University of Dublin

TRINITY COLLEGE

Understanding Electoral Systems

A simulator to aid in the teaching of electoral vote counting and constituency configuration

Lee Campbell
B.A.(Mod.) Computer Science
Final Year Project: April 2020
Supervisor: Professor Brendan Tangney

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism ‘Ready Steady Write’, located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ______________________ Date: ______________________

Acknowledgements

Firstly, I would like to thank my supervisor, Prof. Brendan Tangney, for his unwavering support and encouragement throughout this project, and indeed over the past four years. His inimitable style gave me the opportunity to express my own ideas, whilst always challenging me to look deeper, and this project is undoubtedly better for it. I'm also hugely grateful to my friend Joe for his steadfast feedback and support throughout this project.

To my parents, my grandparents and all of my family for affording me this incredible opportunity - thank you for always encouraging me. I'm eternally indebted to you.

Finally, to my friends, thank you for giving me the breadth to take on this challenge. You've all given me memories to last a lifetime and through all the trials and triumphs, I wouldn't have wanted to share these past four years with anyone else.

L

Abstract

In an era when democracy is under threat from a variety of sources, some of them technical in origin, successful civic education is increasingly difficult, and more important than ever.

Research shows that an understanding of how electoral systems work is a critical factor in voter turnout and promotes equal civic participation across all demographics by equipping young people to navigate new media sources. Despite this, educators often lack the tools to effectively help students to fully understand how electoral systems work, how they can be manipulated, and how that manipulation can affect political outcomes. The aim of this project is to design, build and evaluate the effectiveness of a simulator to aid the teaching of constituency configuration and vote counting. The simulator is designed for use by Transition Year students in secondary school.

Different pedagogical theories are used to underpin the design of the instructional software. In this study, the ideas of constructivism and constructionism are embodied in a micro-world simulation. The simulator allows the learner to configure an election by changing constituency boundaries and electoral voting systems and clearly visualises the effects that these decisions have on the outcome of an election. This study draws on a range of data visualisation and human-computer interaction techniques to develop novel ways of allowing users to do this, for example, the use of Voronoi diagrams to improve the fluidity of constituency manipulation compared to previous systems. The application itself models two electoral systems PR-STV and First Past the Post, and seeks to create realistic scenarios; building phenomena such as transfer-pacts between parties into the dataset. A variety of visualisation libraries are also considered and analysed with D3.js eventually selected.

While it was not possible to evaluate the tool with Transition Year students due to COVID-19, an online general survey was distributed to staff and students of the author's university in order to gather views on the tool's effectiveness, while an educator specific survey was also disseminated to gauge its effectiveness from a pedagogical perspective. Analysis of the general survey showed the potential of the tool to affect changes in user opinions of fair and unfair types of elections. Educators felt that the tool had the potential to increase engagement and improve learning outcomes for students, with 98% of respondents (N=22) stating they would be likely to use or recommend using the tool. Respondents in both surveys also expressed satisfaction with the overall usability of the tool, which was measured by a post-study system usability questionnaire. The results indicate that the simulator encompasses many of the constructionist and pedagogical attributes desired to promote positive learning outcomes and impact individuals' understanding of electoral systems. However, further work is needed to evaluate the tool with real learners in a suitable learning environment as originally intended and to further develop the tool to include real election data and additional systems.

Contents

List of Figures

1. Introduction

1.1 Motivation

Electoral systems matter in any modern representative democracy as they determine the extent to which the preferences of citizens are reflected in a government's policy decisions (Taagepera and Shugart, 1989). Democracy is readily circumvented when variables within its electoral systems are allowed to be arbitrarily manipulated. Monitoring this activity is often up to citizens and, as such, it is important that citizens fully understand how electoral systems work, how they can be manipulated, and how that manipulation can affect political outcomes (Nishino et al., 2016). Furthermore, an understanding of how our political institutions and systems work is a critical factor in influencing voter turnout and civic participation (R. Dassonneville et al,. 2012). By extension, civic education has the potential to yield significant variations in voting rates by race/ethnicity, educational attainment, and other socioeconomic and demographic factors (CIRCLE, 2019).

In spite of this, there is a lack of tools to aid in the teaching of electoral systems, meaning it is often difficult for students to learn about them. Educators can list and describe the characteristics of various electoral systems, but without contextualised examples and experimentation, many concepts remain intangible and easily forgotten (Shigemura et al., 2016)

Recent changes in electoral tactics have required that educators enable young people to navigate new media sources, critically assess information, and participate (online and offline) as active and informed citizens (OECD, 2018). It is important that young people fully understand how different electoral systems change how the views of voters are reflected.

