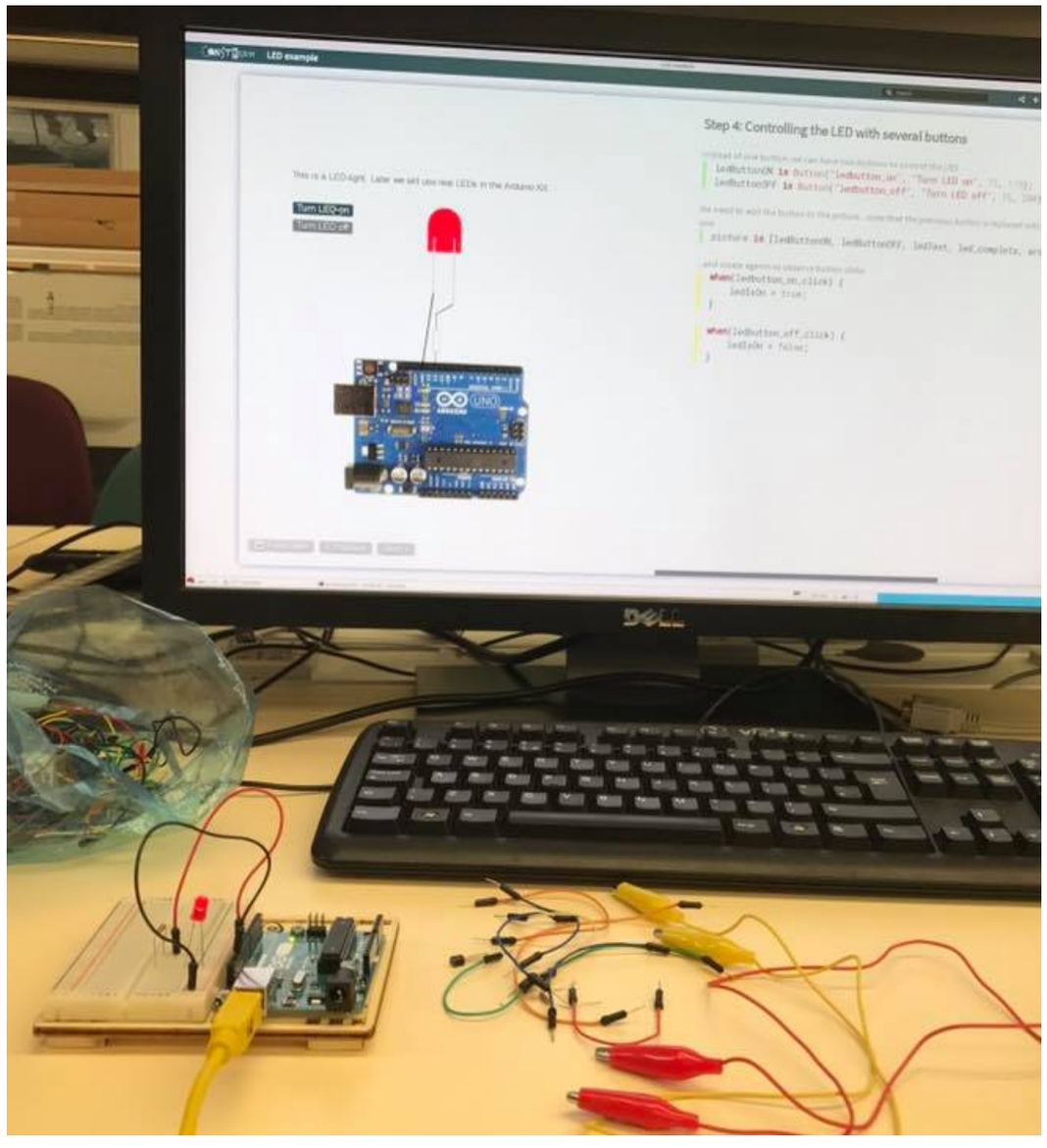
→ learn how a combustion engine works

(for secondary school students)

