Computer assisted brainstorming

Which ideation tool is best and what makes it best?

Xander Bos
Graduation Thesis, August 2016
Media Technology MSc program, Leiden University
Supervisors: Maarten H. Lamers & Peter van der Putten
xander.bos@gmail.com

***Abstract*—The purpose of this study is to determine which ideation software, referred to as tools during the paper, is the best one available and why this is the case. After analyzing 83 different tools, a selection of four tools is made. A fifth tool is developed specifically for this research, with the goal of testing the effectiveness of leveraging a dynamic type of word database, as the four selected tools all rely on static types of word databases. An experiment is then performed with 31 participants, using a specifically designed method for this study and a method based on four well-known characteristics that describe the quality of ideas created through brainstorming methods. The results of the experiment show that a static type of database, focused on specific category, in combination with random selection offer the best ideas.**

***Index Terms*—Creativity, ideation, brainstorming software**

I. Introduction

In this section the main problem handled in this research is introduced, followed by some interesting attempts of solving this problem – which are not satisfactory. The section concludes by providing the research question and a short description about the rest of the paper.

A. Origin of the problem

Generating new ideas can be a challenging task for people, especially when under pressure (Amabile, 2002). This problem forms the base of this research. The process of generating new ideas is known as ideation. A simple online search for ideation software, which will be referred to as tools during this research, provides a vast amount of tools that claim to resolve this problem, but do they really? They mainly come in two different forms, either software that helps structuring brainstorming sessions or idea generators.

The tools that structure brainstorming sessions nearly always advertise heavily that they're the best ideation tools available. Their reasoning being that by removing distracting parts, such as organizing all the ideas, people have more time to actually focus on idea generation. The main problem here is that it still requires people to be creative themselves and the tool doesn't actually assist in the most important part – actually generating ideas. Such tools also require relatively large groups of people, which isn't always ideal. Some situations can be more about individual ideation.

The tools that actually generate ideas entails multiple types of generators. They can generate, for example, titles, stories, game ideas or project ideas. One thing is consistent across all these types of generators, which is that they combine purely randomly selected words using a pre-defined structure. This could certainly give you inspiration, but it is also possible that most of the generated results aren't particularly useful.

B. Interesting attempts at solving the problem

An interesting method to overcome this problem is implemented by a program called ParaMind (ParaMind.Net, 1992)[1]. It works by using a database where all words are connected to other words with similar/overlapping meanings. This method allows the program to be associative, a key element for brainstorming. The associative idea generation allows for divergent thinking. Divergent thinking is about generating a wide variety of ideas for a certain topic or problem (Guilford, 1969). ParaMind works by letting you enter a piece of text or a sentence about what you want to generate ideas about. When the tool starts creating associations, new perspectives or insights might be found. ParaMind was the first tool to implement this and to the best of our knowledge, still is the only one. The only other tool that comes close to ParaMind, is called IdeaFisher (Thoughtrod, 2005). It isn't quite on the same level however. The reason behind this is explained later, in section II.B.

There are some well-known experiments that generate ideas, but these are usually very specific and complex. For example, BRUTUS (Bringsjord et al. 1990) is story generator and Watson (Ferrucci et al. 2010) is an artificial intelligence that initially started by being very successful in an American television game show known as "*Jeopardy!*". Later, Watson was used to create new cooking recipes, with the results now bundled in a book (IBM et al. 2015). While these are very impressive, they don't really fit well within the context of this research. The main reason is that they output final products and not just ideas.

C. Research question

During a global analysis of the most popular tools available, a certain pattern became visible. All tools have a

[1] All tools mentioned in this study are listed in Appendix A, with publisher and web location.

static type of database. There isn't a single tool available that utilizes newer technologies to improve on this, like semantic web[2] (Berners-Lee, 2001 and Feigenbaum, 2007). The inner workings of Wikipedia are an example of semantic web. Whenever someone writes a page of information, after saving, Wikipedia automatically find certain words and links them to other pages for more information. When using a smart implementation of this principle, it would be possible to find associations from a certain starting point. This could be used to construct a dynamic type of word database, where related words are automatically found online.

Based on the findings regarding the proposed problem, a main research question can be constructed. *Which is the best ideation tool, that only needs one individual to operate it, available and what makes it the best?* Additionally, there were no programs found during the global analysis (see section II for more information) that implement a dynamic database. Because of this, a second goal for this research was formulated: *To create and include in our comparison, a tool that uses a dynamic word database.*

In the next section, more information about the relevance of this research is given. The global analysis is explained in more detail here as well, in which also the criteria for the best ideation tool is set. The section ends by mentioning certain researches, from which their conclusions could be useful in this research. After the next section concludes, the tool that will build for this research is explained in more detail, followed by the methodology, results, discussion and conclusion, and finally the future works.

II. Scientific context & Related work

The relevance of this research is explained in this section, both by providing existing research and by the global tool analysis. The section is concluded by exploring options for implementing a dynamic database and designing the experiment.

A. Types of tools and their limitations

In the introduction, the first type of ideation tools mentioned helps structuring the brainstorming sessions and thus improving creativity since secondary tasks are automated/simplified. When looking more in-depth at these applications, it is noticeable that the way they work is inspired by Guildford's definitions of human thinking when problem solving (Guildford 1967). Guildford identifies two types of thinking, called divergent and convergent thinking. Divergent thinking, as also is mentioned in the introduction, is essentially generating as many ideas as possible and convergent thinking is selecting the best idea from that collection of ideas. Usually the selection is done by setting clear criteria. These definitions are incorporated in these tools by giving the user the means to easily manage the ideas generated.

It is definitely possible that using an approach like this to assist in brainstorming will work, but it requires the participants of a brainstorm session to be creative themselves. According to Goldenberg et al. (2002) people are naturally creative, but this creativity is limited because of our routines and pace of life. He also mentions that because brainstorming is so popular, it is also researched quite a bit. Some interesting findings were that, according to some of these researches, brainstorming in groups doesn't yield more ideas than when people are brainstorming individually – but actually generate less ideas per person. The reasons for this are all related to the way in which people interact with each other during brainstorming sessions.

When reflecting on the research question, this means that a tool that only one individual can use could improve the amount of ideas generated by an individual even more. Another conclusion that can be drawn is that the focus of this research lies with divergent thinking and that it may help getting an individual into a more creative flow, solving the limitation caused by our routines, etc. Other methods of generating ideas mentioned by Goldenberg also appear to not be very effective. Duch (2013) mentions in a paper about computational creativity that when adding structure to the generation process, based on higher-order rules and templates, it significantly improves the quality of the generated ideas.

According to Duch, an experiment was done which compared computer generated ideas using templates with human generated ideas without templates. Based on this experiment, it was concluded that computer generated ideas scored much higher on creativity and originality. Apparently the associative process during this test was guided by a set of general rules. These general rules are the so-called higher-order rules. According to Yao et al. (2007), such rules are related to relational learning. For example, the relation of a certain object to a set of other objects – which is basically association, but determined association and not learned associations.

B. Global tool analysis

During the global analysis mentioned in the introduction, 83 different tools that promote themselves as ideation tools, were analyzed (April, 2016). Refer for the complete list to Appendix A. The tools that are not mentioned in the next tables are mainly tools that focus on idea management instead of generation, which is outside of the context of this research. For a tool to be useful in context of this research, it needs to pass the following criteria: *the tool must be able to generate something that is either an actual idea or something that inspires an idea, in the form of at least a short sentence.*

The tools shown in Table 1 have interesting aspects, but in their current state they don't pass the set criteria. Their interesting aspects could possibly be applied in future software.

Name	Interesting aspects
Yutongo	Uses *smart/simple questions* to assist in the process of idea generation, making it easier to come up with ideas.
IdeaScale	During the idea submitting process, ideas are automatically *merged* to create new ones using *predictive word search.*
Crowdicity	Uses trend analyzing, which means that it searches for topics that interest people the most and which ideas are the most popular.

