A Novel Methodology for Creating an Interactive Non-Linear Space in an Eye Tracking Environment

Gemma Fitzsimmons
University of Southampton
Gemma.Fitzsimmons@soton.ac.uk

Abstract

This paper aims to explain the purpose and creation of a novel methodology for creating an interactive non-linear space for exploring reading on the Web using eye tracking. Currently there is no easy way to explore how people read on the Web using an eye movement methodology. There are ways that we can make assumptions through gaze tracking and observe what part of a screen someone is looking, but no easy way to create a highly controlled experimental environment that can replicate results seen in traditional eye movements and reading research. In order to understand how people read on the Web we need two key elements: Firstly, we need a well-controlled stimulus set to be able to explore how people read; Secondly, we need the eye tracking technology to be highly accurate, such as that observed in eye movement and reading research using eye trackers with a high sample rate. This paper describes a new methodology for creating a highly controlled stimulus set of Webpage text which satisfies these two elements. This methodology can customise and control all of the text on the Webpages, allowing for the creation of hyperlinks and experimentally manipulated target words. This can be used with an SR Research EyeLink 1000 (an eye tracker with 1000 samples a second accuracy), which is often used in traditional eye movements and reading research.

Funding: This research was funded by an EPSRC grant for the Doctoral Training Centre in Web Science: EP/G036926/1. This work formed a part of a PhD completed in the Web Science DTC.

Introduction

In order to explore how people read in a non-linear space, such as when we read on the Web, the first task is to design an experiment where the user can navigate a Web-like environment while experimental control is still maintained.

There are currently no software solutions that can create a navigable Web environment while maintaining the control observed in traditional eye movements and reading experiments. Eye trackers cannot simply track a readers' eyes while displaying content in a Web browser, as they are often not connected to the Internet as this can slow down the sampling rate at which eye tracking samples can be collected, for numerous reasons. Furthermore, eye trackers often only collect data within their own experimental software, so you cannot record eye movements in a browser window. If you can record any screen based eye movements it will usually be with a reduced sample/accuracy rate, which is problematic for reading research that requires spatial accuracy of individual letters, or at least words and temporal accuracy to a millisecond level.

This paper focuses on the SR Research EyeLink 1000 eye tracker. The Eyelink 1000 records up to 1000 samples a second, optimum for conducting reading research. SR Research do have a system for tracking eye movements on the Web, but it has some limitations. It is essentially a screen recording, where the screen display is recorded and the eye movements on the screen are also recorded, but then the those two datasets need to be joined together to draw conclusions about what the user looked at. Although this could be used to conduct a reading experiment that explores navigation, it would be very time intensive. All of the content on the screen would have to be mapped after the experiment to work out what the user looked at for each moment. Also, experimental control would be reduced by using real websites, where other variables can influence reading behavior.

What is ideally needed to experimentally explore reading on the Web are Webpages designed for experimental control and the ability to record the eye movements in a process as similar to traditional eye movements and reading studies as possible. That way we can draw conclusions from the decades of previous research in the area and build our understanding of how people read on the Web in a navigable environment, with relevance to previously found reading effects and models. This paper explains the process of creating experimental stimuli in this way.

If we want to create an interactive website, it needs to be displayed in a way that is compatible with the eye tracking system, such as the EyeLink 1000. Essentially, the webpages need to be displayed as individual images with locations in pixel co-ordinates for the reader to click and navigate. Therefore, the aim of this paper was to create subsections of Wikipedia (which we refer to as hubs), that contain enough articles that the reader could navigate through the Web environment without running out of articles so that the experience feels like a real Web environment.

Creating a realistic Web browsing environment that possesses the required experimental control demands a large amount of design and planning. There were a number of issues that needed to be addressed in order to create this experiment and those are also discussed in detail in this chapter.

Traditional Eye Tracking

Traditional eye movement and reading experiments tend to follow a similar pattern of design and analysis. A set of stimuli is created, with various spatial and linguistic controls, and then displayed in a pseudo random order. The readers will read all of the stimuli presented to them and they have no interactive options as they can only read in the pre-specified order and then answer comprehension questions about the stimulus that has just been presented. This is relatively straightforward to create in current eye tracking software such as Experiment Builder. However, once the experimental design becomes more complex, particularly in terms of navigation, like when investigating Web reading behavior, this procedure can be insufficient, and a new methodology needs to be created. A new methodology was devised in this paper based on maintaining the positive points that the traditional methodology offers and we try to build on this to make a new methodology to explore reading on the Web.

The positive aspects of traditional eye tracking and reading research include:

Accuracy and linked to cognitive processes: The EyeLink 1000 eye-tracker has high speed and accuracy. It records a sample every millisecond and participants are calibrated to within a third of a degree of visual angle, which roughly translates to the size of a single character. Moreover, this very detailed track record of eye movement data gives us a good indication of moment-to-moment cognitive processes during reading [1–3].

Well tested and established methodology: There are a number of eye movement measures (e.g. first fixation duration, go-past reading times, etc.) that have been developed to investigate and understand reading behaviour that are well understood and can help us represent not only the presence of effects, but also the time course of effects [2].

Easy to use software for experiment creation and data analysis: The current eye tracking software for the Eyelink 1000 such as Experiment Builder and Data Viewer are adequate to create and analyze simple, static reading experiments. Interest areas and experimental design are easy to establish and implement and the eye tracking data is simple to output and statistically analyse.

There are, however, also a number of drawbacks to the traditional methodology/software:

Complexity ceiling: When experimental design becomes more complex, the Experiment Builder software on its own may not be adequate. It is designed to make the creation of simple, single target word, reading experiments. Therefore, in order to represent an environment that resembles the Web, and can be interactively interacted with, additional software is required. This software needs to create the images and interest areas on-line and integrate additional clicking functionality in order to create a realistic, interactive website environment.

Analysis issues: A large amount of traditional eye tracking measures can also be used in the analysis of more complex passages of text. However, they need to be modified to work in this new scenario. For example, most of the ways data can be output from Data Viewer results in simple, first-pass reading measures for the purposes of target word analysis. However, when reading a website, participants may scan the page before engaging in reading, thus resulting in interest areas on the entire page being marked as re-reading (because they have supposedly been skipped during first passage). Trials where this occurs render all first-pass measures void, reducing the power of traditional reading measure analysis.

Given these positive and negative points about eye tracking research, we wish to maintain the positive points, while addressing the negative aspects. Therefore, a new methodology was devised, with software created specifically to address the required complexity and data analysis issues discussed above. Essentially, an additional stage took place before the experiment was put into Experiment Builder such that we could insert an extra level of complexity into the experimental design, while maintaining the accuracy of traditional eye tracking methodology.

New Methodology

The new methodology required us to first consider the problems with implementing our experimental design into the current eye tracking software. The current software does not allow a website-like environment to be created easily. Although code can be implemented to be able to click a region and it presents an appropriate image linked to that region, it is difficult to keep track of previous pages and practically impossible to have the functionality of a website that includes making any previously visited links purple. Essentially, we needed to add additional functionality that is not covered by default in Experiment Builder. These are:

Clickable regions for navigation: The location of each word and size needs to be identified and monitored such that when a word is linked, the clickable region is known and can be utilized. This allows a user to click the word and navigate to a linked page.