Concurrent technological developments have provided the opportunity to move from traditional didactic teaching to newer techniques, such as simulations which allow learners to interact with variables and construct their own understanding of previously elusive concepts. This technology affords students the ability to test and critically assess ideas, skills and information.

The development of a simulator to aid in the teaching of electoral theory, then, holds the potential to create new understanding and interest in electoral systems, and to positively impact the engagement of young people in democracy as active citizens.

1.2 Goal

The goal of this project is to utilise modern data visualisation techniques and technologies in Computer Science to empower learners to explore the elections in ways otherwise may be unengaging or overly complex. The project should also draw on the area of Human-Computer Interaction (HCI) to inform the design from a usability and accessibility perspective.

The result should be the design and implementation of a simulation tool which allows a learner to configure an election in different ways, by changing constituency boundaries and electoral voting systems, and to clearly visualise the effects that these decisions have on election outcomes. The tool should be suitable for use by Transition Year students (16/17 years old), in anticipation of their approaching eligibility to vote.

Finally, the last goal of the project is to test the effectiveness of the resultant system in increasing understanding of electoral systems, and by extension the effectiveness of the application of different technologies and design theories to do so.

2. Background

This section describes the research this author has done in order to prepare for and inform the design of the project. The background section is broken down into four subsections: related work, data visualisation and interaction theory, electoral data visualisation techniques and similar applications.

2.1 Related Work

Electoral Systems

An electoral system is the set of rules that determine how votes are cast, and how those votes translate into parliamentary seats (Taagepera and Shugart, 1989). While there are many different forms of electoral systems used around the world, they can typically be divided into two categories: plurality and proportional. However, this division is not always distinct (Erik S. Herron et al., 2018).

Sometimes electoral systems are also categorised into single-seat district and multi-seat district systems, with plurality systems considered akin to single-seat districts and proportional systems considered akin to multi-seat districts.

In a typical plurality system, a voter is allowed to vote for a single candidate, and the candidate who receives the most votes (or a plurality among all of the candidates) is elected (Farrell, 2001). First-past-the-post voting is a common plurality-voting method, and is particularly common in systems that use single-seat districts; although it can be used in multi-seat districts. The UK, France and Canada are the most commonly used examples of first-past-the-post systems.

Within the proportional system or PR, there tends to be two main subtypes: the party-list system and the single transferable vote (STV) system. In a party-list system, each party has a list of candidates, to whom seats are allocated in proportion to the number of votes a party received. There is some variation in how the order of the list is constructed - in closed list systems, the party chooses the order internally, and in open list systems, the order is determined by the voter (Farrell, 2001).

Using the STV system, voters put the candidates into an order of preference, by marking a 1 beside their preferred candidate, a 2 beside their next preferred candidate, and so on. In this system, a vote passes from a candidate to the next voter's next preference in the event that the candidate is eliminated or is elected with more votes than are required. (Irish Department of Housing, Planning & Local Government, 2020). Party-list PR is the more common of the two, and is the most popular electoral system in the world, with more than 80 countries using this system or a variation to elect their parliament (UK Electoral Reform Society, 2020). However, many who originally advocated for PR including Mill (1861), Hoag and Hallett (1926) argue that STV, the variation of PR used in Ireland, is the superior of the two (Gallagher et al., 2006).

The variations within PR systems can often be classified by their vote reallocation method. There are two main types, Largest Remainders Methods and Highest Average Methods, or a mix thereof (Erik S. Herron et al., 2018). The Highest Average Method (used in PR List) is sometimes referred to as the divisor method, as it requires the number of votes for each party to be divided successively by a set of divisors. This results in a table of quotients consisting of a column for each party and a row for each divisor. A seat is then allocated to the party who has the largest quotient entry and then to the party who has the next largest entry. This continues until all seats are filled. The most popular method, the D'Hondt formula which tends to favour larger parties, uses natural numbers as divisors - 1, 2, 3, 4 etc. (Gallagher, 1992).

