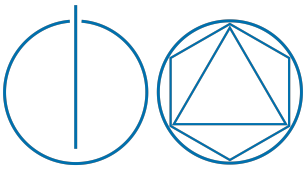
FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Interdisciplinary Project in Mathematics

Random Generation of Tangrams

Wiebke Köpp



FAKULTÄT FÜR INFORMATIK

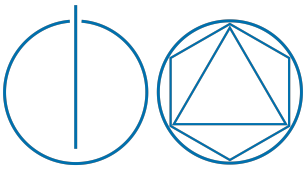
TECHNISCHE UNIVERSITÄT MÜNCHEN

Interdisciplinary Project in Mathematics

Random Generation of Tangrams

Zufällige Generierung von Tangrams

Author:	Wiebke Köpp
Supervisor:	Prof. Dr. Dr. Jürgen Richter-Gebert
Advisor:	Dipl.-Inf. Martin von Gagern
Submission Date:	April 10, 2015



I assure the single handed composition of this interdisciplinary project in mathematics is only supported by declared resources.

Munich, April 10, 2015 Wiebke Köpp

Abstract

Kurzfassung

Contents

1 Introduction **1**

2 Background **3**
2.1 Tangram and Tans 3
2.2 Coordinates 3
2.3 Points 3
2.4 Line Segments 3

3 Design **4**
3.1 Overall Application Structure 4
3.2 Generation Process 4
3.3 Interestingness Measures 4
3.4 Gameplay 4

4 Implementation **5**

5 Results **7**
5.1 Generation Process 7
5.2 Interestingness Measures 7
5.2.1 User testing 7

6 Future Work and Conclusion **8**

List of Figures **9**

List of Tables **10**

Bibliography **11**

1 Introduction

Tangram is an old Chinese dissection puzzle whose rules are easily understood, but which can also be quite challenging. Seven puzzle pieces, called tans, have to be placed within a given shape in a way such that the entire shape is covered. Additionally, none of the pieces are allowed to overlap and all seven tans are to be used. As shown below, the seven puzzle pieces are three- and four-sided convex geometrical shapes and can be derived from cutting a square in a specific way.

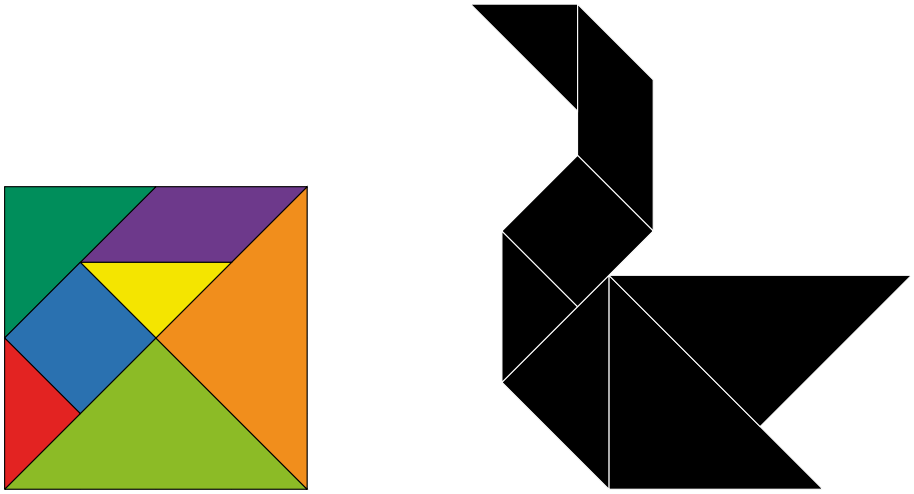


Figure 1.1: Dissection of a square into the 7 tans and an example for a given shape

frequentlly used in education

The objective of this project is to create a tangram game that can be played in a browser. The user should first be presented with a number of tangrams to choose from and then be able to attempt solving the chosen tangram. While there exist many collections of tangrams published in books, adding them to database would require potentially time-consuming manual insertion. Therefore, a different approach will be applied here. Instead of displaying pre-defined shapes, the presented tangrams are supposed to be generated randomly. The resulting tangrams should not be entirely random however, but be interesting in some way, which is achieved generating a large number of tangrams that are then sorted according to an interestingness measure.

