Shopping for Game levels: A Visual Analytics Approach to Exploring Procedurally Generated Content

AHMED M. ABUZURAIQ, OSAMA ALSALMAN, and HALIL ERHAN, Simon Fraser University, Canada

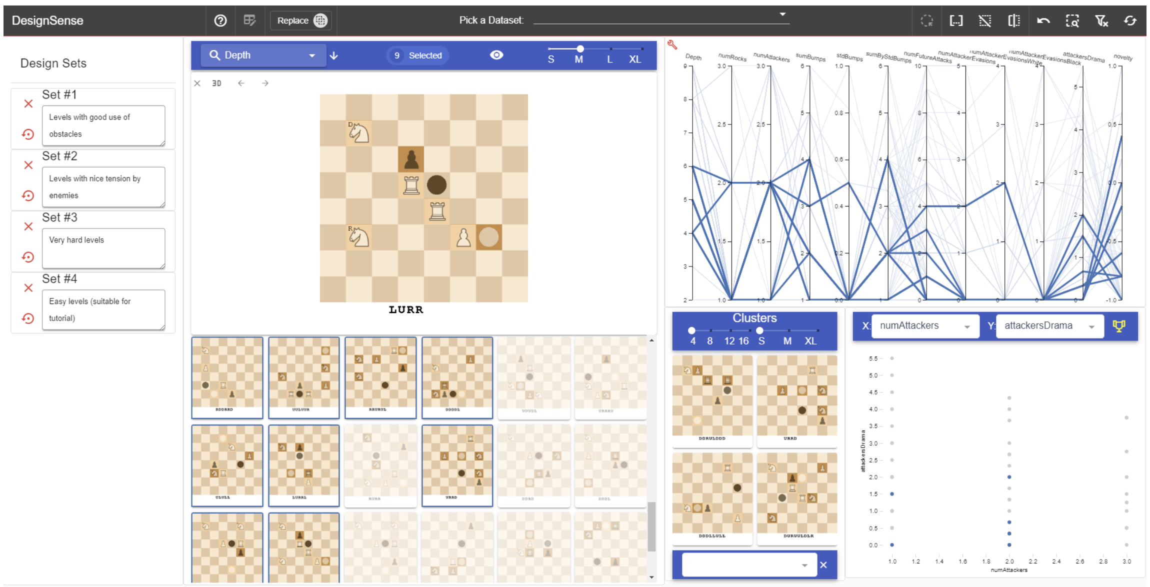

Fig. 1. DesignSense is an interface for exploring generated content through interactive data visualization. The interface enables exploring generated content through coordinated views that presents their quantitative metrics and visual appearance.

[**Accepted Manuscript at FDG 2020**]

Procedural content generation techniques can be used during game design to aid in exploration and expedite the content creation process. But this comes with the challenge of exploring a large quantity of generated content. This challenge is exacerbated by the lack of proper supporting tools. We present a visual analytics tool, DesignSense, developed as a response to a similar challenge in the architectural design domain. In a case study, we applied DesignSense to a dataset of generated levels for a puzzle game. The initial observations suggest a match between the task described and the features present in DesignSense with room for improvement with respect to its integration with the game development process.

CCS Concepts: • **Applied computing** → **Computer games**; • **Human-centered computing** → **Visual analytics**.

Additional Key Words and Phrases: procedural content generation, data visualization, design space exploration