[2] https://www.w3.org/standards/semanticweb/

IdeaFisher	Has associative thinking means, coupled with a large database of words, images and links. The problem is that the user has to manually select the discovered associations for the given input. In that sense, it doesn't generate ideas, only looks for words it can associate with.
Visualthesaurus	Finds associations for words by using synonyms and figure of speech.

Table 1 – Interesting tools, that nonetheless do not pass the set criteria

The tools shown in Table 2 are suited for comparison with other tools, since they pass for the set criteria. Note that the Orteil's Game Generator and Seventh Sanctum are nearly identical, so only Orteil's Game Generator is used during the experiment.

Additionally, ManyThings Sentence Generator is added. This tool doesn't match the criteria, since it generates basic sentences instead of actual ideas. The reason that ManyThings Sentence Generator is added, is that it would be interesting to test whether people prefer these basic sentences over ideas generated by the other selected tools.

Name	Description
ParaMind	Uses a smart database construction to find associations with other words, starting from a user provided set of words – which should have tools to make it fully automatic after giving it a starting point.
Portent Title Generator	Not directly an idea generator, but generates titles based on terms which could lead to new ideas – also matches the kind of output we would expect; a short sentence that describes something.
Orteil's Game Generator	A rather simple online tool, but constructs game ideas by mixing up random words/part of sentences which works quite nicely.
Seventh Sanctum	Also online and works exactly the same as Orteil's Game Generator.
ManyThings Sentence Generator	Generates very simple sentences that has the following elements: subject, verb and object.

Table 2 – Proper tools, that pass the set criteria

Reflecting on the results of the global analysis, excluding the tools in Table 2, no tool was encountered that implements solutions to the findings mentioned earlier. Most of the available tools still assume that brainstorming should take place in groups, which apparently has its flaws. They can potentially be a lot more efficient, by implementing elements that allow individuals to generate more and/or better ideas.

The tools in Table 2 do have some of these elements implemented, since they allow individual use and work by using association and/or structure. The tool that will be built for this research will use a combination of these elements along with the dynamic database mentioned earlier.

C. Semantic web usage proposals

It is rather strange that in the large list of analyzed tools there is not one single tool that actually implements modern technologies, such as semantic web. Even with a direct search, only some proposals for implementations can be found. In these proposals, semantic web is not meant to help in the idea generation part directly, but rather in information structuring. An interesting method is shown by Lorenzo et al. (2011). He proposes to keep a large database of ideas, which is structured by semantic web technology. When someone thinks of an idea, it is stored in the database using semantic web protocols like RDF[3]. When someone else thinks of a similar idea, the idea can be queried using SPARQL[4] – a special query language for semantic web databases. This way, similar ideas can be retrieved quickly, which can then be used to determine things like originality of the idea or getting new insights.

Alternatively, a proposal was given by Angelo et al. (2013) in which the semantic web is used as a vocabulary in a more research-like context. The idea is that semantic web assists participants of brainstorm sessions by providing definitions of certain words in ideas. It can find particular words and link them to information sources expanding on that word, like a dictionary.

Although these are interesting approaches, it is curious that nobody has tried to implement this for actual idea generation. Instead of using semantic web to structure information by a system, why not use it to get information from already existing systems like Wikipedia? For example, when describing a new idea in a sentence or two, you can use semantic web to find information about keywords in that description. Distilling information, using some kind of algorithm, from these sources can provide a way to create unusual associations, which in turn may lead to "divergent thinking". It may be an unconventional application of semantic web, but it may work very well for idea generation.

D. Testing evaluation methods

To properly determine which of the selected tools are the best, a proper way of validating the results/ideas needs to be found. Guilford (1967) determined that divergent thinking involves four characteristics: fluency, flexibility, originality and elaboration. Alongside this, he designed an experiment to test creativity called the Alternate Uses Test. The goal during this test is to find as many alternative uses possible for a certain object, e.g. a paperclip. It can be used to determine how well certain brainstorming techniques work, as an exercise to get into brainstorming or, in this case, to test how well a tool can generate useful ideas. In light of the Alternate Uses Test, the four characteristics can be explained as follows:

- Fluency: the number of alternative uses you can think of;
- Flexibility: how many different categories are covered by the alternative uses;
- Originality – how unusual or unique the alternative uses are;
- Elaboration – how clear and usable the alternative use ideas are (e.g. can they be worked with)

The above method is quite well-known and could possibly be used to test our selection of tools, but there are also

[3] https://www.w3.org/RDF/
[4] https://www.w3.org/TR/rdf-sparql-query/

alternative views on how to test the quality of generated ideas. Reinig et al. (2006), for example, states that researchers normally use three different methods to determine idea quality. Namely “sum of quality scores”, “average quality score” and “count of good ideas”. The scores per idea are given on a 4 point or 5 point scale. Then you use one of the three methods to calculate a final score. According to them, the first two methods are the most subject to bias and thus only the last method should ever be used.

This method doesn’t seem very promising, since giving scores to ideas using a 4 or 5 point scale is in itself very subject to bias as there is no further foundation on what these scores are based on. So for this research, Guilford’s method is favorable.

III. SubMind

The tool that is made specifically for this research is explained in this section. First a simple explanation is given of what it does, followed by a more in-depth explanation of its inner workings.

A. Tool description

As mentioned earlier, existing tools normally have a static type of database. Additionally, if they feature any kind of idea generation it is of the random selection type. Only ParaMind has the functionality of automatically generating ideas using association, according to the global analysis.

The goal of SubMind, which is the name of the tool created specifically for this research, is to determine whether a dynamic type of database in combination with association can improve the quality of ideas generated.

The database of SubMind is dynamic in the sense that the word database will be built based on the input given, which will be in the form of a short sentence. Nouns, or rather keywords, in the sentence are extracted and are then used for the association process. The main difference with ParaMind is that the association process is done using online (semantic web) resources, while ParaMind has a static database for this process. After the process is finished, a set of ideas will be generated using the associations found. This means that the associative aspect of SubMind is seen in its database construction, while actual idea generation is performed using random selection of words in the constructed database.

For every usage, a new database will be constructed. This way it is more likely that the words in the database are somehow related to the original input.

B. Main application flow

Refer to Appendix B for a flowchart of the main application. For starters, a good online information source needs to be determined. Ideally, it is a public API that provides access to a large database with varied information. Wikipedia would come to mind, but that isn’t properly usable for SubMind or any type of software. The reason is that the information on Wikipedia isn’t structured in a way to be usable for a computer. DBpedia[5] provides a solution for this. It has all the information of Wikipedia structured using semantic web protocols like RDF. This allows SubMind to find information about the input it has been given, which in turn allows for finding associations.

SubMind will start off from an input sentence provided by the user. Using NLTK[6], a library for Python, the sentence is split into parts. Each part is then tagged using word-parts, e.g. nouns, verbs, and adverbs. For SubMind, only the nouns are extracted and these are merged into a single “sentence”. This is required for the next step, sending a request to the DBpedia Spotlight API. DBpedia Spotlight can be used to find DBpedia information resources in text.

The response given by the API contains one or more sets of data, which will be referred to as objects. Each object contains both the link to the DBpedia resource and the word associated with this resource. The word is stored in the database for idea generation, while the resource link is stored to recursively continue the associative process. The input provided by SubMind to the API is the merged sentence mentioned earlier. For every object found in this sentence a search sequence is performed, which will be explained in section III.C. When there are no objects returned by the API, SubMind terminates.

If a search sequence is completed, it should ideally return at least one object that contains a new resource link. If the resource link has already been used before, the object is rejected to prevent duplicate values in the database. Like before, the word in the object is stored and the resource link is used to start a new search sequence. An additional step is added at this point however. Every time a search sequence returns data, the database size is measured. If the database contains 200 words (or more), no new search sequences are initiated. SubMind will wait for still running sequences to finish. When all sequences are finished, the database construction is completed.

The database size of 200 was determined by experimenting with the prototype of SubMind. The goal was to have good variation in the ideas generated, while not having to wait for too long for SubMind to finish. A single search sequence can take quite a long time; up to 15 to 30 seconds. On average it takes about 1 to 2 minutes for SubMind to complete its task, which is acceptable given its intended use.