Interest areas: Each word location needs to be known such that fixations within interest areas can be monitored. In Experiment Builder, interest areas can be created, but they tend to be created by hand, drawing each one on the stimulus image. This is time consuming (especially as the number of stimuli increases) and human error could easily occur. For a clickable Web environment, a number of interest areas need to automatically created, ideally one for each word. The interest area file needs to also contain the details about each word such that during analysis each word can be distinguished and its variables are known, such as whether it is linked or not and whether it is a target word.

Visited links: When navigating on the Web, we commonly click blue links that turn purple when they are visited links. The program needs to be able to keep track of the links that have been clicked and the revisited pages and the visited links need to change in color from blue to purple. This adds to the ecological validity of the website experience.

Manageable hubs with no dead links: Each hub needs to have links in the articles that link to other, separate articles. All links need to be functional (i.e. they need to have a location that they link to when clicked) and we need to be programmed in a way to check the web environment is complete in that it has no dead links that lead to nowhere. If a link is clicked and it has no page linked to it then the experience is devalued and the program might crash.

The Web is an expansive, non-linear space: The Web is large and expansive and if the reader is going to feel like they are actually browsing the Web, they need the environment to feel large and expansive. There needs to be enough pages and links present in those pages that the user feels like they are browsing Wikipedia, without the need to re-make Wikipedia with the 39,468,450 pages it has (as of June 2016). Due to the nature of this research, target words need to be inserted into the articles which could be very labor intensive.

Part 1: Software to make images and interest areas

The first task was to create custom software that can be used to create Wikipedia articles with defined and constrained target words, which can be manipulated. The text within a Wikipedia article then needed to be turned into images that can be displayed by Experiment Builder with matching interest areas. Two versions of each article were created, one with all blue links and one with all purple links, representing the visited links (see Figure 1).

The custom software was created in C# and was named The Clicking Gemerator. It imports data from a specifically created input document (Figure 2) and generates images, interest areas and files necessary for creation of the subsequent Experiment Builder experimental set up.

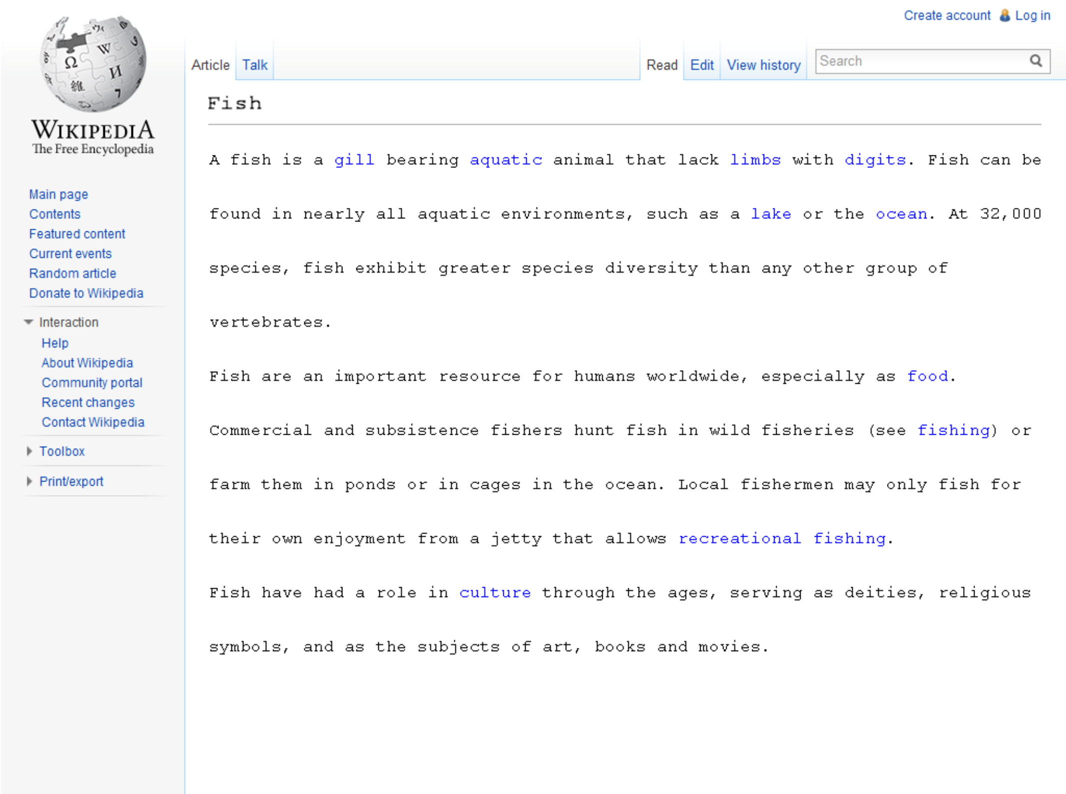
This input document (See Figure 2) contains information for generating each individual Wikipedia article. Each row corresponds to single article. Each page needs the information listed in Table 1. The text that is shown in each article is defined in the “TEXT” column. In order to manipulate the target words and what words are linked, we defined our own markup for target words and whether a word is linked and what corresponding page will be presented when that link is clicked on. This input file is imported into the Clicking Gemerator program (Figure 3)

In Figure 4, we see the example of the article labelled “Fish”. Triple hash signifies the start and end of markup, similar to < and > in HTML. In the example in Figure 4, we see the text “A fish is a **###fish_gill###**gill”. This signifies that the word *gill* is a linked word and the link goes to the page “fish_gill” which is specified as a unique identifier in the “PAGE_NAME” column. A paragraph break is signaled by the markup “##NEWLINE##”. Finally, target words are signaled with the markup “##A##”, “##B##”, “##C##” or “##D##”, dependent on how many target words are in a particular article. In this example there are two: “##A##”, “##B##”. The target words are defined by the following columns of the input file: “X_TARGETS”, “X_TARGET_LINKS” and “X_TARGET_PAGES”. The X can either be A, B, C or D dependent on the counterbalancing condition. These columns represent the target word, with the character given in place of X referring to whether that word should be linked or unlinked, and what page the target word links to when it is linked.

This input is then imported into the Clicking Gemerator program and the markup is interpreted resulting in the creation of images and interest area files from the text input. The interest area files are generated by creating an interest area around each word and marking up each interest area in the file with what the word is, whether it was linked or not and whether it was a target word. It also includes the sentence number and word number in the article (see Figure 5).

The program also creates some input files that are put into Experiment Builder. One file is a list of all the links in the experiment and links to the image that has the visited version of the page where all the links are purple (see Figure 6). The second is a region list that lists out the co-ordinates of every linked word so that when a mouse cursor is inside one of these co-ordinates and is clicked, Experiment Builder can know to present the appropriate corresponding linked page the participant wanted to navigate to (see Figure 7). These co-ordinates are also used to display the purple, visited links.