The Largest Remainder Method (used in PR STV) uses a quota which is calculated based on the number of seats available. Using this method, a party is awarded as many seats in proportion to the number of quotas achieved. In cases where the seats remain unfilled, they are allocated on the basis of the largest remainder, and then the second largest and so on until all seats are filled. Each party's remainder is calculated by dividing its total number of votes by the quota. The quota is calculated by two main methods, namely Hare($\frac{Total\ Votes}{Total\ Seats}$)or Droop ($\frac{Total\ Valid\ Poll}{Total\ Seats + 1} + 1$). As Hare gives a larger quota it tends to favour smaller parties, while Droop, which is used in Ireland, favours larger parties. However, Droop is more likely to produce a single-party government than Hare (Gallagher, 1992).

Electoral systems can be compared and contrasted using any of the factors described above, however, the proportionality of a particular election is often the most scrutinised aspect. As such, there are several techniques used to compare the proportionality of different electoral systems. The Gallagher Index is the most commonly used measure of proportionality (Renwick, 2015). It is calculated as the square root of half the sum of the squares of the difference between the percentage of votes received (vi) and percentage of seats (si) for each party (Gallagher, 1992).

Gallagher index:

$$\mathrm{LSq} = \sqrt{\frac{1}{2}\sum_{i=1}^{n}(V_i - S_i)^2}$$

In general, research suggests proportional systems such as PR-STV ensure much greater representation than plurality systems such as first-past-the-post, as they reduce the wastage of votes, and thus record lower Gallagher indexes (Kelly, 2014, Gallagher, 2019).

The proportionality of any electoral system can be manipulated by the use of a technique called gerrymandering. Governing parties can use gerrymandering to influence the results of future elections (McGann et al., 2016). 'Packing' and 'cracking' are used to redraw constituency boundaries in a manner which wins the most seats for the governing party's candidates. 'Packing' is the practice of placing as many opposition-friendly voters in a single constituency as possible, resulting in an opposition securing significantly more votes than required in that constituency and therefore wasting votes. 'Cracking' places clusters of opposition voters in constituencies where they are outnumbered by government-friendly voters, in essence reducing the concentration of opposition voters and reducing the number of seats won by the opposition (Winburn, 2018). Partisan and racial gerrymandering has long been the subject of the courts in the US, where it is most prevalent (NYT, 2019), and in many countries (such as the UK), independent boundary commissions have been set up to limit this practice (McGann et al., 2016) .

Gallagher et al. argue that while electoral systems are often seen as mundane and are ignored by the public, small changes to a system can yield big changes to *"the shape of the party system, to the nature of government (coalition or single-party), to the kind of choices facing* *voters at elections, to the ability of voters to hold their representative(s) personally accountable, to the degree to which a parliament contains people from all walks of life and backgrounds, to the extent of democracy and cohesion within political parties, and, of course, to the quality of government, and hence to the quality of life of the citizens ruled by that government."* (Gallagher et al., 2006).

Education and Engagement in Elections

Proportional systems should result in higher turnouts than plurality systems, given the greater likelihood of one's vote-counting naturally encourages voter engagement (Johnston and Pattie 2006). However, studies which have analysed turnout rates internationally, suggest that this is not the only influential factor at play (Kelly, 2015).

Kelly suggests that the accessibility of different systems may be a factor. PR-STV, for example, is relatively complex when compared with first-past-the-post. If voters don't understand how their vote makes a difference, they may be less likely to vote and this is a major concern (Kelly, 2015).

Statistics from the Irish Central Statistics Office (CSO) show that voting amongst young people in the 18 - 24-year-old age group tends to be much lower than other age groups with only 50 per cent to 60 per cent turnout in comparison to turnouts ranging from 85 per cent to 93 in the to 50+ year-old age group (CSO, 2011, ERSI, 2017).

Education and social factors are undoubtedly a factor here, data from The Irish Longitudinal Study on Ageing (TILDA) analysed by The Economic and Social Research Institute (ERSI) suggests that each additional year of schooling increases the probability of voting by approximately 5.5% (Ma/ERSI, 2017). Targeted civic education, in particular, has also been shown to be a critical factor in influencing voter turnout (R. Dassonneville et al,. 2012). In this vein, many countries have introduced pre-voting age classes such as Civic, Social and Political Education (CSPE) in Ireland which aims to teach school children about citizenship, democracy and provide an understanding of the PR-STV electoral system (NCCA, 2016).