The next step is the construction of the ideas. As mentioned, all newly associated words are still linked in the database to their originating keyword. For each originating keyword found in the input sentence provided, a random word associated to it is selected from the database. The original keyword is then replaced with the randomly selected word. This process is repeated 25 times, to effectively generate 25 ideas. This number is used because that is the amount of ideas needed for the experiment

C. Search sequence flow

Refer to Appendix C for a flowchart of the search sequence. It requires a resource link to start, which is provided by the DBpedia Spotlight request. The resource link is used to perform a SPARQL query. The SPARQL query extracts the

[5] http://wiki.dbpedia.org/

[6] http://www.nltk.org/

abstract and comment elements from the DBpedia resource. These two elements were selected, since they give an explanation about the resource and are always present in each resource. Quite a few elements are not always available, or are typically used to link to other resources, e.g. elements relating to the subject or category of the resource. Just looking at these two elements keeps things easy to work with and time efficient to implement.

The abstract and comments elements are merged into one block of text. Then, the same pattern is followed as mentioned before. NLTK extracts all nouns from the text, merges them into a new "sentence" and requests new resources from DBpedia Spotlight using this sentence. The objects returned are passed to the main application flow and the sequence is finished.

IV. Experimental methodology

This section is about the process of designing the experiment. Certain problems were foreseen and explained, information about the idea generation process is given and an extensive explanation of how the experiment is structured, tested and distributed is provided.

A. Foreseen problems

As described in the last part of section II, applying the Alternate Uses Test is a possible method of testing the idea quality of the tools. It does introduce a problem however, namely the fact that the selected tools for comparison have different types of input. ParaMind has an open input, which means you can enter just one word, a combination of unrelated words or a sentence. Portent Title Generator only allows input for one single word and both Orteil's Game Generator and ManyThings Sentence Generator don't have any type of input at all. So how can these last three tools be used with the Alternative Uses Test?

One option would be to recreate the tools in such a way that they are usable with the Alternative Uses Test. This solution is much too time intensive. Their functionality is quite simple, so it shouldn't take much time to recreate the tools themselves. Creating a proper database for them is a lot more difficult however. And then it wouldn't be an actual tool comparison.

A far simpler solution is to not use the Alternative Uses Test, while still using the divergent thinking characteristics named flexibility, fluency, originality and elaboration. Participants can grade the ideas based on these characteristics using a method that suits it best. More about this in section IV.C.

B. Idea generation

The best way to test these tools is to use videogames as a knowledge domain, with the main reason being that Orteil's Game Generator is limited to this "field" of idea generation.

Portent Title Generator will be given the word "videogames" as its input, since that is the context of the ideas generated. The associative tools will be supplied with the sentence "A video game about a creature living in an environment". This sentence has not any real meaning by itself, although it does describe a videogame in a very abstract way while still providing a few keywords that can be associated with.

It should be noted that other options for testing the tools were explored. For example, the option of using different inputs had been taken into consideration, especially for the associative tools. The reason why the current method is chosen, is mainly that it can be argued that the input constructed above is designed to be "vague" and thus does not provide the associative tools with any kind of advantage over the other tools.

All tools will generate a set of 25 ideas. The ideas are categorized by their respective tools and are numbered according to their original order when generated. The result of this will be referred to as an idea set. Each idea set consists of five lists, since there are five tools. In Appendix D an example is given of an idea set. Since the number of ideas generated is static, it is useless to test the fluency characteristic – so it is excluded from the experiment.

C. Experiment structure

The experiment starts off by giving participants a brief explanation about the research. Then, the first method of grading is explained.

For the first method of grading, the participant will go through their own unique idea sets and mark the first three good ideas per tool. The criteria for the "good" idea is: "A good idea that can be used as an element (e.g. as story subject or game mechanic) for a video game". It is left to participant's judgement how well an idea matches the set criteria. This data can be used to determine how many iterations of idea generation were needed by the tool before generating "good" ideas and will be used as an indicator how efficient the tool is. It will also give insight in how often "good" ideas occur in total, as it is possible that not always three ideas will be marked.

The explanation of the first method is directly followed by the idea set. It is positioned here, because the participant can then directly apply the first method of grading – which needs to be applied while reading the idea set. After the idea set, the second of method of grading is explained.

The second method of grading will be about grading each tool according to originality and elaboration. These are tested by providing participants with Likert scales. These will give a good indication whether the participants find the ideas both original and clear. It is placed after the lists to prevent an information overload for the participants. Presenting all methods of grading at once would have a confusing effect. The same reasoning goes for the placement of the third method of grading.

The third and final method of grading is about ordering the lists based on flexibility and usefulness in context of videogames. Both are ordered by going from least flexible/useful to most flexible/useful. During the testing phase of the experiment, an alternative method for grading flexibility was used. The participant had to categorize every idea on each list, which was problematic. In section IV.D., more information about this is given. This method should give insight in whether

participants find the ideas to be varied and useful. Analyzing the ordered data for this method may prove difficult, so two different methods will be used. In the conclusion, it will be discussed which of these methods are most suited for taking part in answering the research question.

Finally, the participant fills in some personal information, namely their age, gender and how often he/she plays videogames. This information will provide core information about the characteristics of the participants.

D. Testing the experiment

Before launching the experiment for the broader public, the experiment was performed in five closely monitored sessions. This was mainly done to determine whether the instructions were clear for the participants and how long it would take to finish the experiment. Ideally, the experiment would require 30 or less minutes to prevent wearing out the participant.

For these five sessions, a paper version of the experiment was made. Participants were asked to perform the experiment, while a stopwatch was used to measure time needed. Their overall behavior was also noted, such as getting agitated.

During these sessions, it became clear that there was a problem with the flexibility grading. As mentioned in section IV.C., participants had to categorize every single idea on all five lists according to the definition of flexibility. At minimum, it took the participants 60 minutes to finish grading flexibility alone. During the process, it was clear that participants became agitated. The reason was that the explanation for flexibility was too vague, and even when understood, it took a long time to categorize the ideas.

To resolve this problem, the same approach of grading usability was used for flexibility: ordering the lists based on flexibility. Additionally, an example (based on the Alternate Uses Test) was given for flexibility, to make it more understandable. The results that were gathered using the categorizing approach, could be mapped to the new ordering approach by counting the categories found for each list. More categories would translate to being more flexible. This way, the results from these five sessions wouldn't have to be discarded.

E. Experiment execution

For the experiment a minimum of 30 participants is set. This amount seems doable to achieve, while providing enough data to filter out possible anomalies. The experiment that is meant for the larger public was built in an online environment. This would allow to disseminate the experiment more easily, and would make reaching the 30 participations count a relatively simple task. In the online experiment, the participant gets concise instructions for every exercise. These instructions are based on the method of grading explanations given in section IV.C. Also, multiple checks are built-in, to prevent participants from making errors.

Below, for each method of grading will be explained what checks were implemented to prevent participants from making mistakes. Refer to Appendix E to see an example of the online experiment.

For the first method of grading, if participants would select more than three ideas in each list, the participants are notified by saying that three is the limit. An additional check is done at the point of submitting. If the participant hasn't checked anything in all lists, the participant is notified about this once. If they click the submit button again, submitting will continue regardless.

The second method of grading only requires a check at the moment of submitting. If the participant forgot to fill in one of the Likert scales, the participant is asked to resolve this.

The third method of grading has multiple checks. The participant can only use letter A to E, the rest is blocked. Each letter can only be given once. At the point of submitting, it is checked whether the ordered lists are complete. If not, the participant is asked to complete it.

The experiment was disseminated using multiple resources, such as WhatsApp groups, Facebook and forums, in combination with personal requests. Additionally, certain prices could be won by participating. This was implemented to motivate people to participate.