Additionally, the the generation process is controlled in a way such that puzzles with immediately obvious solutions are avoided.

as well as interesting. Achieving the generation of interesting tangrams. easuring the interestingness of a tangram is with possible candidates being the difficulty explored as a part could incorporate visual aesthetics. The notion of interestingness of.

Within tangram collections, shapes are often explicitly categorized according to their correspondence to real world objects like animals, people, numbers, letters or simple geometrical forms and are usually not ordered according difficulty.

Previous publications about tangrams concern the history of tangrams. An overview on the history on tangrams can found in. This books also contains some analysis of the mathematical structure of tangrams The earliest collections of tangram patterns date back to the beginning of the nineteenth century [4] and since then many collections, some even containing more than 1,000 different shapes have been published all over the world. A common element of these collections is that the tangrams are categorized according to which real world objects they could represent. Some examples for such categories like people, animals, numbers, letters. An extensive enumeration of such collections can be found in [18]

Related Work

There have been some attempts to classify tangrams. History , [19] Have been recently revisited by Fox-Epstein [6], dealing with a Japanese variation of the tangram. Solution [2] [17] [13] Aesthetics [3] Birkhoff's [5] Education and how to use tangrams in geometric concepts like [12] convexity difficulty lever off other well known puzzles like Sokoban [10] or Soduko [9] [1]

Outline

This report is organised as follows. Chapter 2 describes which mathematical concepts are applied in generating tangrams, measuring their interestingness and conducting computations on individual and groups of tans in a game setting. The following chapter first introduces the structure of the overall application and then describes the algorithms involved. Some implementation details are presented in chapter 4. This chapter specifically shows which features of the used programming language JavaScript are advantageous for the implementation of a random tangram generator in a browser. Chapter 5 shows the effect of different parameter settings during both generation and ranking of tangrams, as well as the results of a small user study. Finally, chapter 6 mentions potential future enhancements to different aspects of the application and concludes this report.

2 Background

2.1 Tangram and Tans

representation of a tangram/tan position of the anchor point and orientation fully describes a tan

2.2 Coordinates

Z sqrt 2 addition multiplication

2.3 Points

relative orientation transformation point containment

2.4 Line Segments

line segment intersection

3 Design

3.1 Overall Application Structure

Commucate with database

3.2 Generation Process

First Naive Approach

As a first step towards the generation of interesting tangrams, the generation algorithms should favour configurations in which tans do not only have vertices, but also edges in common.

3.3 Interestingness Measures

convex hull

3.4 Gameplay

Outline computation Snapping

4 Implementation

The tangram generator and its associated interfaces for choosing one tangram out of a given number of presented ones and for solving a chosen tangram are implemented in JavaScript. JavaScript is a scripting language originally designed for adding interactivity to web pages by manipulating the structure and content of HTML-documents, but in recent years has also gained popularity in other domains like game development and server-side applications. Most modern browsers on both desktop and mobile devices include a JavaScript engine, which means that the user is not required to install additional frameworks for an application to execute properly. Other technologies for running client-side computations in a browser, like Java Applets, do not provide such widely spread support and have additionally experienced declining popularity due to security issues. In consequence, Javascript is well suited for an application targeted to support various input paradigms on different devices [16].