Active Learning and Active Citizens

Various research suggests that traditional didactic teaching methods fail to effectively teach learners electoral systems or more broadly, active citizenship as it doesn't give the learner the opportunity to experience, contextualise and interrogate the concepts being taught. (Kelly, 2005, Shigemura et al., 2016, Gollob et al., 2015). This is exacerbated by the digital transformation of all aspects of society which is requiring educators to equip young people with new skills to enable them to critically assess information in the media (OECD, 2018).

Gollob et al. argue that *"rote-learning oriented models that are simply reduced to instruction are insufficient in creating the kind of active, informed and responsible citizenship that* *modern democracies require."* and that the role of the active citizen is analogous to that of the active learner (Gollob et al., 2015).

Active learning is a process that places student learning at its core by actively engaging learners using various methods such as problem solving, discussions or role-plays. In other words, active learning requires students to do meaningful learning activities and think about what they are doing (Prince, 2004). Active learning is generally said to take place using collaborative learning, where students work together towards a common goal in small groups or cooperative learning where there is also an emphasis on individual accountability (Smith, B., and J. MacGregor., 1992). There is considerable evidence to support the effectiveness of this form of learning in increasing students' conceptual understanding, recall and engagement in a topic (Prince, 2004, Hake, R., 1998, Bonwell, C.C., and J. A. Eison, 1991).

The idea of learning as a collaborative and engaged activity underpins constructivist learning theories; which suggest that knowledge cannot be transferred to the learner, but instead must be constructed by the learner for themselves. Piaget argues that learners come with existing beliefs, and conceptual changes only emerge through learners creating and testing their own theories of the world (Piaget, 1969). This is also emphasised by Vygotsky who suggests that people learn with personal significance in mind and thus language and culture play an important role (Vygotskiĭ, L. S., and Michael Cole, 1978).

Seymour Papert of MIT, built on the constructivist notion of building knowledge structures, arguing that this occurs most effectively in situations where the learner is actively engaged in the construction of a *"public entity"* which can be shared for others to see (Papert, 1980, Papert 1993). Core to this theory of learning is the emphasis on individual learners over the universal - where knowledge is formed through a learners' conversation with their own favourite representations, artefacts, or objects-to-think with (Ackermann, 2001). One of the most successful applications of this theory was LEGO Mindstorms - where a child can design, build and program a robot using LEGO and the Logo programming language - borne out of Papert's 1980's book Mindstorms (Mindell, 2000). Papert describes how children wanted to learn about physics and computer science because they wanted to make their cars go faster (Mindell, 2000).

The digital transformation of society over the last decade has supported the growth of this model with the development of rich technological learning environments that enable both educators and students in the creation and sharing of artefacts. Microworlds are a type of digital learning environment that serve as a terrain within which a learner may explore, navigate and test embedded ideas and concepts about the real world. The most effective microworlds, according to Andrea diSessa, have *"an easy-to-understand set of operations that students can use to engage tasks of value to them, and in doing so, they come to understand powerful underlying principles"* (diSessa, 2000). Microworlds afford students to do the kinds of thinking and cognitive exploration that would not be possible without the technology. When a microworld is well designed, utilising Paivio's dual coding theory, learners have the opportunity to construct strong referential connections between both explanatory and interactive representations of concepts being modelled in the simulation and thus enhance recall/recognition (Mayer, 2005, Sadoski & Paivio, 2000)

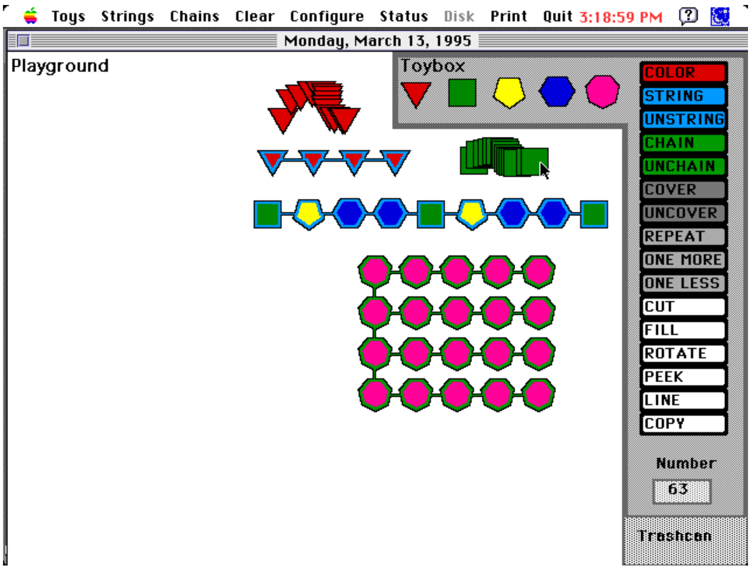


Figure 2.1: Toys microworld to engage children in units-coordinations
http://math.coe.uga.edu/olive/Tech2000/TIMA.html