V. RESULTS

At the end of the experiment, a total of 31 responses were collected. Based on the answers given to the questions about the participants themselves, the following can be said: 71% of the participants were between the age of 21 to 30. 68% were male and 32% were female. 74% of the participants play videogames on at least a weekly basis.

For each method of grading, the results will be put into a separate table, followed by an explanation by how the results were determined and some observations. The best values are marked by a bold font.

A. First method results

	Total marked ideas	*Avg. 1st marked idea*	*Avg. 2nd marked idea*	*Avg. 3rd marked idea*	*Avg. overall*
Orteil's Game Generator	**93**	**3.7**	**8.7**	**15.0**	**9.2**
SubMind	60	6.1	12.4	19.2	10.5
ParaMind	58	4.7	14.2	20.2	10.2
Portent Title Generator	71	6.3	13.0	17.1	11.0
ManyThings Sentence Generator	70	5.0	11.9	16.2	9.9

Table 3 – Results from the first method of grading

The second column of Table 3 contains the total number of marked ideas, for each list, across all participations. The participant was required to at mark at most three ideas they thought were a "good" video game idea per list. The maximum value is 93, since 31 participations times 3 ideas equals 93. This means that the higher the value, the better. Orteil's Game Generator is the only tool that has reached this number.

Column three in Table 3 represents how many iterations were required on average before the first "good" idea on a list was found. The values are calculated by the sum of the idea numbers (ideas are numbers from 1 to 25), divided by the count of how many times a first "good" idea is marked across

all participations for that list. The lower the value, the better. Column four and five follow the same principle, but focus on the second or third marked idea respectively. Orteil's Game Generator scores best in all three columns.

The final column of Table 3 represents how many iterations on average were needed to get to a "good" idea, for each list. The lower the value, the better. Orteil's Game Generator scores the highest.

B. Second method results

	Avg. originality score	*Avg. elaboration score*
Orteil's Game Generator	**3.68**	**4.1**
SubMind	3.65	2.5
ParaMind	2.4	2.0
Portent Title Generator	2.9	3.0
ManyThings Sentence Generator	2.7	2.9

Table 4 – Results from the second method of grading

Table 4 contains the average scores for both the originality and elaboration score. These values were measured by using Likert scales, and so calculating the mean would be the best method of analyzing. Note that for originality, both Orteil's Game Generator and SubMind have a score with two digits. This was done to determine a clear "winner", since the scores are very close. For originality, Orteil's Game Generator scores the highest, but SubMind is very close as well. For elaboration, Orteil's Game Generator has a clear advantage over the other tools.

C. Third method results

	Point-based flexibility score	*Point-based usefulness score*	*1st place flexibility counter*	*1st place usefulness counter*
Orteil's Game Generator	94	92	7	**7**
SubMind	82	89	2	**7**
ParaMind	93	93	9	4
Portent Title Generator	**107**	**101**	**10**	**7**
ManyThings Sentence Generator	89	90	3	6

Table 5 – Results from the third method of grading

The results in the second and third column of Table 5 are calculated using a point-based scoring system. The system works by giving points based on the position in the ordered list. If a tool is in first place of the list (meaning most flexible/useful), than it would get 5 points. If a tool is second place, than it would get 4 points, etc. The sum of all the points, from all participations, are represented in these two columns. Interestingly, Portent Title Generator scores the highest.

The final two columns in Table 5 are the number of times the tool was in first place. This approach of analyzing was also included, since ordered lists are difficult to properly analyze. Additionally, for the research question, it is good to know how many times a tool was put in first place. For flexibility, Portent Title Generator has the highest score, shortly followed by ParaMind. For usefulness, there is a tie between Orteil's Game Generator, SubMind and Portent Title Generator.

VI. Discussion & Conclusion

The aim of this research was to determine which tool for generating ideas (in a videogame context) is the best, and why this is the case. The best way to determine this, using the collected results, would be to select the tool that has scored the highest in most methods of grading. Before a final conclusion can be drawn regarding the research question, the results are discussed per method of grading below.

A. Discussing the observations

The first method of grading shows that Orteil's Game Generator generates a "good" idea with the least amount iterations. This means that when using Orteil's Game Generator, the least amount of time needs to be spend before finding an idea that is satisfying. The fact that, only for this tool, consistently across all participants three ideas were marked, proves that there is a very high "satisfaction" rate. In all results for the first method, this tool has a clear advantage. On a side note, this method of grading that was specifically designed for this research provided interesting results. Future researches of a similar nature could also use this method in their experiment.

For both originality and elaboration Orteil's Game Generator scores the highest. It would mean that the ideas generated by it were found to be unique or unusual and clear enough for further usage. Interestingly, SubMind has only a slightly lower score for originality, while also having a rather low elaboration score. This can be explained due to the fact that SubMind uses an enormous database of information (DBpedia) for collecting words, which are then randomly used to generate ideas. It makes sense that this would lead to unusual results, but also has a high chance of being vague since there are no checks to see whether the generated sentences make sense. More about this in section VI.C.

When looking at the point-based scores of flexibility and usefulness, it is rather interesting that Portent Title Generator is the highest scoring one by a fair margin. This makes these scores hard to explain when looking at the results from the previous methods, although looking at the 1st position resolves this. Portent Title Generator's higher flexibility score may stem from the fact that its output is less fixated on videogames and can be applied more broadly. The same can be said for ParaMind, which has only a slightly lower score for flexibility. The results from the first method indicate that this doesn't necessarily lead to more "good" ideas, however. This makes the value of the flexibility score questionable. Furthermore, it is interesting to see that for usefulness, Orteil's Game Generator, SubMind and Portent Title Generator are tied. The best way this result can be explained, is that when "good" ideas are generated by one of these tools, it definitely can be useful in a video game idea context.

B. Answering the main research question

With the above information, the research question can be answered. Orteil's Game Generator is the best tool available. The reason for this is that this tool has a very high chance to produce a "good" idea, is very efficient in terms of iterations needed to produce "good" ideas and that the ideas generated are both original and clear for further usage.

Additionally, the results seem to indicate that when an idea generator has a focus on a certain topic, the quality of the ideas generated improves. At least when using video games as a topic. Ideas generated by Portent Title Generator or ManyThings Sentence generator might be too general, and thus have a lower chance to trigger inspiration. To be sure whether this indication is actually correct, the experiment would have to be done with multiple other topics, e.g. song titles or movie concepts.

Also having complex mechanisms such as association in tools don't seem to provide an advantage over tools that work using far simpler mechanisms. Less is more, would be a saying that fits here.

C. Reflecting on SubMind

SubMind, the tool created especially for this research, has one important issue that can be observed. When generating a database dynamically, it is very difficult to determine whether the words in the generated database can be put together in one sentence and whether the original word from the input sentence can be blindly replaced by an associated word. For example, during the experiment the following sentence was used: *a video game about a creature living in an environment*.

One idea that SubMind generated using this sentence is: *A user interface design about a pelycosaurs living in an force*. Even though the words in the generated idea are indeed somehow associated with their original keywords, they don't make sense when they actually replace the original keywords.

This is likely the reason that the elaboration score is rather low. The ideas generated are quite original however, according to results. If the elaboration score can be improved, which is done by tackling this important issue, SubMind's scores would likely improve in all methods and make it work as a proper idea generator.

For tools like both SubMind and ParaMind, it might work better if participants could interact with them directly, instead of letting the tools generate ideas based using a pre-set sentence and presenting that output to the participants. This would allow users to play with the input, which could give a far more satisfying result. Possible variations of the experiment are mentioned in section VII.B., which could work to test this without changing the current experiment setup too drastically.

VII. Future work

There are multiple ways to how this research can be expanded. Some of the ways are mentioned in the text of this paper, although not explicitly. In this section, an overview of them is given.

A. SubMind improvements

Further experimentation with dynamic databases can be done. As mentioned in section VI.C., elaboration is the main aspect that should be improved. The best way to approach this would be to improve the way SubMind generates ideas. Currently, all nouns in the input sentence generate their own list of associations. Variations can be made here. For example, one noun from the input sentence can be selected randomly and only associations from this noun are used to replace all nouns from the input sentence. This could possibly lead to having the words relate more to each other, improving the elaboration score. Another possible improvement is to build a type of filter that removes very difficult words, such as scientific words, from the final output. This could also assist in elaboration score improvements. A possibility to achieve this, would be by checking all words in a simple dictionary.