Learning method: Guided exploration



eCraft2Learn

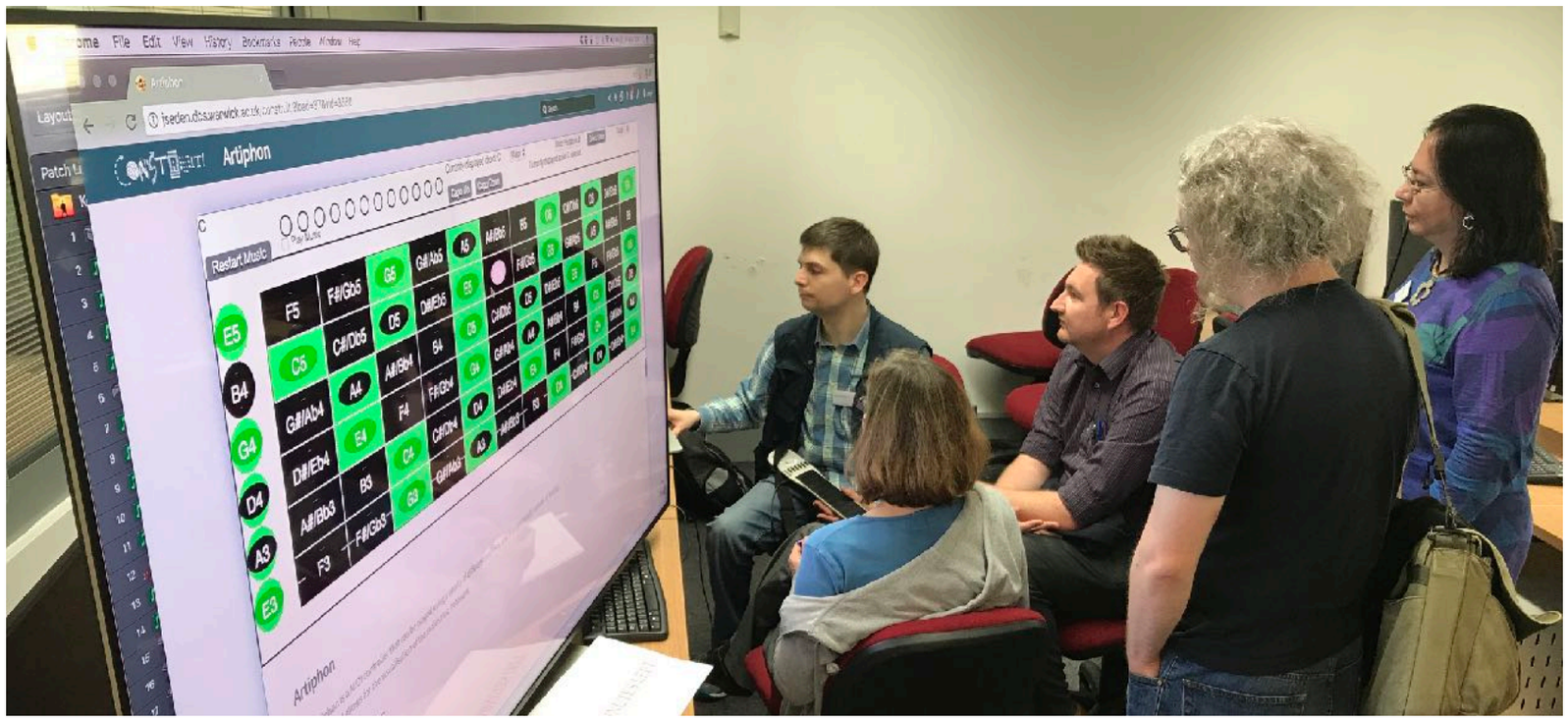


→ learn how to be a maker

(for secondary school students)

Learning method: Tutorial

Artiphon / Sound workshop



→ learn about playing an instrument (for musicians)

Learning method: Experimentation

Tales of Regular Polygons

→ highlight the richness and depth to making construals

(for Construit attendees!)