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

Manuscript submitted to ACM

ACM Reference Format:
Ahmed M. Abuzuraiq, Osama Alsalman, and Halil Erhan. 2020. Shopping for Game levels: A Visual Analytics Approach to Exploring Procedurally Generated Content. In *Proceedings of FDG'20.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Procedural content generation (PCG) has a long history in games with goals such as compressing game content, improving game replayability, generation of personalized content, augmenting content creation, etc. [24]. Akin to other domains, game development have its own story with generative techniques. An essential side of this story relates to the nature of the relation between the game designers and the game content explored. This relation have been mediated via design metaphors [10] and captured in taxonomies [9, 25]. We are particularly interested in how game designers interact with the game content created using offline generative systems [25]. Regardless of the technique applied, designers face a large design space comprised of innumerable possible content alternatives. Any generative method, which is of course set up consistent with its potentials and limitations, can rapidly create hundreds, if not thousands, of alternatives. To avoid *sifting* through these alternatives, designers may be tempted to limit their exploration in the first place, which may defy the purpose of using generative methods at the outset. In such scenario, designers' cognitive system is overloaded with choices [5]. When lacking tools that can support navigating through and studying the created alternative contents, designers mainly rely on improvised or ad hoc methods such as random sampling, rapidly scanning images of the generated content in an image gallery, or curating content alternatives into sets or groups, etc.

We contend that there is a surpassing need for specifically design tools that aim to enable exploring a large number of generated content (alternatives). These tools should not compromise exploration, do ameliorate the choice overload, and respect the nature of design judgment (with its tacit and explicit dimensions).

We observe a similar challenge in the domain of architectural design where generative methods are used for Design Space Exploration. In this domain, the use of generative methods has a rich history starting from the early applications of rule-based systems and shape grammars [19] to the recent rise in using meta-heuristic optimization using parametric models [2] (or more generally Generative Design). Examples of Generative Design applications in architecture include the generation of exhibit halls [15], and sustainable buildings [7].

To explore and filter collections of generated designs in architecture, several design exploration tools were proposed [8, 13, 26]. These tools rely on interactive visualization techniques to support their tasks and emphasize representing and evaluating both the quantitative (e.g. performance) and the qualitative (e.g. form and aesthetics) aspects of design solutions.

Following arguments by [4, 10] on calling for tools that aid in understanding and interacting with procedural generators, and motivated by the commonalities we found between the two task environments (of PCG and Generative Design in architecture) we present a preliminary case study on applying a design exploration tool in exploring a dataset of procedurally generated game levels.

2 BACKGROUND

2.1 Representation and Evaluation of Game Content

An obvious approach to evaluate a piece of generated game content concerns with how well the content satisfies the designer's initial goals. For example, if the content is a game level, the designer may expect it to be fun and of appropriate difficulty. This can be measured mainly by play-testing the levels within the game context. Playing every single generated level can be labor intensive and often redundant, in particular when many of those levels differ slightly from each other. Instead, a second level of evaluation can depend on an illustrative visual representation of the generated content. For example, this can be a top-down image of a dungeon layout (possibly with annotations). Designers can use those to evaluate a generated content relying on their experience and acquired intuition.

But exploring large quantities of even static visuals remains a challenge without analytical assistance. This can be accomplished by using quantitative metrics that capture certain aspects of the structure or behaviour of those pieces. The use of metrics alone can be shortsighted as they inherently abstract away important qualities of the content alternatives. Instead those metrics can be used to sort or filter the aforementioned visuals. The use of analytical assistance via metrics appears in different ways in the literature of PCG. A quick glance shows proposing generic metrics [12] or employing metrics during generation (e.g. embedded in evaluation functions [25]) or as a guide in mixed initiative interfaces [3, 11].

2.2 Expressive Range Analysis

Generation can take place either as a part of the design process or during the runtime of a game. This distinction is referred to as online and offline generation respectively [25]. Techniques such as expressive range analysis (ERA) have been used in evaluating both online and offline generators. ERA starts by quantifying individual pieces of content via computed metrics and proceeds by visualizing chosen metrics in two-dimensional heatmaps [18] or density contours [22]. The merit of a generator stems in no small part from the variety in its generative space. Heatmaps are effective at the task of identifying and comparing data distributions. This is important when evaluating and iterating on a generator or when comparing multiple generators to the same PCG problem [22].