B. Experiment variations

There are different ways of performing the experiment. If a research was dedicated to improving SubMind, a smart way to go about it is to create multiple variations of SubMind and only test those versions first. When a way of serious improvement is discovered, the comparison to the other tools used in this research can be performed again. This would require a large amount of time however, since the variations of SubMind need to be build first, and two separate experiments need to be performed.

Another way to perform the experiment is by using different seed texts for the associative tools. It might be possible that this would allow for a better representation of the associative tool's abilities. Keep in mind that this will require a substantially larger amount of participations in the experiments. Alternatively, participants could be requested to directly interact with all brainstorming tools. This would require some serious changes to the current experiment setup, however.

As an addition to the last suggestion, it might also be interesting to combine two tools. For example, Orteil's Game Generator is used to generate a few ideas. Then, an associative tool is used to expand on that idea.

References

Amabile, T., Hadley, C. N., & Kramer, S. J. (2002, August). Creativity Under the Gun. Special Issue on The Innovative Enterprise: Turning Ideas into Profits. Harvard Business Review, 80(8), 52-62.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. Scientific American, 284, 35-43.

Bringsjord, S., & Ferrucci, D. A. (2000). Artificial intelligence and literary creativity: Inside the mind of BRUTUS, a storytelling machine. Mahwah, NJ: L. Erlbaum Associates.

Dell'Angelo, L., Gaeta, M., Laria, G., Mangione, G. R., Orciuoli, F., & Ritrovato, P. (2013). A Semantic Web Vocabulary Supporting Brainstorming for Research Communities. Information Systems, E-learning, and Knowledge Management Research. Communications in Computer and Information Science, 278, 522-527.

Duch, W. (2013). Computational Creativity. In Dubitzky, W., Wolkenhauer, O., Yokota, H., Cho, K. H. (Eds.), Encyclopedia of Systems Biology, 464-468. New York: Springer-Verlag.

Feigenbaum, L., Herman, I., Hongsermeier, T., Neumann, E., & Stephens, S. (2007, November). The Semantic Web in Action. Scientific American, 297, 90-97.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., et al. (2010). Building Watson: An overview of the DeepQA project. AI Magazine, 31(3), 57-79.

Goldenberg, J., & Mazursky, D. (2002). Creativity in product innovation. Cambridge, U.K.: Cambridge University Press.

Guilford, J. P. (1967). The Nature of Human Intelligence. New York: McGraw-Hill.

IBM, Watson, Institute of Culinary Education (2015). Cognitive cooking with Chef Watson: Recipes for Innovation from IBM & the Institute of Culinary Education. (n.p.): Sourcebooks.

Lorenzo, L., Lizarralde, O., Santos, I., & Passant, A. (2011). Structuring E-Brainstorming to Better Support Innovation Processes. Fifth International AAAI Conference on Weblogs and Social Media (ICWSM11). Barcelona, Spain.

Reinig, B., & Briggs, R. (2006). Measuring the Quality of Ideation Technology and Techniques. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), 1, 1-7.

Sternberg, R. J. (1998). Handbook of Creativity. Cambridge, U.K.: Cambridge University Press.

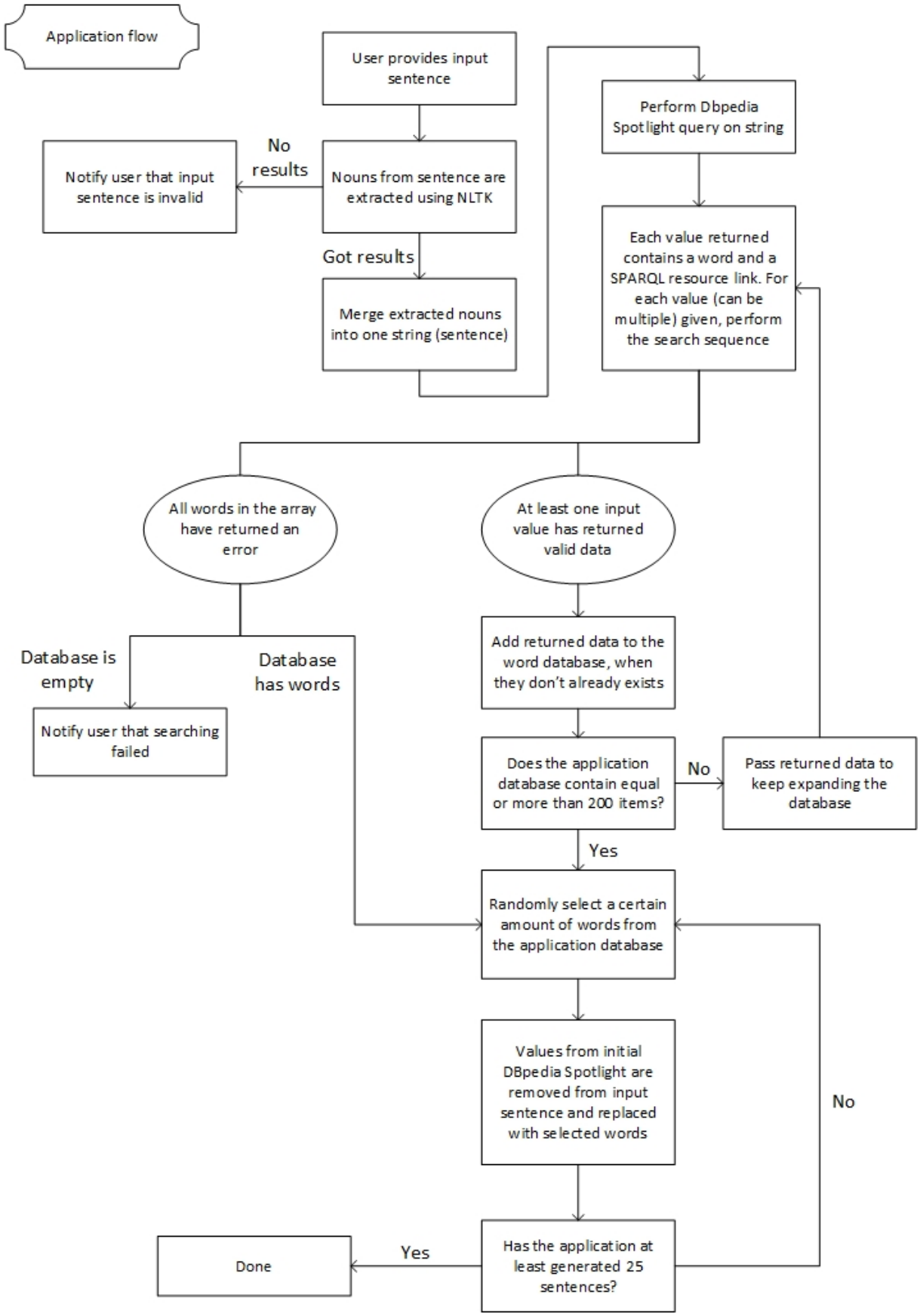
Yao, Y., Zhou, B., & Chen, Y. (2007). Interpreting Low and High Order Rules: A Granular Computing Approach. Rough Sets and Intelligent Systems Paradigms. Lecture Notes in Computer Science, 4585, 371-380.