The visited links are worked out from a mixture of these two files that define what words are links, the co-ordinates for those linked words and the image of the visited linked version of that article. In Experiment Builder, a variable is created to keep track of what articles have been visited and if the article has been visited before. If so, Experiment Builder can use the co-ordinates and the link to the visited version of the article to show the visited link in purple.



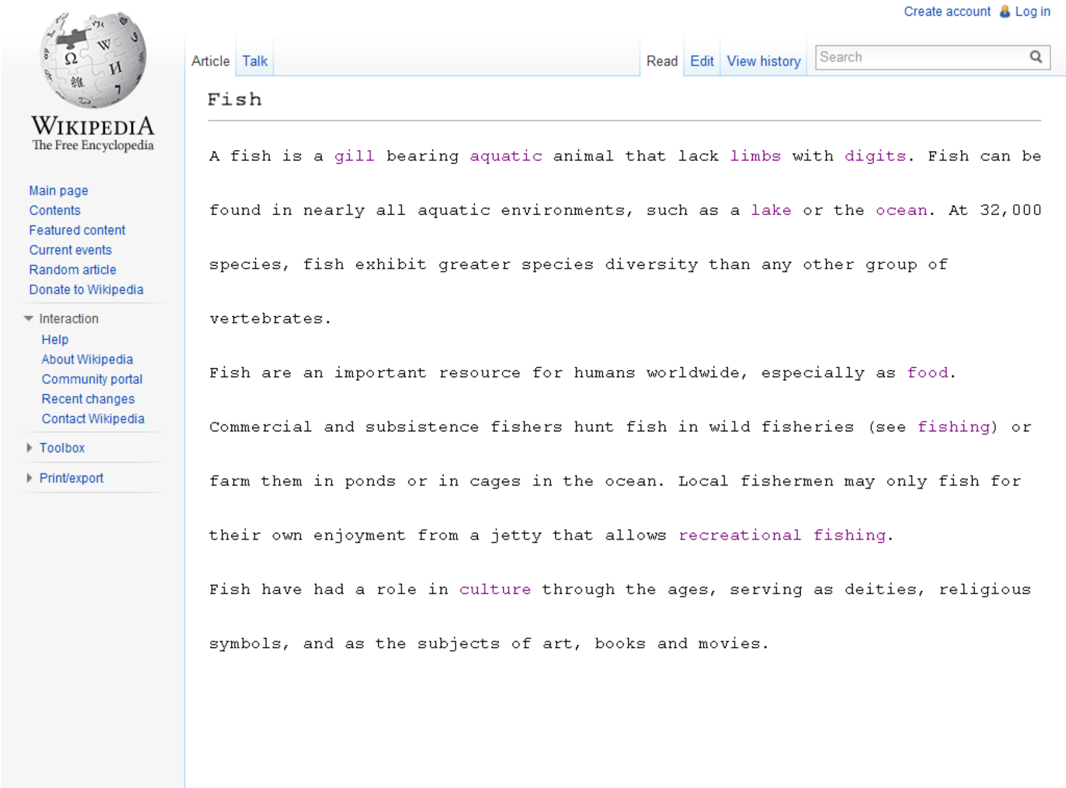


Figure 1 Unvisited and visited links an article. The top article is the unvisited article where all the links are blue. The bottom article is the visited version of the article where all the links are purple.

Table 1

All of the columns in the input file that is imported into the Clicking Gemerator and a description of what the role of each variable is.

Column name	Description
PAGE NAME	Unique identifier
TITLE	The title displayed at the top of the article
SITE	The hub the article belongs to
IS ENTRY POINT	If TRUE then this is the first page in the hub
TEXT	The text displayed in the article including code for signalling links and target words
A_TARGETS	What target words to display for counterbalancing group A
A_TARGET_LINKS	Whether the target words are linked (TRUE) or unlinked (FALSE) for counterbalancing group A
A_TARGET_PAGES	The pages each target word should link to for counterbalancing group A
B_TARGETS	What target words to display for counterbalancing group B
B_TARGET_LINKS	Whether the target words are linked (TRUE) or unlinked (FALSE) for counterbalancing group B
B_TARGET_PAGES	The pages each target word should link to for counterbalancing group B
C_TARGETS	What target words to display for counterbalancing group C
C_TARGET_LINKS	Whether the target words are linked (TRUE) or unlinked (FALSE) for counterbalancing group C
C_TARGET_PAGES	The pages each target word should link to for counterbalancing group C
D_TARGETS	What target words to display for counterbalancing group D
D_TARGET_LINKS	Whether the target words are linked (TRUE) or unlinked (FALSE) for counterbalancing group D
D_TARGET_PAGES	The pages each target word should link to for counterbalancing group D

QUESTION_TEXT	The comprehension question for the article if it has one or not (NONE)
EXPECTED_ANSWER	The expected answer for the comprehension question, either true (T) or false (F)
NUMBER_OF_TARGETS	Number of target words in the article
TARGET_ONE	First target word pair (high frequency_low frequency)
TARGET_TWO	Second target word pair (high frequency_low frequency)
TARGET_THREE	Third target word pair (high frequency_low frequency)
TARGET_FOUR	Fourth target word pair (high frequency_low frequency)

