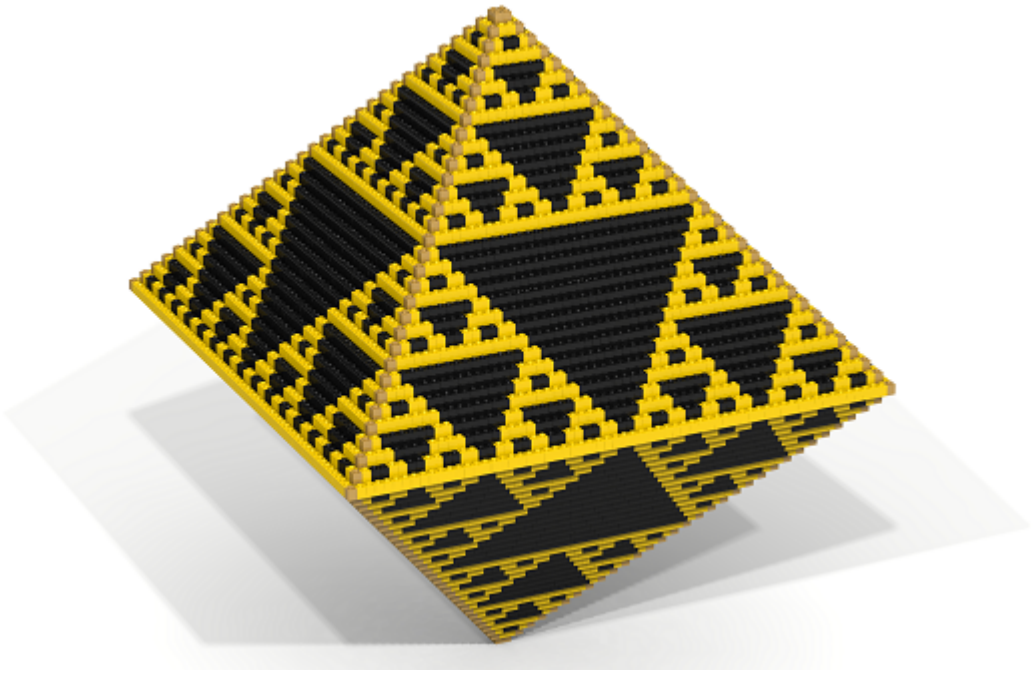
Bricklayer 10 Session Curriculum A



Copyright: bricklayer.org

Overview of Tool Supported Content

Tool	Topic/Activity	Covered
Vitruvia	Concept 1 - 2D coordinates	✓
	Concept 2 - property-based patterns	✗
	Concept 3 - placing a single brick	✓
	Concept 4 - placing multiple bricks	✓
	Concept 5 - calling a function to "put" a single brick	✓
	Concept 6 - sequences of "put" function calls	✓
	Concept 7 - overwriting, putting on brick on top of another	✓
	Concept 8 - students execute programs written by other students	✓
	Concept 9 - debugging programs	✗
	Concept 10 - nullary function declarations and calls	✓
	Concept 11 - use of offsets to shift the origin of the coordinate system	✓
	Concept 12 - parameterized function declarations and calls	✓
	Concept 13 - functions that create circles	✓
	Concept 14 - multi-parameter function *put2D*	✓
	Concept 15 - the multi-parameter function *lineXZ*	✓
	Concept 19 - the 3D function *put*	✓
Grid	Creation of user-defined patterns	✓
	Pentomino tessellations	✓
	Geometric Patterns - Laces (also uses Lace Animator tool)	✗
	Evolutionary Patterns	✓
	• Sierpinski Gasket	
	• Random Binary tree algorithm	
	• Elementary Cellular Automata rules	
	Students transfer pixel art images onto the Grid	✗
	Students execute programs written by other students	✗
bricklayer-lite	block-based coding to create Concept 5-6 artifacts	✓
	block-based coding to create Concept 7 artifacts	✓
	block-based coding to create Level_1 (limited palette) pixel art	✓
	block-based coding that uses *circleXZ* and *ringXZ* functions	✗
	block-based coding to create Concept 13 artifacts	✗
	block-based coding to create Concept 14 artifacts	✗
	block-based coding to create Level_3 (full palette) pixel art	✗
	writing code for Concept 15 artifacts	✗
bricklayer	text-editor basics	✗
	text-based coding for Concept 19 artifacts	✗