APPENDIX A – TOOLS USED IN THE GLOBAL ANALYSIS

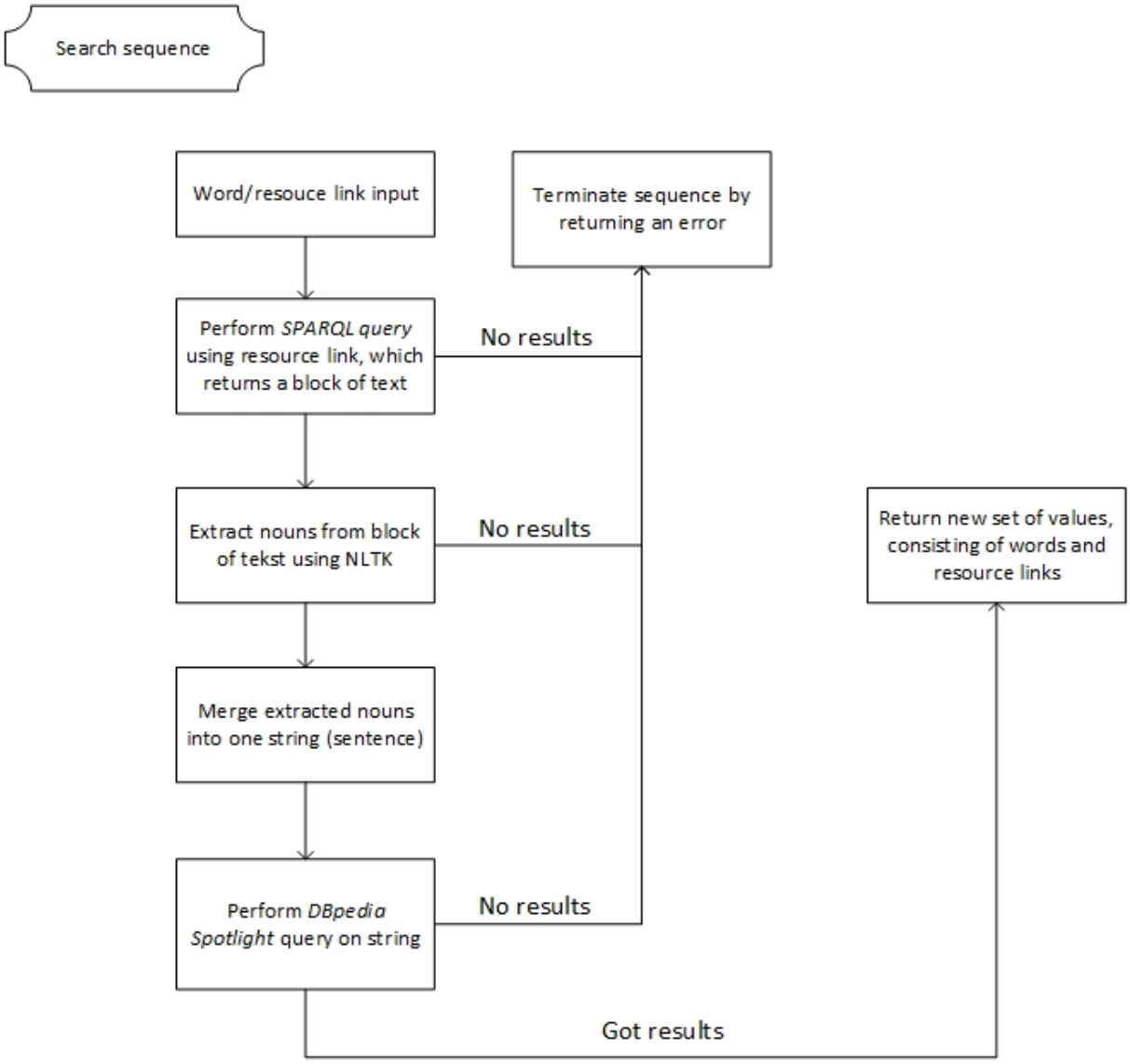
Name	Publisher	Homepage
Groupmap	GroupMap Technology	http://www.groupmap.com
BrightIdea	BrightIdea	http://www.brightidea.com
Stormboard	Stormboard	https://stormboard.com
Brainreactions.net	BrainReactions	http://brainreactions.net
Mindmeister	MeisterLabs	https://www.mindmeister.com
Hype.Go!	HYPE Softwaretechnik	http://www.hypego.net
Cognistreamer	Cognistreamer	http://www.cognistreamer.com
IdeaClouds	IdeaClouds	https://www.ideaclouds.net
MindManager	Mindjet	https://www.mindjet.com
XMind	XMind	http://www.xmind.net
Wisemapping	*Open-source*	http://www.wisemapping.com
The Brain	TheBrain Technologies	http://www.thebrain.com
Scapple	Literature & Latte	https://www.literatureandlatte.com
MindApp	Varatek Software	https://mindapp.com
MyThoughts	Mode de Vie	http://www.mythoughtsformac.com
Includer	Includer	https://includer.io
Idea Market	Exago	http://www.exago.com
Zoho Projects	Zoho	https://www.zoho.com
Agile Enterprise Architecture	Corso	https://www.corso3.com
CrowdWorx Innovation Engine	CrowdWorx	https://www.crowdworx.com
Idea Station	Datastation	http://datastation.com
ConceptDraw MINDMAP	CS Odessa Corp.	http://www.conceptdraw.com
Innovation Cloud	Datastation	http://innovationcloud.com
Mazzum	ACAR	http://mazzum.com
Ideacomb	SpadeworxSoftware	http://www.ideacomb.com
InnoCentive@Work	InnoCentive	https://www.innocentive.com
ClickCharts	NCH Software	http://www.nchsoftware.com
De Idee Management Tool	Inpaqt	http://inpaqt.nl
ErisSolver	Eris Consulting	https://www.erissolver.com
germ.io	SmokeyBeaker Labs	https://germ.io
Idea Challenges	Imaginatik	http://imaginatik.com
IdeaSwipe	Realize AB	http://ideaswipeapp.se
InnovationCast	InnovationCast	https://innovationcast.com
Innovbook	IdeaConnection	https://www.ideaconnection.com
MultiBrainStorm	Dicolab	https://www.dicolab.com
PIT	KPMG Innovation Factory	https://kpmg-innovationfactory.com
Smart Decisions Platinum	Cogentus Consulting	http://www.cogentus.co.uk
Yambla	Yambla	https://www.yambla.com
Coggle	Coggle	https://coggle.it
Freemind	*Open-source*	http://freemind.sourceforge.net
MindNode	IdeasOnCanvas	https://mindnode.com
Mindomo	Expert Software Applications	https://www.mindomo.com
Bubbl.us	LKCollab	https://bubbl.us
iMindMap	iMindMap	http://www.imindmap.nl

SpiderScribe	SpiderScribe	https://www.spiderscribe.net
Text2MindMap	Text2MindMap	https://www.text2mindmap.com
Popplet	Notion	https://popplet.com
Mind42	*Open-source*	https://mind42.com
Freeplane	Stefan Schuster	http://www.freeplane.org
LucidChart	Lucid Software	https://www.lucidchart.com
Mapul	Mapul	http://www.mapul.com
CoMapping	Comapping	https://www.comapping.com
Ideation	Qmarkets	http://www.qmarkets.net
OrganisedFeedback	OrganisedFeedback	https://organizedfeedback.com
Helprace	Stuffix	https://helprace.com
Co:tunity	Co:tunity	http://www.cotunity.com
IdeasMine	AL Consulting	http://www.ideasmine.net
Kindling	Kindling	https://www.kindlingapp.com
BrainBank/Planbox	BrainBank/Planbox	http://brainbankinc.com
iMindQ	iMindQ	https://www.imindq.com
Innovation5g	Innovation5g	http://innovation5g.com
WE THINQ	WE THINQ	https://www.wethinq.com
2-plan Desktop	2-plan	http://2-plan.com
BrightSpark	Component Workshop	http://www.brightsparkapp.com
CrowdLogic Engage	CrowdLogic	http://www.crowdlogicsystems.com
DebateMyBusiness	DebateMyBusiness	http://www.debatemybusiness.com
GainX	GainX	http://www.gainx.ca
Glint Innovation	Glint Innovation	https://www.glintinnovation.com
Ideasbank	Nucleus	http://ideasbank.io
IDhall SC	Humanperf Software	http://www.humanperf.com
Inova Suite EXPLORE	Inova	http://www.inova-software.com
MindView 6	MatchWare	http://www.matchware.com
OI Engine	IDEO	http://oiengine.com
PrIsM	ISDE	http://www.isde.com
Viima	Viima	https://www.viima.com
Yutongo	Yutongo	https://www.yutongo.com
IdeaScale	Ideascale	https://ideascale.com
Crowdicity	Crowdicity	http://crowdicity.com
IdeaFisher	Thoughtrod	http://www.thoughtrod.com
Visualthesaurus	Thinkmap	https://www.visualthesaurus.com
ParaMind Brainstorming Software	ParaMind	http://www.paramind.net
Portent Title Generator	Portent	https://www.portent.com/tools/title-maker
Orteil's Game Generator	Orteil	http://orteil.dashnet.org/gamegen
Seventh Sanctum	Seventh Sanctum	http://www.seventhsanctum.com
ManyThings Sentence Generator	Charles Kelly	http://www.manythings.org/rs/svo.html