2.3 Interactive Visualizations for Selection

Data in its raw form is hard to comprehend. By representing data graphically, we start capitalizing more on the ability of our perceptual system to recognize visual patterns. Each different graphical representation (visualization) is suited for different analytical tasks. From a visualization design perspective, the heatmaps employed by the ERA are not as effective as other techniques at identifying outliers and correlations or comparing individual points (e.g. compared to scatterplot). They are also not suitable for visualizing and filtering multidimensional data (e.g. compared to parallel coordinates plots [14]). So for the task of selecting the 'best' pieces from a larger pool, different visualization techniques are more suitable.

2.4 Visualization of Design Spaces

As we described in Section 2.1, to support the qualitative evaluation of game content, we can provide static images each corresponding to a different piece of content. We can examine those by placing them into a navigatable gallery format.

To foster selection based on both the quantitative and qualitative aspects of game content we can imagine a dashboard where both an image gallery and a suitable visualization of the quantitative metrics are present. They can then be coordinated such that an interaction on any of them directly influences the other. For example, selecting the platformer levels with high linearity using a scatterplot will only show the static images corresponding to those levels. We can also select levels based on their visuals in an image gallery which highlights their corresponding points on a scatterplot.

When working with many metrics, we need to pick a visualization technique that can visualize multidimensional data. A common choice in design space exploration tools is the parallel coordinates plot (PCP). In practice the PCP is used to provide an overview over the data dimensions. It is also commonly used for filtering and identifying outliers [14].

The literature on design space exploration includes a number of examples for tools combining the elements described above. Recent examples include: DreamLens [13], Design Space Explorer [8], and the DesignExplorer [20].

3 CASE STUDY

To explore the application of design space exploration tools in PCG we used DesignSense which is an under development interactive visualization dashboard. DesignSense was designed by learning from the gaps identified in the design space exploration literature [1] in addition to studies on how architectural designers explore a large number of design alternatives [17]. Its techniques are based on the field of Visual Analytics [23], which combines both interactive visualization and automated analysis. We, the authors, collaboratively generated a dataset to visualize and experiment as part of the case study. Each item in the dataset is a different level in a turn-based puzzle game[1]. One of the authors is the lead designer and developer for the game and so had the full capacity to judge the generated levels.

3.1 The Problem

The mechanics of the game revolves around moving a white and a black board pieces, that always move in opposite directions to their respective final targets while avoiding moving enemy pieces. Finally, the player can choose to 'bump' one of the pieces with walls or obstacles so as to freeze it while moving the other piece freely in the opposite direction. The solution comprises of a sequence of moves from the start to the end for one of the pieces (e.g. left-up-right-right).

3.2 Levels Generation

The creation of the dataset was done as follows: first a few hand designed levels were created. By analyzing these levels a few metrics were identified which signified an 'interesting use of the mechanics' (e.g. more use of the bumping mechanism), whether they are solvable, and the number of moves in their solution (which gave an estimate of their difficulty). They also included the quantity and type of obstacles and enemies in the level.

An Evolutionary Strategy (ES) algorithm (later turned into NSGA-II) was first used to generate different puzzles by varying the initial positions of the pieces (black, white, targets, enemies, obstacles) and evaluating each according to a weighted sum of the identified metrics. By continuously assessed the outputs by sampling few levels and playtesting them. We continually returned back to the fitness function, adding/refining the metrics and refining the hyperparamters of the search algorithm. Finally, and to explore the thesis in this paper, we exported images of the puzzles' layout after each move in their solution. This was intended as a way to quickly run through the levels without running them in the game engine (Unity). The final dataset had roughly 70 levels.

3.3 DesignSense: A Design Exploration Tool

DesignSense is an interactive data visualization dashboard. Its interface is composed of a parallel coordinates plot and scatterplot as abstract data visualizations (e.g. Fig 2-A), a visuals (game level image) gallery (Fig 2-B) and exploration support components like curated Design Sets (Fig 2-D) and automated clustering.