Learning method: Storytelling

Making construals

… those were just some of the examples

Topic Groups [work-in-progress]

Topic Groups devoted to specialist concerns and domains of application are intended to promote collaborative work towards the two goals for CONSTRUIT 2017 set out above. There are twelve of them at present such as ... Mathematics Education, Play and Games, Museum Projects and Studies, Decision Support and Policy Making, Medical Education. (Click here to see the index to Topic Groups.) Please feel free to suggest new Topic Groups (and potential co-ordinators of them).

Each group will have a range of relevant resources to include prototype construals and publications relating to the CONSTRUIT! project that is made available through this construal of the Topic Groups. The groups are distributed in time and place starting now and with members who may or may not be attending the conference. Those who are at the conference may participate in sessions to be scheduled as part of the CONSTRUIT 2017 Lab activities on Saturday July 15th. They are also encouraged to organise their own - "out of hours" [i.e. not part of the daily programme] - splinter group sessions.

A specific practical goal, and focus, for each group is to prepare a paper for publication and/or construct an interesting learning artefact by the end of October. Each group has an associated Google group to act as a forum for exchange and will have one or two co-ordinators to offer guidance, suggest issues or questions to address, promote collaboration, and monitor progress.

Index to Topic Groups

The following topic groups have been created:

- Programming and Making Construals >>
- Mathematics Education >>
- History of Science >>
- Museum Projects and Studies >>
- Modelling Mechanisms >>
- Educational Robotics >>
- Play and Games >>
- Practices and Texts >>
- Music >>
- Decision Support and Policy Making >>
- Medical Education >>
- Design and 3D printing >>

Click on a topic to access associated pages under construction.

http://jseden.dcs.warwick.ac.uk/construit/?load=226&vid=3179

Making construals

What kinds of learning/teaching does ‘making construals’ afford?

- Tutorial → step-by-step
- Guided exploration → steps + tinkering
- Experimentation → elaborating
- Storytelling → start from scratch

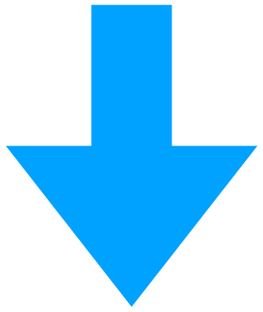
Making construals

What kinds of learning/teaching does 'making construals' afford?

Known method / Known outcome
Formal / Rehearsed

→ step-by-step **(for beginners)**

more creative?



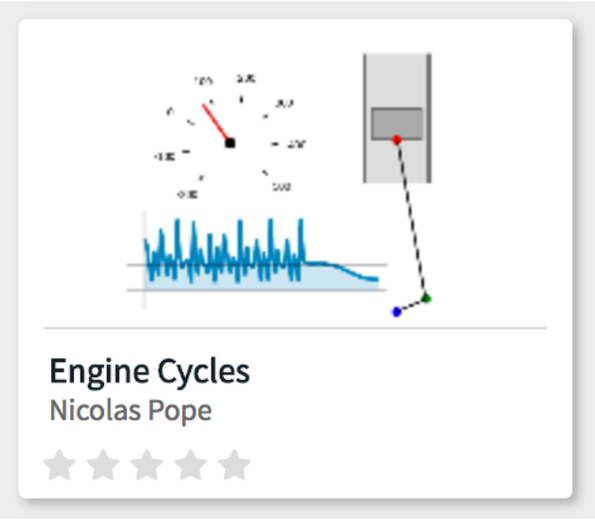
→ steps + tinkering

→ elaborating

Unknown method / Unknown outcome
Informal / Spontaneous

→ start from scratch **(experts)**

Engine Cycles



An example of how each kind of learning is relevant

- Create shapes/dependencies → step-by-step
- Explore how size affects speed → steps + tinkering
- Add 3 new cylinders → elaborating
- Show how cycle timing works → start from scratch

Making construals

- Supports kinds of learning from rote/repetitive through to the flexible/experiential/creative kinds.
  (Perhaps we recognise and appreciate this already?)

- But the same could be said for programming and 3D printing… so is there anything else special about “making construals”?