Copyright: bricklayer.org

Comments

- Links in this document are "clickable".
- Web navigation is highlighted on a blue background.
- It can be very beneficial for the teacher to work problems alongside students. For example, if students are coding in bricklayer-lite, then the teacher also codes in bricklayer-lite with their work projected on the screen for all to see. Another example (very useful) is for the teacher to create a lace (projected on the screen) alongside the students. This way students whose understanding is unclear can see (in real time) someone creating a lace. Additionally, when a teacher works alongside their students they get a better sense of time. Of course, working alongside students must be balanced against inspecting the work of individual students and giving them help and feedback.
- It can be very beneficial to "warm up" every class period by doing Vitruvia exercises. Repeating exercises is good, so even if students have already done all the exercises they can repeat exercises.
- Students prefer coding using the bricklayer-lite visual editor over coding using a text-based editor (e.g., notepad++).
- Vitruvia concept 2 should be skipped initially. However, more advanced students can do this.
- Vitruvia concepts 10 and 11 can be skipped depending on class interest. These concepts provide an intuitive lead-in to parameterized functions (concept 12). Offsets (concept 11) become important when using Bricklayer to create artifacts in Minecraft.
- Students generally prefer to come up with their own artifacts. If suitable student generated assignments can be used in place of Bricklayer assignments. Bricklayer assignments can be seen as a suggestion for what kinds of artifacts can be created.
- Pixel art is generally very popular. One can literally spend an entire semester doing just pixel art. Pixel art can be coded individually or in groups. Pixel art generally requires access to brick colors that only become available at Level 3. It is at Level 3 where the "put" function first becomes parameterized on the shape, brick, and coordinate. The set of bricks supported by Bricklayer can be found here:

  `http://wintercoding.com/BricklayerWeb/legend_LEGO.html`

  One can (partially) get to Level 3 quickly by skipping Level 2 altogether. Then later one can return to Level 2.
- 3D is also very popular.

Problems and Solutions

- Be sure to confirm that your school's *web policy* does not block the bricklayer.org web site. Note that schools will block YouTube (for obvious reasons). This is okay if the teacher can still access YouTube. Instructional videos for students are at screencast.com.
- If there is a problem with a computer then have students pair up on a working computer.
- Unplugged exercises (e.g., tessellation on a sheet of graph paper) can be given to students that misuse the computer.

Resources

Web site	`http://bricklayer.org`
Educator Video	`https://bricklayerdotorg.wordpress.com/about/`
Introductory Videos	`https://bricklayerdotorg.wordpress.com/level-1/`

Copyright: bricklayer.org

Standards

Computer Science Teachers Association
`http://csta.acm.org/Curriculum/sub/CurrFiles/CS_K-8_Building_a_Foundation.pdf`

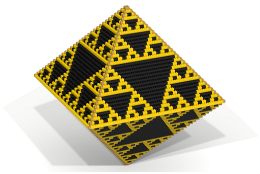
Goals

- Develop coding and math skills.
- Develop and understanding of various categories of patterns.
- Develop an appreciation/awareness of patterns (e.g., floor tilings, clothing, flowers, etc.).
- Develop the ability to navigate the Bricklayer web site in order to answer questions. Question examples include:
  - What colors of LEGO bricks are supported?
  - What is the syntax of a function call that creates a circle?

Terminology

- the GRID (or grid) – refers to the grid web app found at *bricklayer.org* → *WEB APPS* → *GRID*.
- code-along – this is an activity where the teacher and student both do the same thing at the same time.
- activity – an activity is something that is primarily done by students. However, it is beneficial for the teacher to also work the activity alongside their students and display their work on the screen for all to see.