Selection: The main goal of DesignSense to to enable selecting a subset of items from a larger collection of them. This is accomplished through filtering by data dimensions (Fig 2-A) as well as by interacting with visual representations of those items (Fig 2-B). When items are selected through the gallery they also get selected automatically in the other visualizations (and vice versa). This is an example of a Coordinated Multiple Views technique [16]. The premise is that

[1]Unpublished by the time of writing.

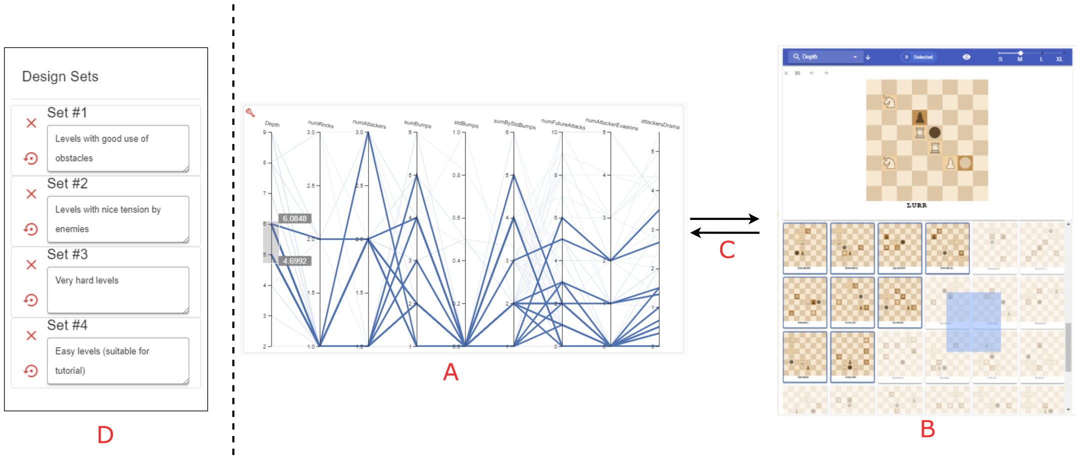

Fig. 2. **A:** The parallel coordinates plot (PCP). Each game level is represented by a line and each axis a metric. Users can select levels by selecting ranges on the chosen metric's axis. **B:** The visuals gallery. At its bottom is a scrollable grid. Levels can be selected dragging a box around their static images. Selected levels are highlighted. At the top is a larger image of the currently inspected level. Arrow buttons can be used to change to the next/previous step in solving the puzzle. **C:** The gallery and the PCP are coordinated such that selecting levels on one automatically select them on the other. **D:** the current selection can be saved to a set or be replaced with the content of a set.

no single view on the dataset is sufficient for all purposes and so instead we rely on multiple views that coordinate while showing different aspects of the data. When a new selection is made (e.g. by dragging a box around data ranges as in Fig 2-A) it is integrated into the current selection of items by replacing it, adding the new to the old or intersecting them. This makes it possible to fine-tune the selection of items in a plethora of ways.

Inspection The best measure of the quality of a piece of content stems from inspecting it closely or play-testing it (as we described in Section 2.1). Any piece of content can be inspected by hovering over its representation on the gallery or the data visualizations. When an item is inspected then its visual appearance will be shown at the top of the gallery (Fig 2-B). From there it can currently be examined closely via a series of images. The dataset in this case study contains turn-based puzzle levels and the series of images present a stepping through its solution.

Saving to Sets When a group of levels are selected (whether because of their metrics or visuals or both) they can be saved in a set and recalled back when needed. Sets can also have comments and titles that describe what they are about.

3.4 The Exploration Process

The case study started by presenting the process of generating the dataset and the final set of metrics with the intuition behind each. Next, a matrix of heatmaps (not part of DesignSense) were used to visualize the expressivity of the generator. The designer noted that in order to pick the best metrics to visualize, he first needs to study some good levels. This way, he could get an intuition about their metrics and learn how they relate to the 'goodness' of the levels. This lead us to DesignSense –which was intended to help in just that– before revisiting the expressive range analysis again.