Making construals

- Built up incrementally, by expressing observables, dependency and agency (ODA)

- States and transitions closely connected with an object of study in the learning domain

- Learner should be able to directly experience a correspondence between the state of a construal and the state of an external object

Sets of connected ODA…

The Combustion Cycle

Automate the temperature changes based on stroke number.

```
when (intake_open) fuel = true;

when (spark && fuel) {
    temperature = room_temperature+200;
    fuel = false;
}

when (exhaust_open) temperature = room_temperature;
```

```
intake_open is stroke == 1;
exhaust_open is stroke == 4;
spark is stroke == 3;

crankVelocity += 50;
```

- Correspondence to some referent
- Also live, ’unstructured’, readily decomposable, self-documenting artefact

Making construals

Why ‘making construals’ instead of procedural programming?

- ODA…
    - make the connection between what is in the computer and what is in the world
    - manage the complexity of what is going on in the computer (“remove the state maintenance”, “focus on the glue”)
    - can be interacted (reasoned) with in a live, unstructured, de/recomposable way

Making construals

(bold claim alert)

… is *more accessible, easier to understand and more readily attainable* for learners than the constructing through conventional procedural programming (whether blocks or symbol based)

Constructionism and Computational Thinking

The Computational Thinker: Concepts & Approaches

Barefoot Computing

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

Concepts

Logic
predicting & analysing

Algorithms
making steps & rules

Decomposition
breaking down into parts

Patterns
spotting & using similarities

Abstraction
removing unnecessary detail

Evaluation
making judgement

Approaches

Tinkering
experimenting & playing

Creating
designing & making

Debugging
finding & fixing errors

Persevering
keeping going

Collaborating
working together

www.barefootcas.org.uk

Barefoot would like to acknowledge the work of Julia Briggs and the eLIM team at Somerset County Council for their contribution to this poster.

Computational Thinking

- A set of concepts and approaches (taken from CS and SE) that enable children to create software

- Focussed on creating programmers

Programming in schools

- Papert: Children should not learn programming for the sake of programming
- Instead, children can use programming to *create contexts in which a wide variety of learning experiences might occur*

Suggestion

- Procedural view of programming:
  - unsatisfactory for supporting current learning activities
    - insufficient for the vision of construction as a way to learn in any domain
- Instead, “making construals”:
  - conceptually more accessible to young people, while supporting a breadth of activities (kinds of learning)

Constructionism

("Dictionary definition")