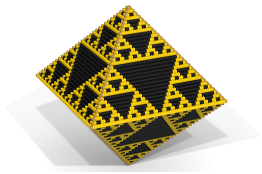
Copyright: bricklayer.org



Lesson 1 (50 minutes)

Day	Activity
1	• 10 minutes: Lecture – What is Coding? – *(Navigate to) bricklayer.org* – Overview of Bricklayer – Live/Video. `https://www.youtube.com/watch?v=-G-wEhvVRBo` • 20 minutes: Code Along – *(Navigate to) bricklayer.org → STUDENT → UNPLUGGED*. – Discuss pattern categories: arithmetic, geometric, evolutionary. – Discuss creation of user-defined repeating patterns. • 20 minutes: Activity (could be longer, the teacher can work alongside the students) – *bricklayer.org → WEB APPS → GRID* – Demonstrate how the grid works. – Create your own repeating pattern. It must repeat. – Then switch with your neighbor and try extending their pattern.

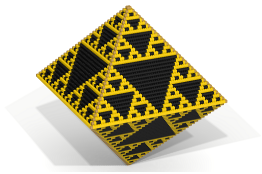
Copyright: bricklayer.org



Lesson 2 (50 minutes)

Day	Activity
2	• 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concept 1*. – Discuss concept 1. – Do as many exercises as possible in the allotted time. If you finish early, go to the GRID and start working on creating a user-defined repeating pattern. • 10 minutes: Lecture – *bricklayer.org → STUDENT → UNPLUGGED*. View collection of polyominoes. – Demonstrate how to perform a tessellation of the GRID using a pentomino. • 30 minutes: Activity – *bricklayer.org → WEB APPS → GRID* – Display collection of polyominoes on projector. – Perform tessellations of the GRID using various polyomino shapes (e.g., pentominoes).

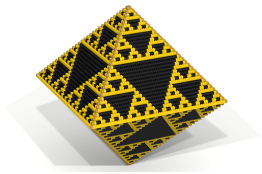
Copyright: bricklayer.org



Lesson 3 (50 minutes)

Day	Activity
3	• 10 minutes: Lecture – *bricklayer.org → WEB APPS → LACE ANIMATOR*. – Use the GRID and the LACE ANIMATOR to explain the algorithm for constructing geometric patterns. – Detailed instructions for how to create a lace can be found here: `http://wintercoding.com/assignments/By%20Hand/content/Laces.pdf`. • 10 minutes: Code-Along – *bricklayer.org → WEB APPS → GRID*. – Construct a lace (i.e., geometric) pattern. • 30 minutes: Activity – *bricklayer.org → WEB APPS → GRID* – Design your own geometric pattern or re-create existing ones. – Lace examples can be found at: *bricklayer.org → STUDENT → UNPLUGGED* – in the Geometric Patterns section.

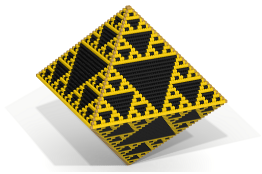
Copyright: bricklayer.org



Lesson 4 (50 minutes)

Day	Activity
4	• 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concept 3* – Discuss concept 3. – Do as many exercises as possible in the allotted time. • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concept 4* – Discuss concept 4. – Do as many exercises as possible in the allotted time. • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concept 5* – Discuss concept 5. – Do as many exercises as possible in the allotted time. • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concept 6* – Discuss concept 6. – Do as many exercises as possible in the allotted time. • 10 minutes: Lecture – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Give overview of bricklayer-lite Level_1.

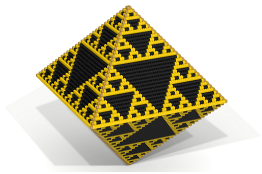
Copyright: bricklayer.org



Lesson 5 (50 minutes)

Day	Activity
5	• 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → Concepts 3-6* – Get more practice with these concepts. Alternately, more advanced students can code in bricklayer-lite. • 10 minutes: Code Along – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Create one or more Level_1 programs using bricklayer-lite that produce an artifact corresponding to a Vitruvia concept 5/6. That is, go to a Vitruvia exercise look at the pattern created and try to code that up. • 20 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – free form coding • 10 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 1 coding assignments (*bricklayer.org → STUDENT → LEVEL 1*) corresponding to concepts 5 and 6.