The Document Object Model (DOM) is an interface to HTML and XML documents. It allows accessing and changing the elements of a document and their properties as well as attaching event handlers to elements. Almost all changes in the interface of the tangram generator are realised with DOM manipulations. On startup, the web page contains structural elements for all parts of the interface that will be displayed during execution. This includes elements for each of the six tangrams, an area for playing the game and buttons for invoking processes not directly associated with a specific element. While some elements, like the buttons, are only hidden when first visiting the page, others have yet to be filled with content, like the elements displaying tangrams or the game play. When displayed, tangrams are drawn as Scalable Vector Graphics (SVG) [14], exploiting the fact that SVG is XML-based. The elementry of an SVG-element are therefore part of the DOM and can be treated like any other element. An alternative to using SVG as a drawing method is the HTML5 canvas element. Contrary to SVG, the canvas element is raster-based. After an element has been drawn it can not be updated in any way. Using SVG, moving a tan corresponds to updating the `x` and `y` attributes of the corresponding polygon. Achieving the same result with canvas on the other hand requires for the entire scene to be redrawn. Another advantage of using SVG as the graphics rendering methods, is the possibility of attaching event handlers to SVG elements. Event handlers are functions that are executed in case a certain event happens. Typical events in the scope of web pages are events involving user interaction

through mouse, keyboard and touch or browser actions. The tangram generator almost solely makes use of mouse and touch events in order to make the interface interactive on both desktop and mobile devices. Event handlers for clicking and dragging are attached to an element once it is added to the DOM and the translation, rotation and flipping of tans.

JavaScript code is executed in a single thread and reacts asynchronously to events such as the ones described above. This implicates that heavier computations like the generation of a large number of tangrams block the simultaneous execution of any other code. Informing the user about the current state of the application during such computations is crucial to provide a satisfying user experience. Web workers [8] are a technology introduced to Javascript to allow the execution of scripts in an additional thread in the background. In contrast to the main execution thread, workers cannot directly access the DOM or use methods and properties associated with the current window. They can however communicate with the main thread in the form of messages that can be handled in the same way as any other event. Thus, Web workers enable showing the progress of the generation process without having to repeatedly interrupt it. The web worker handling the entire generation process is started immediately after the webpage with the tangram generator is entered. Each time a tangram has been generated, the worker sends a message to the main thread, which then updates the progress bar accordingly. After the desired number of tangrams has been generated, the web worker finishes by sending the generated tangrams to the main thread.

Which kind of messages can be exchanged between main thread and web workers is browser-dependent. Fortunately, all browsers are capable of sending `String` messages between threads. JavaScript Object Notation (JSON) is a key-value based data-interchange format with which an object can easily be transformed into a sendable String, so that objects like the generated tangrams can be exchanged as well. An example for a JSON representation of tan can be seen below.

```
{"tanType": 0,
 "anchor": {"x": {"coeffInt": 42, "coeffSqrt": -6},
            "y": {"coeffInt": 30, "coeffSqrt": -6}},
 "orientation": 3}
```

JSON objects are also used to send statistics about chosen tangrams and played games to a simple HTTP Server written in Node.js [11], a platform often referred to as server-side JavaScript. The server writes all JSON files it receives into a database. The database used here is MongoDB [15], which as a document-oriented database, functions very well with JSON objects.

5 Results

5.1 Generation Process

Comparison of the two algorithms Comparison of what happens with different settings for probability -> sufficiently large number of tangrams generated no visible difference on the first 6 tangrams

5.2 Interestingness Measures

Results of different measures

5.2.1 User testing

In addition to the interface described in section **??** evaluation interface Out of 1000 tangrams with properties. compact holes just one hanging piece small edges First six were the same for all users Overall .. took part.

6 Future Work and Conclusion

Interestingness: Finding the number of solutions. Even the number of possible locations for the large triangle could provide Including combinations Let the user decide on which properties he finds interesting potentially be providing sliders or checkmarks Apply techniques from information retrieval like Learning to rank algorithms and relevance feedback [7]

Game: Hint based on already places tans Save and Share tangrams Local storage to come back later More advanced touch interaction. Rotation by two fingers Rotations by multiples of 15 degrees

List of Figures

1.1 Dissection of a square into the 7 tans and an example for a given shape 1

List of Tables

Bibliography

[1] S. Cai, Y. Jiang, and M. Zeng. *The Research on the Difficulty Grade and the Piecing Skills of Tangram*. Shenzheng Foreign Languages School. 2008. URL: http://www.yau-awards.org/Notice.php (visited on 1/19/2015).