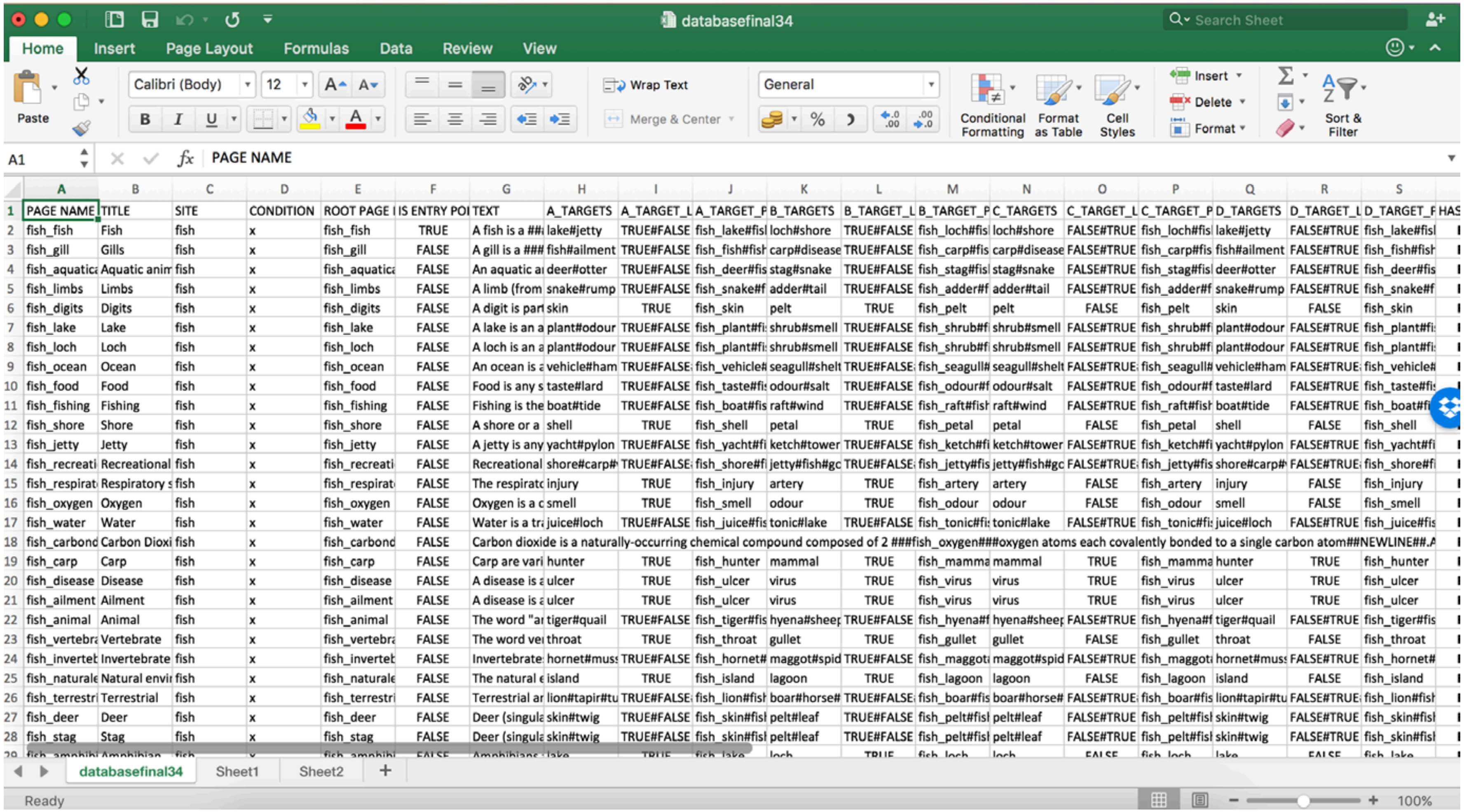


Figure 2 Input file that is created in Excel and imported into the Clicking Gemerator. One row corresponds to a single article.

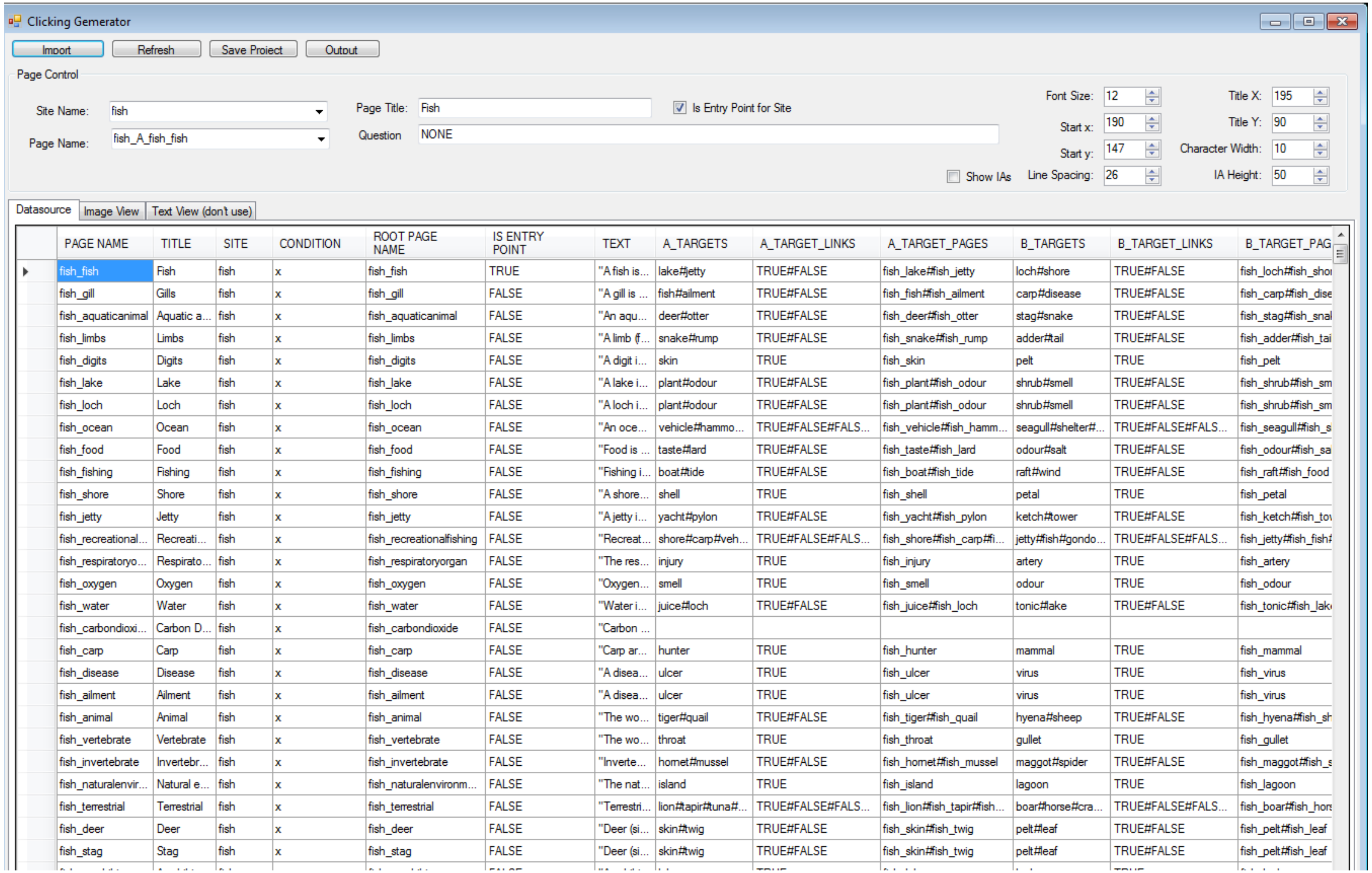


Figure 3 The interface for the Clicking Gemerator. The input file has already been imported and there are a number of variables that can be manipulated such as font size, interest area height, starting location (in pixel co-ordinates) and line height for the text to be displayed.

A fish is a ###**fish_gill**###gill bearing ###**fish_aquaticanimal**###aquatic animal that lack ###**fish_limbs**###limbs with ###**fish_digits**###digits. Fish can be found in nearly all aquatic environments, such as a ##**A**## or the ###**fish_ocean**###ocean. At 32,000 species, fish exhibit greater species diversity than any other group of vertebrates##**NEWLINE**##. Fish are an important resource for humans worldwide, especially as ###**fish_food**###food. Commercial and subsistence fishers hunt fish in wild fisheries (see ###**fish_fishing**###fishing) or farm them in ponds or in cages in the ocean. Local fishermen may only fish for their own enjoyment from a ##**B**## that allows ###**fish_recreationalfishing**###recreational ###**fish_recreationalfishing**###fishing##**NEWLINE**##.Fish have had a role in ###**fish_culture**###culture through the ages, serving as deities, religious symbols, and as the subjects of art, books and movies.