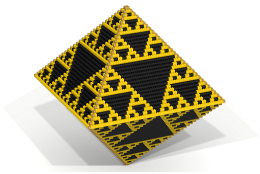
Copyright: bricklayer.org



Lesson 6 (50 minutes)

Day	Activity
6	• 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 7* – Discuss concept 7. – Do exercises from concept 7 • 40 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 1 coding assignments (*bricklayer.org → STUDENT → LEVEL 1*) corresponding to concept 7.

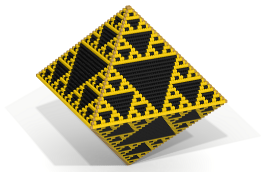
Copyright: bricklayer.org



Lesson 7 (50 minutes)

Day	Activity
7	• 10 minutes: Lecture – Discuss algorithms for creating pixel art. • 40 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Pixel art! One can spend many many lessons doing pixel art. Pixel art can be done individually or in groups. ∗ Individual Activity - transfer (i.e., copy) a pixel art image onto the Bricklayer grid. ∗ Group Activity 1 - one person creates an artifact on the Grid and the other person codes it up in bricklayer-lite. ∗ Group Activity 2 - (2-4 students) a pixel art image is printed and cut into pieces, each student codes up their piece. Composing these pixel art pieces can be done in bricklayer-lite, but it is a bit tricky. It is easier to compose the resulting code using its textual representation (e.g., compose the Bricklayer code not the bricklayer-lite code).

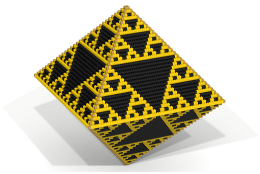
Copyright: bricklayer.org



Lesson 8 (50 minutes)

Day	Activity
8	• 10 minutes: Lecture – Talk about nullary functions. – *bricklayer.org → STUDENT → LEVEL 2 → Nullary Functions* – Direct link: `https://bricklayerdotorg.wordpress.com/level-2-nullary-functions/` • 10 minutes: Lecture – A powerpoint presentation that uses animation to show how sequences of function calls can be converted to become the body of a nullary function declaration can be found here: `http://wintercoding.com/assignments/By%20Machine/level_2/reading/Nullary_function_animations.pptx` This link also is on the Bricklayer web page in the text associated with the following bullet point: *bricklayer.org → STUDENT → LEVEL 2 → Nullary Functions* • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 10* – Discuss concept 10. – Do as many exercises as possible in the allotted time. • 20 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 2 coding assignments (*bricklayer.org → STUDENT → LEVEL 2*) corresponding to concept 10. – Note - an additional lesson can/should be devoted to getting comfortable with declaring and using nullary functions.

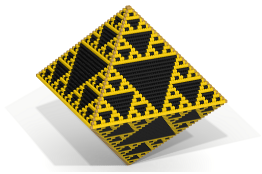
Copyright: bricklayer.org



Lesson 9 (50 minutes)

Day	Activity
9	• 10 minutes: Lecture – Talk about offsets. – *bricklayer.org → STUDENT → LEVEL 2 → Offsets* – Direct link: `https://bricklayerdotorg.wordpress.com/level-2-offsets/` • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 11* – Discuss concept 11. – Do as many exercises as possible in the allotted time. • 30 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 2 coding assignments (*bricklayer.org → STUDENT → LEVEL 2*) corresponding to concept 11. – Note: offsets can be extremely tricky. Several lessons can be devoted to experimenting with offsets.