APPENDIX B – MAIN APPLICATION FLOW



APPENDIX C – SEARCH SEQUENCE FLOW



APPENDIX D – EXAMPLE OF IDEA SET USED FOR THE EXPERIMENT

Note: Titles of generators are removed in the actual experiment, and are replaced with a numbering from A to E. Codes are used to identify the lists during result analysis.

Orteil's Idea Generator

1. A simulation game where you reticulate farms to dominate the market.
2. A shooting game where you ally with babies in a single closed room.
3. A turn-based strategy game where you destroy cities with robots.
4. A war game where you investigate citadels with businessmen.
5. An FPS where you drive trucks with robot nazis and go back in time.
6. A mobile game where you solve puzzles involving animals and you can customize your character.
7. An indie game where you stop breathing if you don't hug sarcasm while telling jokes.
8. A student project where you breastfeed hipsters and you control light and darkness.
9. An experimental game where you wander in search of mountains in binary.
10. An arcade game where you draw balls with your friends.
11. A horror game where you goof around with city guards with a snarky sidekick.
12. A war game where you nuke spaceships with love.
13. An action game where you slap cops except you're 3 years old.
14. A student project where you run away from the 4th wall and the game won't stop scrolling.
15. A tycoon game where you displace ants to establish your brand.
16. A tycoon game where you invent countries until you're bored.
17. An MMO where you go to war with goblins in gigantic strongholds.
18. An experimental game where you procedurally generate cubes but doing so loses the game.
19. An artsy game where you type clams from outside the game.
20. A shooting game where you date cavemen because they looked at you weird.
21. An MMO where you battle paladins on a boat.
22. A turn-based strategy game where you rename nations using the nuclear power.
23. An experimental game where you rethink yourself while collecting them all.
24. A tycoon game where you displace humans ad infinitum.
25. An online game where you touch numbers to buy virtual items.

ManyThings Sentence Generator

1. Those musicians flew a kite.
2. They rode a unicycle.
3. That photographer keeps a dog.
4. Debbie wrote a letter.
5. I kick a ball.
6. Those dentists played tennis.
7. Those pilots mail a package.
8. Mr. Hanson replaced a fuse.
9. Those car mechanics drove a car.
10. I catch butterflies.
11. Christine takes medicine.
12. Those science teachers painted the door.
13. That lawyer tied a knot.
14. Those photographers played the organ.
15. I wash clothes.
16. They fed the baby.
17. I tied a knot.
18. That doctor takes a test.
19. That manager shot a gun.
20. I wear glasses.
21. Those barbers buy juice.
22. They smell a flower.
23. Dick stole money.
24. I rode a unicycle.
25. I threw a ball.

ParaMind

1. Any video game about a creature living in an environment.
2. Sink video game about a creature living in an environment.
3. A character game about a creature living in an environment.
4. A lighting game about a creature living in an environment.
5. A ambition game about a creature living in an environment.
6. A baseball gloves game about a creature living in an environment.
7. A video fundraising about a creature living in an environment.
8. A video award about a creature living in an environment.
9. A video mystery about a creature living in an environment.
10. A video game into a creature living in an environment.
11. A video game for a creature living in an environment.
12. A video game about whose creature living in an environment.
13. A video game about electronics creature living in an environment.
14. A video game about a reptile living in an environment.
15. A video game about a creature dry in an environment.
16. A video game about a creature kind in an environment.
17. A video game about a creature dog in an environment.
18. A video game about a creature living for an environment.
19. A video game about a creature living in an legal environment.
20. This video game about a creature living in an environment.
21. Bowsprit video game about a creature living in an environment.
22. A camera game about a creature living in an environment.
23. A tripod game about a creature living in an environment.
24. A Madison Ave game about a creature living in an environment.
25. A supplements game about a creature living in an environment.

SubMind

1. A video device about a crocodyliforms living in an frog.
2. A tablet computers about a deserts living in an frog.
3. A fibre about a body plan living in an carbon dioxide.
4. A pinball about a infraclass living in an vacuum.
5. A trucks about a metabolic rate living in an dam.
6. A web browsers about a reptiliomorph living in an physics.
7. A TV about a centipedes living in an executive branch.
8. A cable about a sea air living in an potential energy.
9. A radius about a tetrapod living in an respiration.
10. A video device about a soil litter living in an canopy.
11. A iron about a paraphyletic living in an standard conditions.
12. A user interface design about a pelycosaurs living in an force.
13. A application software about a science living in an photosynthesis.
14. A printing about a nephridia living in an glaciers.
15. A analog stick about a digestive system living in an predators.
16. A analog about a nephridia living in an glaciers.
17. A evolution about a abyssal living in an herbivory.
18. A compasses about a toads living in an climate.
19. A evolution about a biomass living in an vacuum.
20. A mass medium about a molluscs living in an nitrogen.
21. A analog stick about a metamorphosis living in an gamma ray.
22. A pixels about a calcium carbonate living in an momentum.
23. A force feedback about a hagfish living in an joules.
24. A mainframe about a tidepools living in an silicate.
25. A microphone about a habitat living in an bedrock.