Figure 4 An example of the "TEXT" column in the input file. This includes the text that will be displayed on the article page including custom markup that will be interpreted by the Clicking Gemerator program in order to understand what words are linked and where to display target words.

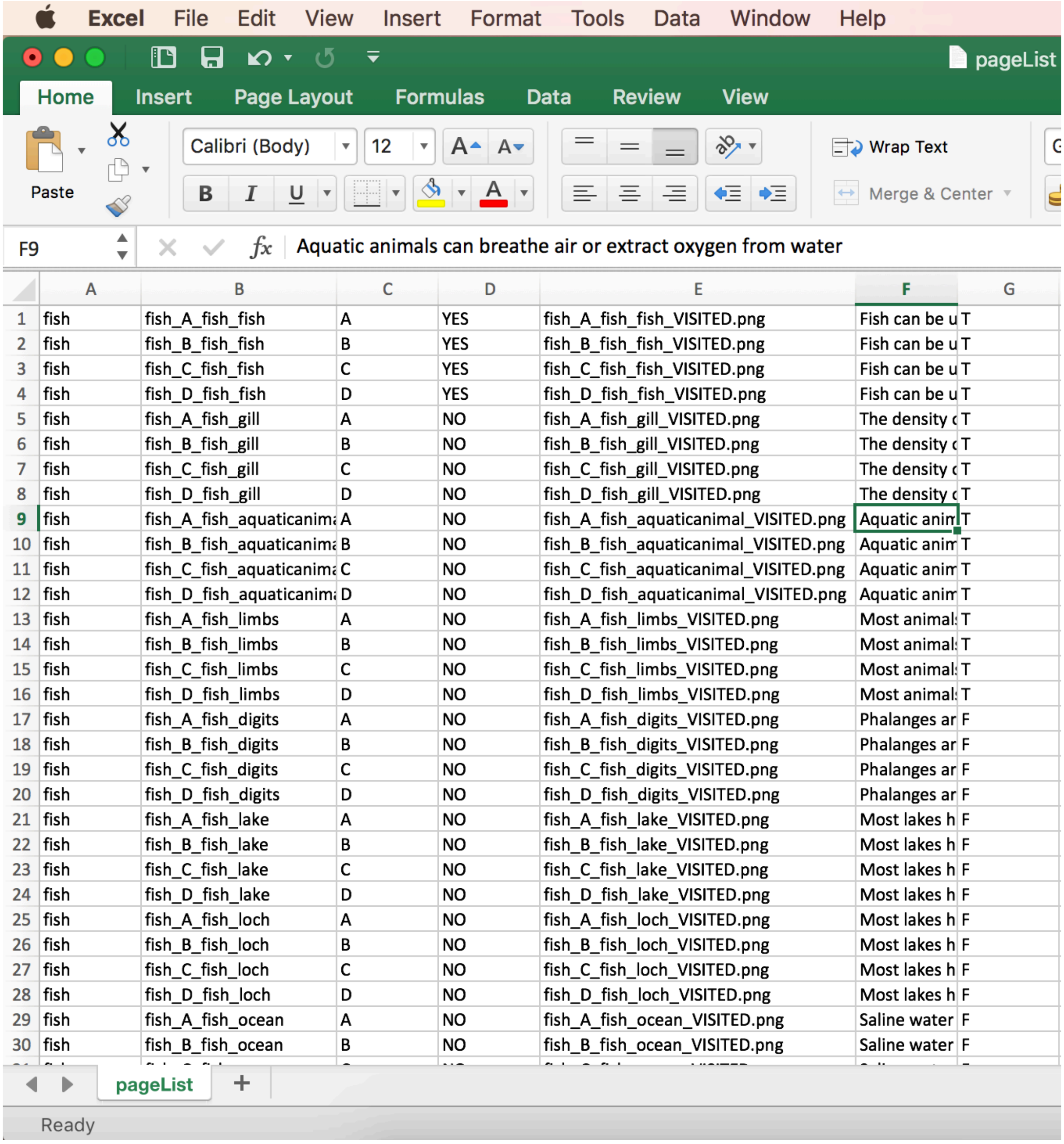
fish_A_fish_fish.ias

File Path ▾ : ~/Downloads/fish_A_fish_fish.ias

fish_A_fish_fish.ias

```
# EyeLink Interest Area Set created on 06-21-15 by the wikiGemerator
RECTANGLE	0	190 134 210 184 0_1_A
RECTANGLE	1	210 134 260 184 0_2_fish
RECTANGLE	2	260 134 290 184 0_3_is
RECTANGLE	3	290 134 310 184 0_4_a
RECTANGLE	4	310 134 360 184 0_5_gill_LINKED
RECTANGLE	5	360 134 440 184 0_6_bearing
RECTANGLE	6	440 134 520 184 0_7_aquatic_LINKED
RECTANGLE	7	520 134 590 184 0_8_animal
RECTANGLE	8	590 134 640 184 0_9_that
RECTANGLE	9	640 134 690 184 0_10_lack
RECTANGLE	10	690 134 750 184 0_11_limbs_LINKED
RECTANGLE	11	750 134 800 184 0_12_with
RECTANGLE	12	800 134 880 184 0_13_digits._LINKED
RECTANGLE	13	880 134 930 184 1_14_Fish
RECTANGLE	14	930 134 970 184 1_15_can
RECTANGLE	15	970 134 1000	184 1_16_be
RECTANGLE	16	190 186 250 236 1_17_found
RECTANGLE	17	250 186 280 236 1_18_in
RECTANGLE	18	280 186 350 236 1_19_nearly
RECTANGLE	19	350 186 390 236 1_20_all
RECTANGLE	20	390 186 470 236 1_21_aquatic
RECTANGLE	21	470 186 610 236 1_22_environments,
RECTANGLE	22	610 186 660 236 1_23_such
RECTANGLE	23	660 186 690 236 1_24_as
RECTANGLE	24	690 186 710 236 1_25_a
RECTANGLE	25	710 186 760 236 1_26_lake_TARGET_LINKED
RECTANGLE	26	760 186 790 236 1_27_or
RECTANGLE	27	790 186 830 236 1_28_the
RECTANGLE	28	830 186 900 236 1_29_ocean._LINKED
RECTANGLE	29	900 186 930 236 2_30_At
RECTANGLE	30	930 186 1000	236 2_31_32,000
RECTANGLE	31	190 238 280 288 2_32_species,
RECTANGLE	32	280 238 330 288 2_33_fish
RECTANGLE	33	330 238 410 288 2_34_exhibit
RECTANGLE	34	410 238 490 288 2_35_greater
RECTANGLE	35	490 238 570 288 2_36_species
RECTANGLE	36	570 238 670 288 2_37_diversity
RECTANGLE	37	670 238 720 288 2_38_than
RECTANGLE	38	720 238 760 288 2_39_any
```