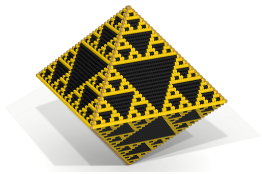
Copyright: bricklayer.org



Lesson 10 (50 minutes)

Day	Activity
10	• 10 minutes: Lecture – Talk about pameterized functions. – *bricklayer.org → STUDENT → LEVEL 2 → Parameterized Functions* – Direct link: `https://bricklayerdotorg.wordpress.com/level-2-parameterized-functions/` • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 12* – Discuss concept 12. – Do as many exercises as possible in the allotted time. • 30 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 2 coding assignments (*bricklayer.org → STUDENT → LEVEL 2*) corresponding to concept 12.

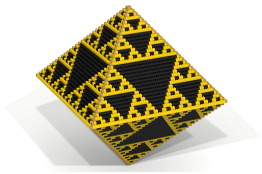
Copyright: bricklayer.org



Extra Lesson I (50 minutes)

Day	Activity
11	• 10 minutes: Lecture – *bricklayer.org → BRICKLAYER → DOCUMENTATION → LEVEL 2* – Explain how to read the descriptions of Level 2 functions. – Discuss how to call the functions: *circleXZ* and *ringXZ*. • 10 minutes: Code-Along (this should probably be skipped) – *bricklayer.org → WEB APPS → VITRUVIA → concept 13* – Discuss concept 13. • 30 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 2 coding assignments (*bricklayer.org → STUDENT → LEVEL 2*) corresponding to concept 13.

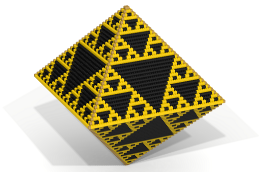
Copyright: bricklayer.org



Extra Lesson II (50 minutes)

Day	Activity
12	• 5 minutes: Lecture – Talk about function parameters – Example 1: an oven is a function that accepts parameters – Example 2: a microwave also has preset parameters – discuss *put2D* vs *put2D_1x1_BLUE* • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 14* – Discuss concept 14. – Do as many exercises as possible. • 10 minutes: Code-Along – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – free form – create one or more programs that use the put2D function. • 25 minutes: Activity – *bricklayer.org → WEB APPS → BRICKLAYER-LITE* – Level 3 coding assignments (*bricklayer.org → STUDENT → LEVEL 3*) corresponding to concept 14.

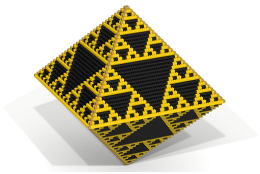
Copyright: bricklayer.org



Extra Lesson III (50 minutes)

Day	Activity
13	• 15 minutes: Lecture/Code Along – using the Bricklayer IDE/APP – copy, paste, undo, save, open, find, replace, clone, block tab, column edit, name a file with a dot-bl extension, etc. • 10 minutes: Activity – Translate one or more of bricklayer-lite Level 1 or 2 programs (e.g., it is okay to translate example programs on bricklayer-lite web site) so that it uses the *put2D* function found in Level_3. • 10 minutes: Lecture – Error messages – `http://wintercoding.com/guide/syntax_errors/syntax_errors_01.pptx` – engage students – can they spot the syntax error's in the code? • 15 minutes: Activity – Bricklayer code.

Copyright: bricklayer.org



Extra Lesson IV (50 minutes)

Day	Activity
14	• Note: Our experiences suggest that transitioning from 2D to 3D is a big cognitive step – especially for younger students. • 10 minutes: Activity – *bricklayer.org → WEB APPS → VITRUVIA → concept 19* – Discuss concept 19. – Do as many exercises as possible. • 40 minutes: Activity – *bricklayer.org → STUDENT → LEVEL 4* – Do as many coding assignments related to (Vitruvia) concept 19 as you can. When you finish an assignment show us your result. If you need help, let us know.

Copyright: bricklayer.org

Extra Lesson V (50 minutes)

Day	Activity
15	• 10 minutes: Lecture – *BRICKLAYER → DOCUMENTATION → LEVEL_4* – *sphere*, *hollowSphere* – *ringY* • 40 minutes: Activity – Create Saturn or something...

Copyright: bricklayer.org