Microworlds are not a panacea however, and must form part of an overall learning design to obtain effective pedagogical outcomes (Blumenfeld et al., 1991). The question of how much structure or scenarization is required in effective learning design is the subject of much debate. Central to it is Vygotsky's idea of the Zone of Proximal Development (ZPD) - the difference between what a learner learns independently versus what they learn as part of a group or from an educator or any "more able other" (Vygotsky, 1934). There are a range of non-intrusive scaffolding techniques such as process constraints, prompts and heuristics which can be used in the design of a microworld or lesson to increase ZPD in guided discovery learning (De Jong, T.; Sotiriou, S.; Gillet, D. 2014).

2.2 Data visualisation and interaction

Data visualisation is an increasingly popular field of computer science, concerned with the graphical presentation of data, with the goal of providing the viewer with a qualitative understanding of the data contents (Unrwin, 2020). It draws on several areas including human factors, statistics, computer graphics, psychology, and graphic design. Data visualisation is particularly interesting for this application in the context of the pedagogical theories, as visualisation techniques can be used to create extraordinarily powerful thinking tools.

Data visualisation takes advantage of the visual nature of humans, by shifting the balance from cognition to perception to take better advantage of the brain's abilities. The transformation of data into computer graphics exploits users' ability to find patterns and trends, thus allowing users to extract greater meaning more quickly (Murray, 2017). Patterns and symbols are essential to data visualisation as they dictate how efficiently visual queries can be processed by the user (Ware, 2008). The entity (point, line, solid etc.) used and its attributes (colour, location, size, relative position etc.) all play a key role in conveying meaning. One of the earliest theories of perception in this respect came from the Gestalt School of Psychology; centring around the idea that humans perceive objects as organized patterns and objects before seeing the individual objects (Benyon, D, et al., 2005). This resulted in several Gesult principles of grouping: proximity, similarity, enclosure, closure, and continuity which are still widely used in the design of graphical user interfaces today.

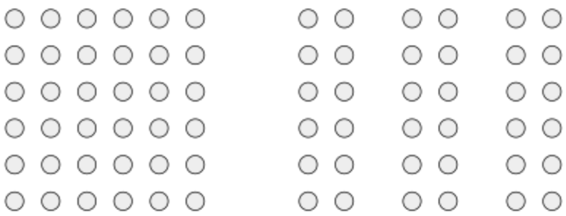


Figure 2.2: Proximity - Objects that are close together are perceived as a group.
https://en.wikipedia.org/wiki/Principles_of_grouping#/media/

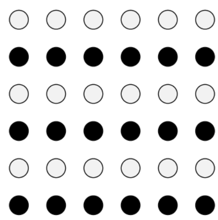


Figure 2.3: Similarity - Objects that share similar attributes (e.g., color or shape) are perceived as a group.
https://en.wikipedia.org/wiki/Principles_of_grouping#/media/

Triesman describes the concept of "Pop-out effects" which are properties of simple patterns that made them easy to find (A. Triesman, and S. Gormican, 1988). Pop-out effects describe the relationship between a visual search target and its surrounding objects. If a target is distinct in some feature channel of the primary visual cortex it can be programmed as the centre point of fixation (Ware, 2008). These are interesting as they make visuals easily searchable, and multiple channels can be used to make several things searchable at the same time. Where visuals are easily searchable thinking will be fluid, where they are not thinking can often be inefficient and frustrating.