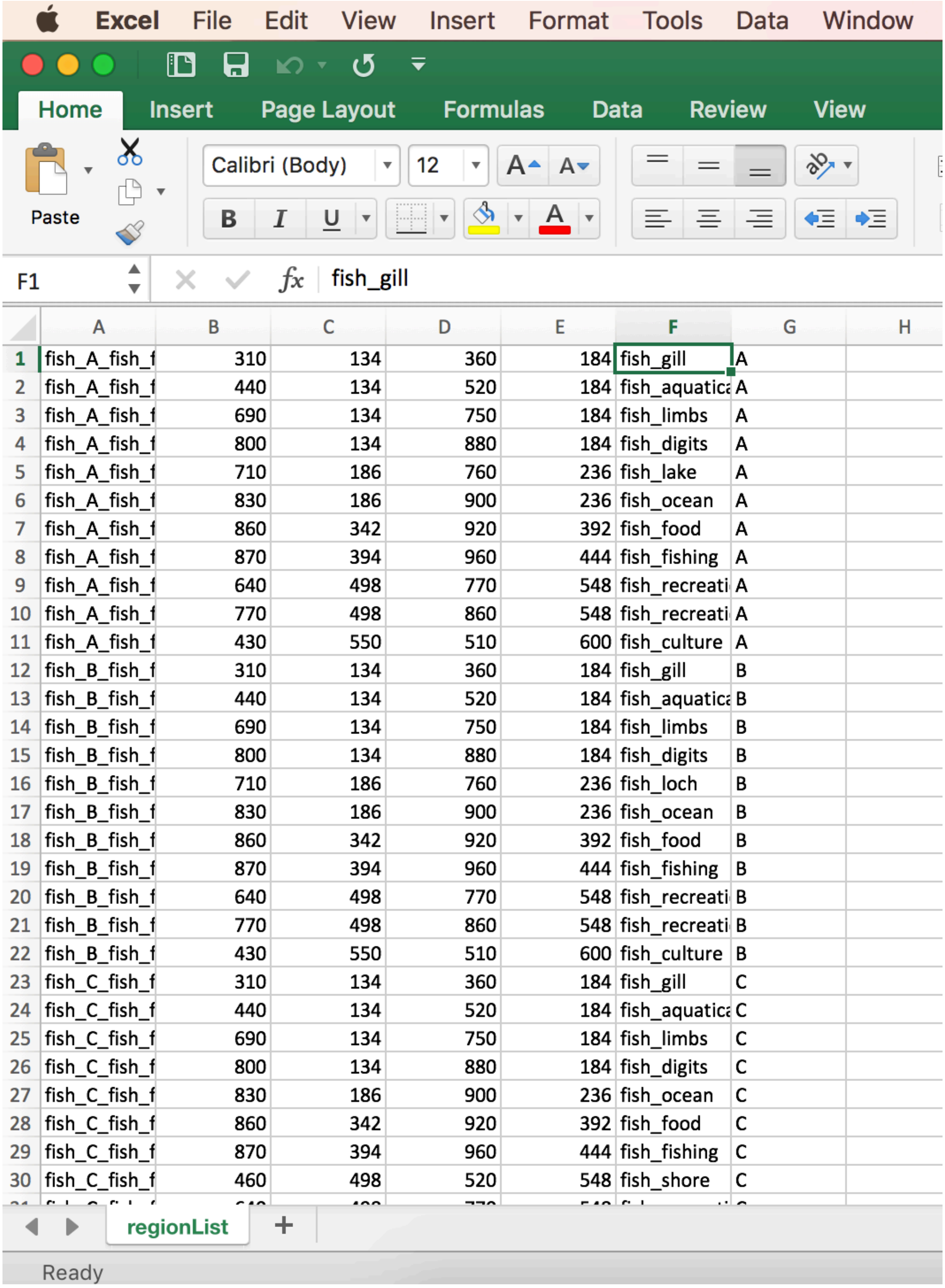
Figure 5 Example of an interest area file creating by the Clicking Gemerator.



F9 Aquatic animals can breathe air or extract oxygen from water

	A	B	C	D	E	F	G
1	fish	fish_A_fish_fish	A	YES	fish_A_fish_fish_VISITED.png	Fish can be u	T
2	fish	fish_B_fish_fish	B	YES	fish_B_fish_fish_VISITED.png	Fish can be u	T
3	fish	fish_C_fish_fish	C	YES	fish_C_fish_fish_VISITED.png	Fish can be u	T
4	fish	fish_D_fish_fish	D	YES	fish_D_fish_fish_VISITED.png	Fish can be u	T
5	fish	fish_A_fish_gill	A	NO	fish_A_fish_gill_VISITED.png	The density c	T
6	fish	fish_B_fish_gill	B	NO	fish_B_fish_gill_VISITED.png	The density c	T
7	fish	fish_C_fish_gill	C	NO	fish_C_fish_gill_VISITED.png	The density c	T
8	fish	fish_D_fish_gill	D	NO	fish_D_fish_gill_VISITED.png	The density c	T
9	fish	fish_A_fish_aquaticanima	A	NO	fish_A_fish_aquaticanimal_VISITED.png	Aquatic anim	T
10	fish	fish_B_fish_aquaticanima	B	NO	fish_B_fish_aquaticanimal_VISITED.png	Aquatic anim	T
11	fish	fish_C_fish_aquaticanima	C	NO	fish_C_fish_aquaticanimal_VISITED.png	Aquatic anim	T
12	fish	fish_D_fish_aquaticanima	D	NO	fish_D_fish_aquaticanimal_VISITED.png	Aquatic anim	T
13	fish	fish_A_fish_limbs	A	NO	fish_A_fish_limbs_VISITED.png	Most animals	T
14	fish	fish_B_fish_limbs	B	NO	fish_B_fish_limbs_VISITED.png	Most animals	T
15	fish	fish_C_fish_limbs	C	NO	fish_C_fish_limbs_VISITED.png	Most animals	T
16	fish	fish_D_fish_limbs	D	NO	fish_D_fish_limbs_VISITED.png	Most animals	T
17	fish	fish_A_fish_digits	A	NO	fish_A_fish_digits_VISITED.png	Phalanges ar	F
18	fish	fish_B_fish_digits	B	NO	fish_B_fish_digits_VISITED.png	Phalanges ar	F
19	fish	fish_C_fish_digits	C	NO	fish_C_fish_digits_VISITED.png	Phalanges ar	F
20	fish	fish_D_fish_digits	D	NO	fish_D_fish_digits_VISITED.png	Phalanges ar	F
21	fish	fish_A_fish_lake	A	NO	fish_A_fish_lake_VISITED.png	Most lakes h	F
22	fish	fish_B_fish_lake	B	NO	fish_B_fish_lake_VISITED.png	Most lakes h	F
23	fish	fish_C_fish_lake	C	NO	fish_C_fish_lake_VISITED.png	Most lakes h	F
24	fish	fish_D_fish_lake	D	NO	fish_D_fish_lake_VISITED.png	Most lakes h	F
25	fish	fish_A_fish_loch	A	NO	fish_A_fish_loch_VISITED.png	Most lakes h	F
26	fish	fish_B_fish_loch	B	NO	fish_B_fish_loch_VISITED.png	Most lakes h	F
27	fish	fish_C_fish_loch	C	NO	fish_C_fish_loch_VISITED.png	Most lakes h	F
28	fish	fish_D_fish_loch	D	NO	fish_D_fish_loch_VISITED.png	Most lakes h	F
29	fish	fish_A_fish_ocean	A	NO	fish_A_fish_ocean_VISITED.png	Saline water	F
30	fish	fish_B_fish_ocean	B	NO	fish_B_fish_ocean_VISITED.png	Saline water	F

Figure 6 One of the two additional files created by the Clicking Gemerator. This file lists all of the links in each article, one row per link. It also contains the name of the image that displayed all of the links as visited. This file is used by Experiment Builder to display the visited links.