First, the main features of the system were presented upon which a talk-aloud run through the dataset was initiated with interleaving discussions. The main bulk of the run followed this pattern: 1) out of the 15 metrics available in the dataset, the designer limited those on the parallel coordinates plot to 9. He noted that he would favour only using those initially until he understood what they meant before adding more metrics. This was followed by filtering the levels down based on one of the metrics (number of obstacles, a non-behavioural metric) by highlighting a chosen range on the axis representing that metric in the parallel coordinates plot. This was followed by visually inspecting the selected levels through the images gallery. Upon finding an interesting level (visually and possibly data-wise), that level was played through by moving through the sequence of images showing the steps to solve it. After inspecting a level, the designer placed it in a set for future reference. The designer hypothesized that he would use the sets in the interface to categorize the interesting levels based on their difficulty or common dominant features (gameplay-wise).

A discussion about the metrics used lead us to the realization that the ones used were all aggregated summaries and that sometimes a more detailed reporting might be useful. For example, the number of moves that can be made at each step without touching an enemy, or in other words, the *branching factor* after each move. Finally the designer also argued that the relative difference between the metrics mattered more than how well they approximated an 'absolute goodness' and that he would be interested in the metrics up to the point where they lead to good levels and not any more afterwards (except for the purpose of learning more about what they meant).

4 DISCUSSION

Section 2.1 argues on the important of using illustrative visuals for making sense of the generated content. We propose that they should ideally allow comparing and evaluating the content at a glance. A grid gallery (Fig 2-B) is as effective as the visuals presented on it. After the case study, we recommend to support multiple means for inspecting a level in addition to moving manually through a sequence of images. These may include animated images (GIFs) or loading the inspected level into the game engine where it can be play-tested. To complete a full cycle design process, the designer in the case study suggested a system feature to enable exporting the composed sets of content to folders in the game project. The last two suggestions are similar in their emphasis on tying DesignSense with the game design process.

The process we is similar to the one described by [21] as *design-by-shopping* where individual content pieces are generated, evaluated and then filtered down with the support of interactive visualizations. The advantages of *design-by-shopping approach* includes the following. First, as a generic approach it does not assume a specific type of content. Second, the interactive visualizations can enable deriving crucial insight about game content through its data, e.g. the relation between input parameters and metrics, among metrics, or between metrics and the resulting visuals. Third, the approach presented in the case study with DesignSense provides a substitute for the common practice of exploring generative spaces via tweaking parameters which suffer from change blindness [6].

5 CONCLUSION AND FUTURE WORK

Exploring a large collection of generated game content can be challenging without the proper support tools. We present a visual analytics tool, DesignSense, developed as a response to a similar challenge in the architectural design domain. In a case study, we applied DesignSense to a dataset of generated levels for a puzzle game. The initial observations suggest a match between the task described and the features present in DesignSense with room for improvement with respect to its integration with the game development process.

In the future, we intend to experiment with a closer integration between DesignSense and a game development environment, such as Unity. We would like to also extend our evaluation of this approach to other PCG problems and to bring it a wider audience.

ACKNOWLEDGMENTS

REFERENCES

[1] Ahmed M. Abuzuraiq and Halil Erhan. 2020. The Many Faces of Similarity: A Visual Analytics Approach for Design Space Simplification. In *25th CAADRIA proceedings*.

[2] Yassin Ashour and Branko Kolarevic. 2015. Optimizing Creatively in Multi-objective Optimization. In *Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD '15)*. Society for Computer Simulation International, San Diego, CA, USA, 128–135.

[3] Alexander Baldwin, Steve Dahlskog, Jose M Font, and Johan Holmberg. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 25–32.

[4] Michael V. Cook, Simon Colton, Jeremy Gow, and Gillian Smith. 2019. General Analytical Techniques For Parameter-Based Procedural Content Generators. *2019 IEEE Conference on Games (CoG)* (2019), 1–8.