[2] E. S. Deutsch and K. C. Hayes Jr. "A Heuristic Solution to the Tangram Puzzle." In: *Machine Intelligence* 7.Bernard Meltzer and Donald Michie (1972), pp. 205–240. Edinburgh University Press.

[3] R. S. Eberle. "The Role of Children's Mathematical Aesthetics: The Case of Tessellations." In: *The Journal of Mathematical Behavior* 35 (2014), pp. 129–143.

[4] J. Elffers. *Tangram. Das alte chinesische Formenspiel*. Du Mont, 1978.

[5] D. Filonik and D. Baur. "Measuring Aesthetics for Information Visualization." In: *Information Visualisation, 2009 13th International Conference*. IEEE. 2009, pp. 579–584.

[6] E. Fox-Epstein and R. Uehara. "The Convex Configurations of "Sei Shonagon Chie no Ita" and Other Dissection Puzzles." In: (2014). arxiv preprint: arxiv:1407.1923v1[cs.CG].

[7] C. He, C. Wang, Y.-X. Zhong, and R.-F. Li. "A Survey on Learning to Rank." In: *2008 International Conference on Machine Learning and Cybernetics*. Vol. 3. IEEE. 2008, pp. 1734–1739.

[8] I. Hickson, ed. *Web Workers*. Candidate Recommendation. W3C. May 1, 2012. URL: http://www.w3.org/TR/2012/CR-workers-20120501/ (visited on April/9/2015).

[9] M. Hunt, C. Pong, and G. Tucker. "Difficulty-Driven Sudoku Puzzle Generation." In: *UMAP Journal* (2007), pp. 343–362.

[10] P. Jarušek and R. Pelánek. "Difficulty Rating of Sokoban Puzzle." In: *STAIRS 2010: Proceedings of the Fifth Starting AI Researchers' Symposium*. IOS Press. 2010, p. 140.

[11] Joyent, Inc. *Node.js v0.12.2 Manual & Documentation*. 2014. URL: https://nodejs.org/api/ (visited on April/9/2015).

[12] J. Köller. *Mathematische Basteleien - Tangram*. 1999. URL: http://www.mathematische-basteleien.de/tangram.htm (visited on 02/01/2015).

[13] S. Z. Kovalsky, D. Glasner, and R. Basri. "A Global Approach for Solving Edge-Matching Puzzles." In: (2014). arXiv preprint: arXiv:1409.5957v1[cs.CV].

[14] C. McCormack, J. Watt, D. Schepers, A. Grasso, P. Dengler, J. Ferraiolo, E. Dahlström, D. Jackson, J. Fujisawa, and C. Lilley, eds. *Scalable Vector Graphics (SVG) 1.1 (Second Edition)*. Recommendation. W3C. Aug. 16, 2011. URL: `http://www.w3.org/TR/2011/REC-SVG11-20110816/` (visited on April/9/2015).

[15] MongoDB, Inc. *The MongoDB 2.4 Manual*. 2015. URL: `http://docs.mongodb.org/v2.4/` (visited on April/9/2015).

[16] Mozilla Developer Network and individual contributors. *JavaScript*. Mar. 24, 2015. URL: `https://developer.mozilla.org/en-US/docs/Web/JavaScript` (visited on April/9/2015).

[17] K. Oflazer. "Solving tangram puzzles: A connectionist approach." In: *International Journal of Intelligent Systems* 8.5 (1993), pp. 603–616.

[18] J. Slocum. *The Tangram Book*. Sterling Publishing, 2003.

[19] F. T. Wang and C.-C. Hsiung. "A Theorem on the Tangram." In: *The American Mathematical Monthly* 49.9 (1942), pp. 596–599.