	A	B	C	D	E	F	G	H
1	fish_A_fish_f	310	134	360	184	fish_gill	A	
2	fish_A_fish_f	440	134	520	184	fish_aquatic	A	
3	fish_A_fish_f	690	134	750	184	fish_limbs	A	
4	fish_A_fish_f	800	134	880	184	fish_digits	A	
5	fish_A_fish_f	710	186	760	236	fish_lake	A	
6	fish_A_fish_f	830	186	900	236	fish_ocean	A	
7	fish_A_fish_f	860	342	920	392	fish_food	A	
8	fish_A_fish_f	870	394	960	444	fish_fishing	A	
9	fish_A_fish_f	640	498	770	548	fish_recreati	A	
10	fish_A_fish_f	770	498	860	548	fish_recreati	A	
11	fish_A_fish_f	430	550	510	600	fish_culture	A	
12	fish_B_fish_f	310	134	360	184	fish_gill	B	
13	fish_B_fish_f	440	134	520	184	fish_aquatic	B	
14	fish_B_fish_f	690	134	750	184	fish_limbs	B	
15	fish_B_fish_f	800	134	880	184	fish_digits	B	
16	fish_B_fish_f	710	186	760	236	fish_loch	B	
17	fish_B_fish_f	830	186	900	236	fish_ocean	B	
18	fish_B_fish_f	860	342	920	392	fish_food	B	
19	fish_B_fish_f	870	394	960	444	fish_fishing	B	
20	fish_B_fish_f	640	498	770	548	fish_recreati	B	
21	fish_B_fish_f	770	498	860	548	fish_recreati	B	
22	fish_B_fish_f	430	550	510	600	fish_culture	B	
23	fish_C_fish_f	310	134	360	184	fish_gill	C	
24	fish_C_fish_f	440	134	520	184	fish_aquatic	C	
25	fish_C_fish_f	690	134	750	184	fish_limbs	C	
26	fish_C_fish_f	800	134	880	184	fish_digits	C	
27	fish_C_fish_f	830	186	900	236	fish_ocean	C	
28	fish_C_fish_f	860	342	920	392	fish_food	C	
29	fish_C_fish_f	870	394	960	444	fish_fishing	C	
30	fish_C_fish_f	460	498	520	548	fish_shore	C	

Figure 7 One of the two additional files created by the Clicking Gemerator. This file contains all of the links present (one row per link) and their pixel co-ordinates in each article. This file is used to detect whether a mouse click is within a linked region and also to assist with showing visited links in the correct location.

Part 2: Code to manage hubs and visited links

The input needs to be checked to ensure that there are no dead-end links that could crash the experiment. Each link needs to have a unique identifier and any link needs a corresponding article to exist. This is important for tracking which pages people have already seen and making sure the visited links appear when appropriate. An R script was created to go through the input file and check whether every link in the 'X_TARGET_PAGES' column exists in the 'PAGE_NAME' column, which is the unique identifier for each page. If there are any links to pages that do not exist, the script flags up any of these instances of broken links. These were usually simple typographical errors that were easily corrected.

Part 3: Combine in Experiment Builder to get a complex, interactive Web browsing environment

Taking the images and interest areas and putting them into Experiment Builder is the final step towards building an interactive Web environment. We defined earlier that we would need four issues addressed if we wanted a well-controlled, interactive Web environment. These were:

Clickable regions for navigation: We have a list of all of the regions of each linked word and these can be fed into Experiment Builder and the experiment program can be created to respond to a mouse click in these regions.

Interest areas: We have created our own interest area files that contain the location for every word in the display. These interest area files also contain additional information about every word, i.e., whether it is linked, whether it is a target word and its sentence number and word number in the article.

Visited links: This issue requires a few steps to solve, in order to include visited links in the experiment. Firstly, we need to be able to monitor which pages have been visited and display those links in purple. Therefore, we need the location of all links and the ability to list all the pages the reader has already visited. Secondly, we need to have a way of displaying the articles including purple links.

Solving the first problem is simple, given the fact we already have a list of links and the regions they occupy. The only additional programming required in the Experiment Builder experimental file is to include a function that adds each article that is visited to a list. When a new article is visited/generated that list is checked to know whether to display the link as blue (unvisited) or purple (visited).