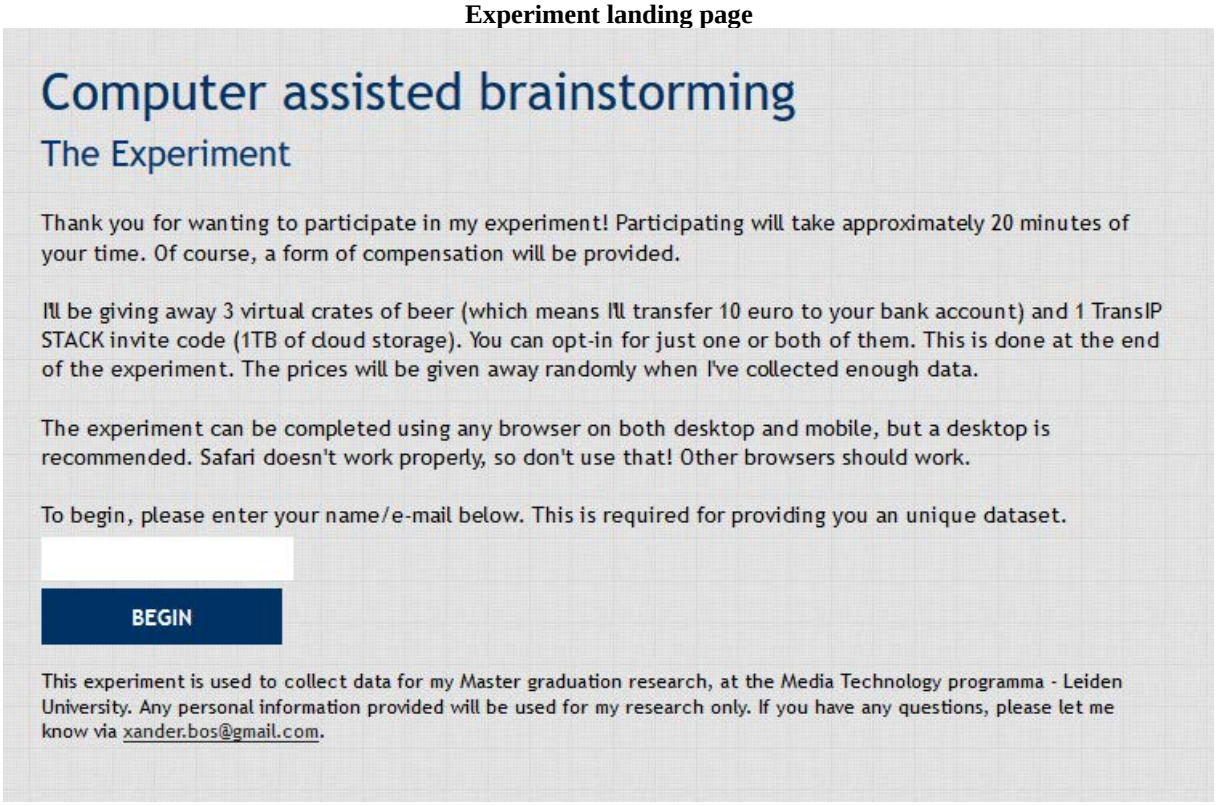
Portent Title Generator

1. The 6 worst videogames in history.
2. Why videogames are afraid of the truth.
3. 17 things you don't want to hear about videogames.
4. How videogames changed how we think about death.
5. 14 problems with videogames.
6. Why Kim Kardashian will never be good at videogames.
7. How videogames could help you win the Game of Thrones.
8. 9 BS facts about videogames everyone thinks are true.
9. Why videogames are lamer than James Franco.
10. Why videogames are afraid of the truth.
11. How videogames are bringing sexy back.
12. The oddest place you will find videogames.
13. 12 reasons videogames are the weakest links.
14. 5 ways videogames can increase your productivity.
15. Why the next 10 years of videogames will smash the last 10.
16. What Wikipedia can't tell you about videogames.
17. The best ways to utilize videogames.
18. 12 ways videogames are cooler than Michael Jordan.
19. How to start using videogames.
20. 19 reasons videogames are sweeter than Christmas morning.
21. 14 facts about videogames that'll make your hair stand on end.
22. It did what? 19 secrets about videogames.
23. How Hollywood got videogames all wrong.
24. Why videogames are more tempting than a cinnabon.
25. The 10 worst songs about videogames.

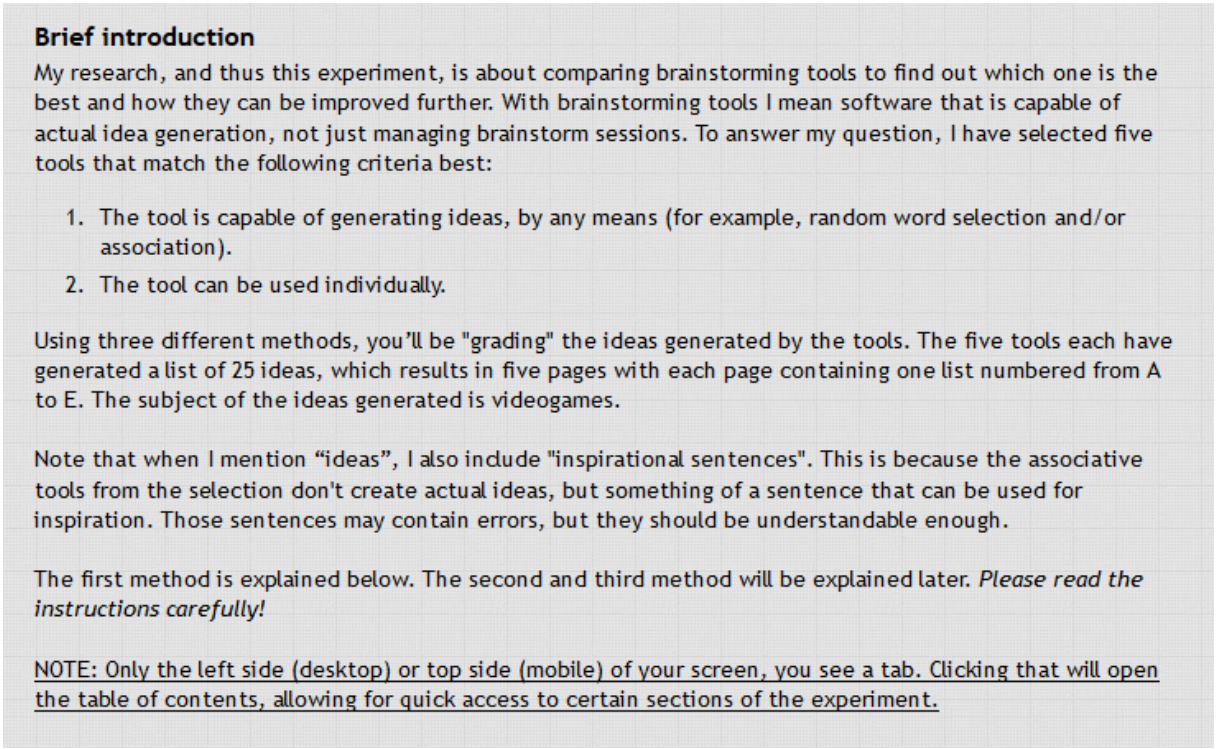
APPENDIX E – ONLINE EXPERIMENT EXAMPLE

Direct link to the experiment website: http://xandersworkshop.com/experiment/

Experiment landing page



Introduction



First method explanation

First method of grading

For each list, begin reading the ideas following the numbered order - so starting from the top, go down. While doing so, as soon as you find an idea that matches the criteria below (according to your judgement), mark it using its corresponding checkbox. Do this at maximum 3 times *per list*. If you've marked 3 items before you've reached the end of a list, you must stop and move on to the next list. If you find that none of the ideas on a list match the criteria, don't mark anything.

The criteria is:
A good idea that can be used as an element (for example, as story subject or game mechanic) for a video game.

Second method explanation

Second method of grading

The second method will use two aspects, which are commonly used in brainstorming method grading, that describe important idea characteristics. Below, a short description for each aspect is given.

- **Originality:** How original/unique are the ideas?
- **Elaboration:** Are the ideas understandable enough to develop them further/use them?

These aspects are graded using Likert scales. Every list has their own Likert scale for both "originality" and "elaboration". Since only one Likert scale per aspect is given, you'll need to grade the list as a whole.

Third method explanation

Third method of grading

For the third list, you need to order lists based on two aspects. Below, a short description for each aspect is given.

- **Flexibility:** How wide is the variety of the ideas?
- **Usefulness:** In the context of videogames, how useful are the ideas?

Additional information on flexibility: An example of flexibility grading is seen in the Alternative Uses Test. In this test, an object is presented for which as many as possible alternative uses must be found. The more categories the newly found usages can be placed in, the higher the flexibility score.
For this example, let's use a *brick*. Alternative uses could be a paperweight, a doorstop or a weapon. The first two could be placed in the "office accessory" category. A weapon would fall in the "weapon" category.

Try to use the same principle as explained above when ordering the lists based on flexibility. If it is too vague or difficult, just go with your own definition of "variety".
The lists, as you may have noticed, are "numbered" using the letters A to E. Use those letters to order the lists in the forms below.

Final question section

Final questions

These are the final questions, which will (mostly) be about you. When you've filled in the questions, you are done!

What is your age?

What is your gender?

- ○ Male
- ○ Female

How much time do/did you used to spend on playing videogames?

- ○ Never
- ○ A few times per year
- ○ A few times per month
- ○ Once a week
- ○ Multiple days per week
- ○ Every day

If you have suggestions on how ideas on the lists can be improved, leave them here. Please mention to which list(s) your suggestion applies.

To receive a copy of my paper when it's finished, leave your e-mailaddress below.

To opt-in for the free crate of beer price, leave your bank account information here (name + IBAN or PayPal).

To opt-in for the STACK invite code, leave your e-mailaddress below.