[5] Halil Erhan, Janelynn Chan, Gilbert Fung, Naghmi Shireen, and Ivy Wang. 2017. Understanding Cognitive Overload in Generative Design: An Epistemic Action Analysis. In *Proceedings of the 22nd CAADRIA*. 127–136.

[6] Halil Erhan, Nahal Salmasi, and Rob Woodbury. 2010. ViSA: A Parametric Design Modeling Method to Enhance Visual Sensitivity Control and Analysis. *International Journal of Architectural Computing* 8, 4 (2010), 461–483.

[7] Ralph Evins. 2013. A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews* 22 (June 2013), 230–245. https://doi.org/10.1016/j.rser.2013.02.004

[8] Ekaterina Fuchkina, Sven Schneider, Sven Bertel, and Iuliia Osintseva. 2018. Design Space Exploration Framework-A modular approach to flexibly explore large sets of design variants of parametric models within a single environment. In *Proceedings of the 36th eCAADe*. CUMINCAD, 367–376.

[9] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–22. Publisher: ACM New York, NY, USA.

[10] Rilla Khaled, Mark J Nelson, and Pippin Barr. 2013. Design metaphors for procedural content generation in games. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1509–1518.

[11] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2013. Sentient sketchbook: computer-assisted game level authoring. (2013).

[12] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2013. Towards a generic method of evaluating game levels. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[13] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. 2018. Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. In *Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 369:1–369:12.

[14] Tamara Munzner. 2014. *Visualization analysis and design.* CRC press.

[15] Danil Nagy, Lorenzo Villaggi, Dale Zhao, and David Benjamin. 2017. Beyond Heuristics: A Novel Design Space Model for Generative Space Planning in Architecture. In *Disciplines and Disruption - Proceedings Catalog of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2017*. CUMINCAD.

[16] Jonathan C Roberts. 2007. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*. IEEE, 61–71.

[17] Naghmi Shireen, Halil Erhan, Robert Woodbury, and Ivy Wang. 2017. Making Sense of Design Space. In *Computer-Aided Architectural Design. Future Trajectories*, Gülen Çağdaş, Mine Özkar, Leman Figen Gül, and Ethem Gürer (Eds.). Springer Singapore, Singapore, 191–211.

[18] Gillian Smith and Jim Whitehead. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. 1–7.

[19] George Stiny. 1980. Introduction to Shape and Shape Grammars. *Environment and Planning B: Planning and Design* 7, 3 (1980), 343–351.

[20] Thornton Tomasetti CORE Studio. 2019. Design Explorer 2. https://github.com/tt-acm/DesignExplorer

[21] Gary M. Stump, Mike Yukish, Timothy W. Simpson, and E. Nathan Harris. 2003. *Design Space Visualization and Its Application to a Design by Shopping Paradigm.* International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. Volume 2: 29th Design Automation Conference, Parts A and B. https://doi.org/10.1115/DETC2003/DAC-48785

[22] Adam Summerville. 2018. Expanding Expressive Range: Evaluation Methodologies for Procedural Content Generation. In *AIIDE*.

[23] James J. Thomas and Kristin A. Cook. 2006. A Visual Analytics Agenda. *IEEE Comput. Graph. Appl.* 26, 1 (Jan. 2006), 10–13.

[24] Julian Togelius, Noor Shaker, and Mark J. Nelson. 2016. Introduction. In *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Noor Shaker, Julian Togelius, and Mark J. Nelson (Eds.). Springer, 1–15.

[25] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 172–186. Publisher: IEEE.

[26] Jamal van Kastel. 2018. *Visual Analytics for Generative Design Exploration: An interactive 3D data envirionment for a computational design system facilitating the performance-driven design process of a nearly Zero-Energy sports hall.* Master's thesis. TUDelft. http://resolver.tudelft.nl/uuid:ad6f454b-0e67-4664-88d4-87d2132a1f71