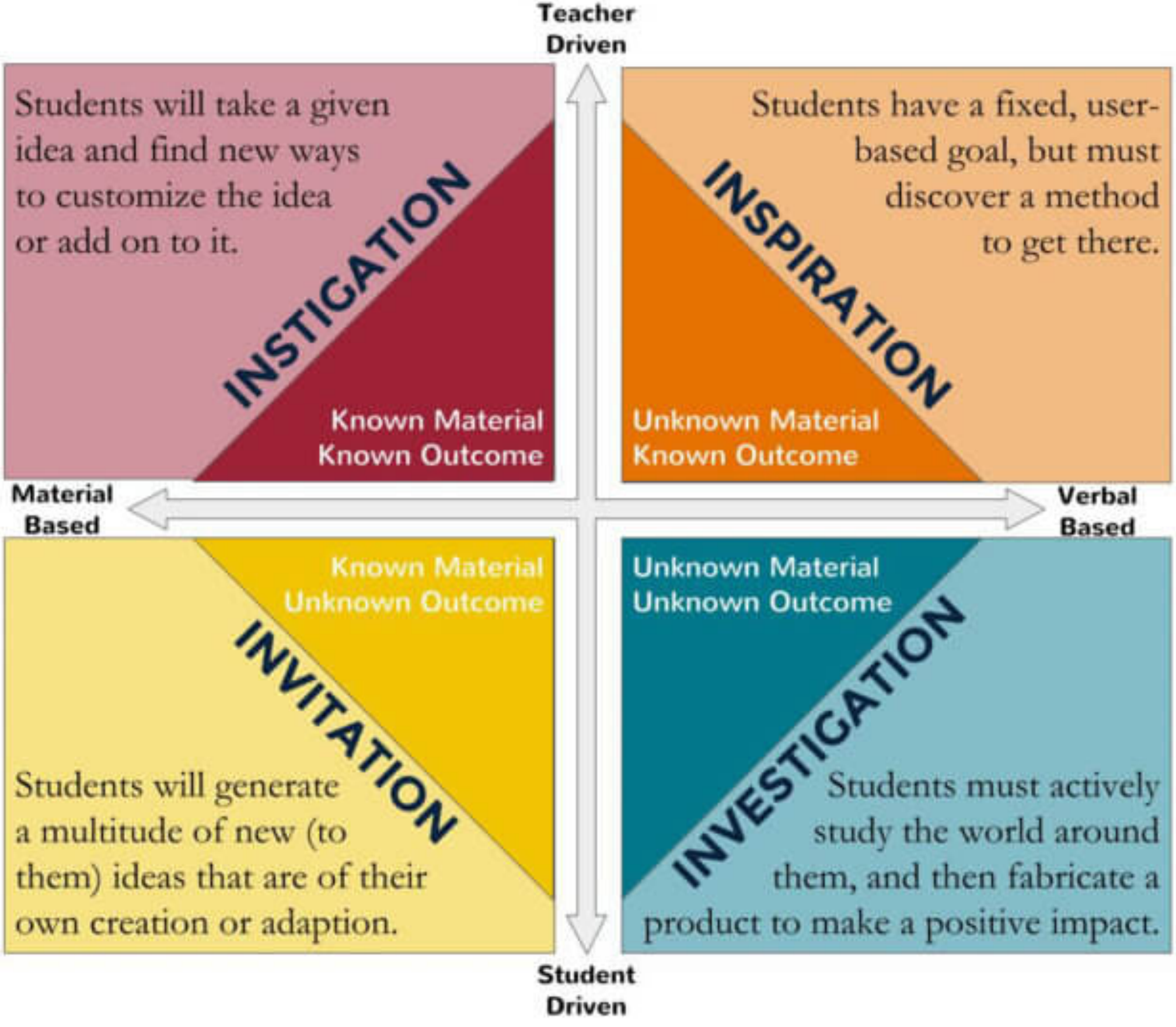
- "Construction that takes place in the head often happens especially felicitously when it is supported by construction of a more public sort 'in the world' - a sand castle or a cake […] Part of what I mean by 'in the world' is that the product can be shown, discussed, examined, probed, and admired." (Papert, 1980)

Constructionism

- “Computation has had a profound impact by concretizing and elucidating many previously subtle concepts in psychology, linguistics, biology, and the foundations of logic and mathematics. […] The most important (and surely controversial) component of this impact is on the child’s ability to articulate the working of his own mind and particularly the interaction between himself and reality in the course of learning and thinking.” (Papert, 1971)

(artefacts, referents, ODA)

Provocations: Constructionism applied to maker-centred classrooms

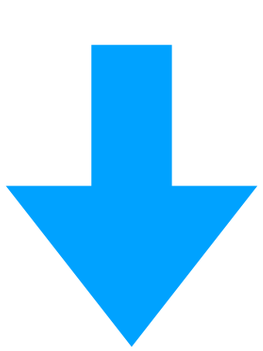


Making construals

What kinds of learning/teaching does 'making construals' afford?

Known method / Known outcome
Formal / Rehearsed

→ step-by-step **(for beginners)**



→ steps + tinkering

→ elaborating

Unknown method / Unknown outcome
Informal / Spontaneous

→ start from scratch **(experts)**

Conclusion

- Computational Thinking
    - A set of concepts and approaches (from CS) that enable children to create software (be programmers).
- Constructionism
    - A theory that learning is enhanced by creating, and the practical use of computers to support such construction (which may or may not involve programming).

- Computational Thinking
    - A set of concepts and approaches (from CS) that enable children to create software (be programmers).
- Constructionism
    - A theory that learning is enhanced by creating, and the use of computers to support such construction (which may or may not involve programming).
- Making construals
    - A set of concepts and tools that enable learners to create and explore software artefacts (…)
    - Backed by principles/theory that are compatible with constructionism