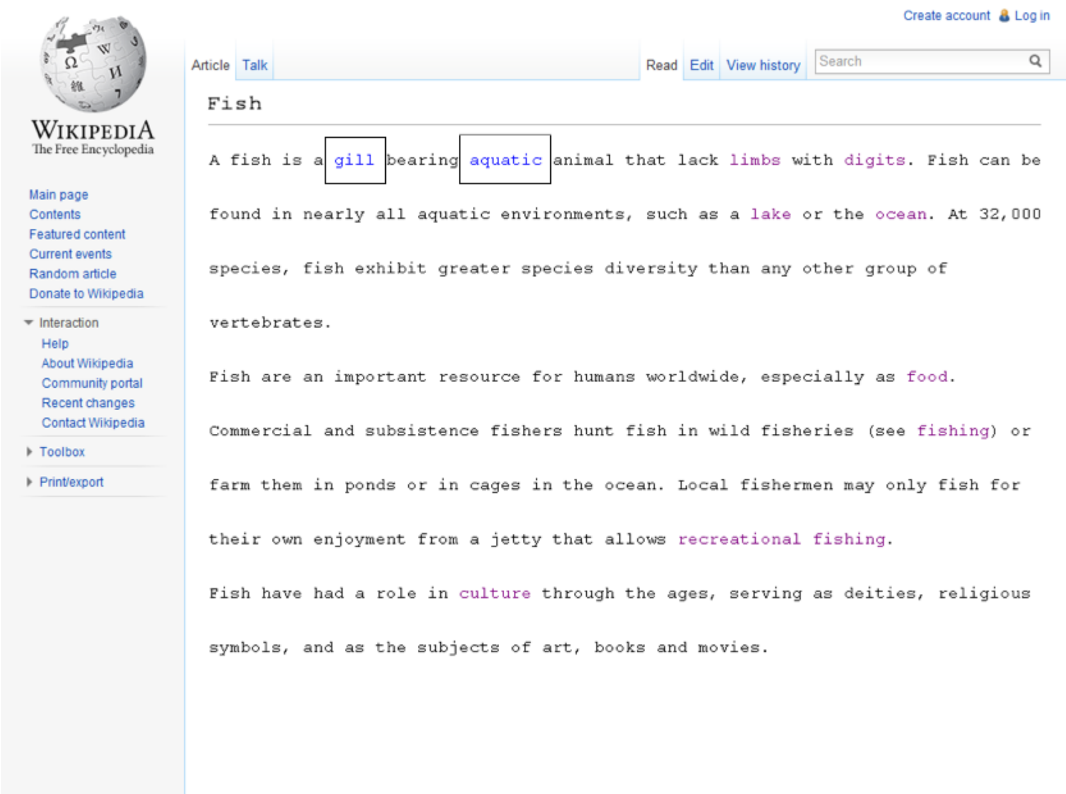


Figure 8 An example of how the visited and unvisited links are created. The visited page (with all purple links) is the default page. The linked regions of the unvisited page containing blue links is shown on top of the visited page unless that particular link has been visited. In this example, we can see that the regions for the linked words 'gill' and 'aquatic' have the unvisited linked words laid on top of the default visited image (black border is only present to illustrate the edges of the link region).

The second problem of actually displaying the visited links is slightly more complicated. From the Clicking Gemerator, we generated two versions of each article, one where all the linked words were blue and one where all the linked words are purple. Experiment Builder runs using Python and there is the ability to insert custom classes into the experiments. In this case, custom classes were created to display the visited version as the default image for the display of each article. The link co-ordinates are known from the region file (see Figure 7) that was generated by the Clicking Gemerator. Experiment Builder is instructed in the custom Python class to display the linked regions of the unvisited version of the page on top of unvisited version of the article. If the page has been visited before, Experiment Builder is instructed to not display the unvisited region for that particular link (see Figure 8).

Manageable hubs with no dead links: This issue was resolved by checking that all page links existed as actual pages. This was completed via the creation of the R script mentioned in: Part 2: Code to manage hubs and visited links.

Part 4: The final issue: The Web is an expansive, non-linear space

Finally, we need to address the, non-software based issue with the following experiment:

The Web is an expansive, non-linear space: This issue is essentially a time-based problem. There was a need to create enough articles that the reader could navigate and read articles with target words, but still had the freedom and range of articles to still feel like they are in an expansive space. This was addressed by creating four hubs rather than one giant website. This way the reader could have four unique starting pages that all readers would have and then the hubs were made to only go 4 levels deep from these starting pages.

The reason for limiting the number of levels is to limit the expansiveness of each hub. The top level of the hub is the starting page and any link on that page links to a page with its own links, which in term link to pages with their own links (see Figure 9). Quickly, the amount of pages that need to be created can expand to an exponential amount. Therefore, the number of levels was restricted and at level four most of the links must link back to articles that already exist in the hub. Through this process, the creation of additional articles can be avoided. On average, the first page had ten links and the pages those linked to would also attempt to have ten links, but after that the number of links was dependent on the article length. Some dead-end pages (pages with no links where the reader would have to press the back button to go to the previous article) were also included to limit how expansive each hub could be. There were 843 articles created in total across four hubs. In terms of target words, out of the 843 wiki pages, 326 contained target words and these were mostly focused on the top three levels of the hubs. Some pages had multiple target words: 191 pages contained 1 target word; 124 contained 2 target words and 11 contained 4 target words. There were 472 target word pairs used, but only one target word per experimental sentence. Each article ranged between 1 and 14 sentences long; on average there were 5.44 sentences in the articles. Some of the dead-end articles contained a single sentence to reduce time spent of them when there was no links or target words that would be of interest to analyze. Each reader needed to look at ten unique pages in each hub to end the experiment so that each participant had read 40 articles.

In conclusion, this paper outlines the creation of a stimuli set for use in a pseudo-Web environment that allows for navigation of a non-linear space while controlling the characteristics of the text on each page and what words are hyperlinked. This methodology allows the creator to create custom Web hubs that the user can navigate and allows the creator to control the stimuli text and target words to the same degree as in traditional eye movement and reading experiments. The analysis of the data can be run in a similar way to that completed in traditional eye movement and reading studies, where each page is a trial that can contain target words that can be identified from the tags on the interest area report. As with most eye movement and reading experiments the bulk of the work is related to the preparatory work related to creating the stimuli text. For the running and analysis of the data, however, the work is similar to that observed in other reading experiments.

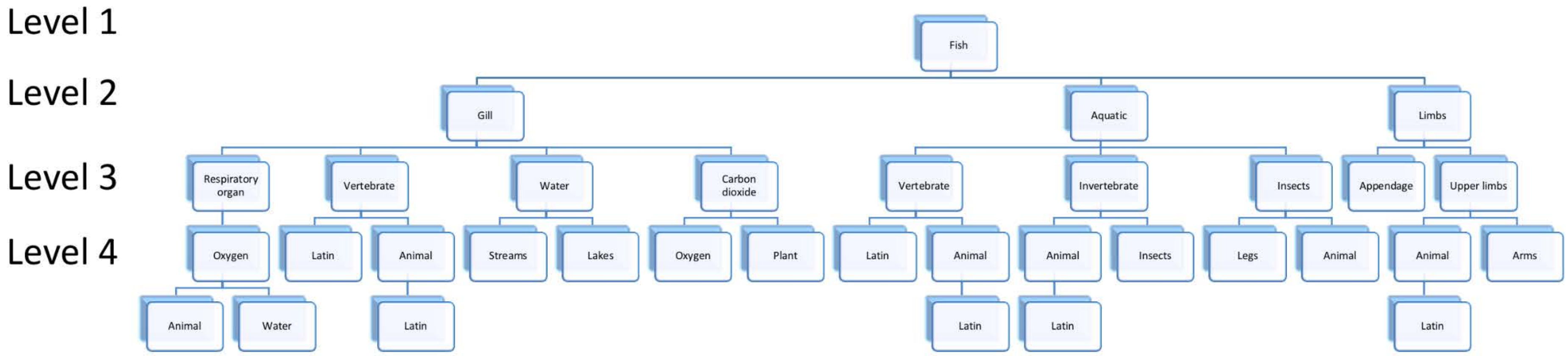


Figure 9 A simplified example of a hub and the number of levels. You can clearly see how from a single page of ' Fish' the number of articles needed dramatically increases at each level. This is why when level 4 is reached, the majority of links should link back to other articles that already exist in the hub and the creation of new articles is avoided. This simplified example does not include all level 4 links for the purposes of clarity.

References

1. Liversedge SP, Findlay JM. Saccadic eye movements and cognition. Trends Cogn Sci. 2000;4: 6–14. doi:10.1016/S1364-6613(99)01418-7
2. Rayner K. Eye movements in reading and information processing: 20 years of research. Psychol Bull. 1998;124: 372–422. doi:10.1037/0033-2909.124.3.372
3. Rayner K. Eye movements and attention in reading, scene perception, and visual search. Q J Exp Psychol. 2009;62: 1457–1506. doi:10.1080/17470210902816461