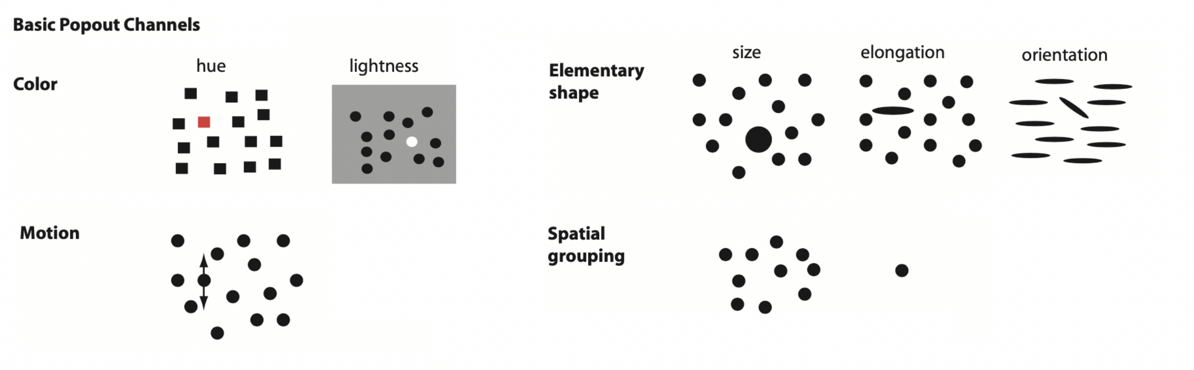


Figure 2.4: Basic Popout Channels (Ware, 2008)

visualisation is often enhanced by interactive techniques - providing information from multiple views using multiple techniques. Therefore in designing a GUI of an interactive computer program, it is important to consider how information should change in response to every mouse event (Ware, 2008). In this sense, it is helpful to think of the user and the computer together as a single cognitive entity where the computer acts as a cognitive coprocessor to the human brain (S. Card et al., 1999). In an ideal scenario, a system should responsively present the most relevant information exactly at the time it is needed, forming a cognitive loop. To make this work effectively, a user must also be able to easily drill down and get more information related to something already discovered. (S. Card et al., 1999). Drilling down is typically invoked by a mouse event, some of the quickest and most effective are mouse-over hover queries as they enable rapid interaction where *" the computer display seems like part of the thinking process, rather than something to be consulted."* (Ware, 2008).

In this model low-bandwidth information flows to the computer via mouse and keyboard events and high-bandwidth information is returned based on the processing of vast amounts of information by the computer system allowing for flexible pattern discovery via the GUI (Ware, 2008).

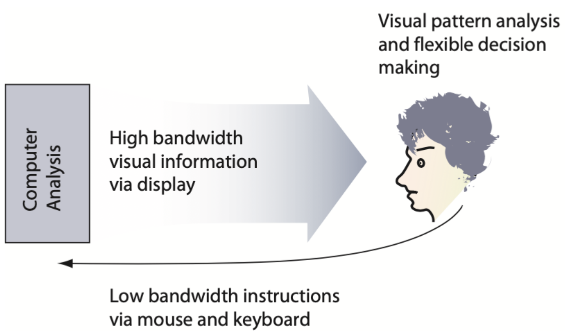


Figure 2.5: Human and computer as a single cognitive entity (Ware, 2008)

2.3 Electoral data visualisation techniques

This section will present a range of common data visualisation techniques used to visualize maps, voter data and election results, and investigate how they might be utilised in this tool.

Techniques for map visualisation

Map visualisations are used to represent geographic relationships. They are an interesting application of computer graphics as they allow users to quickly interpret complex datasets to and solve related problems by making underlying patterns and trends easy to see (Ware, 2015). Map visualisations are common in the representation of electoral data, as geography is an important aspect in elections and so it is often helpful to contextualise electoral data geographically.

Square tile-grid

In a square tile grid map, geographical information is represented by a uniform grid of squares. Square tile-grids are typically used to simplify a map and allow for easy vertical and horizontal scanning.

The example below shows a tile grid-map used by the Washington Post to illustrate the concept of gerrymandering; each square represents votes whilst the black border lines represent constituencies. In the context of this tool, this solution could be extended to enable boundary manipulation, by dragging the boundaries across squares. This is a simple and effective solution, however, the manipulation has the potential to be slow.

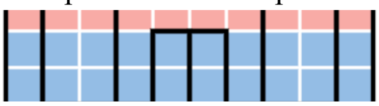


Figure 2.6: Square tile-grid example https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/

Hexagon tile-grid (Cartogram)

Often squares in a tile-grid map are switched out for hexagons, providing more flexible tile arrangement. Hexagon tessellation provides a more interesting and comfortable experience for the user, as the tiles can be arranged closer to original border adjacencies.

The example below shows the US divided into hexagons of equal size and grouped by state boundaries - each hexagon is coloured to represent the likelihood of the Democratic Party or the Republican Party winning in that area.

Similar to a square tile grid map, this solution could be extended to enable boundary manipulation, by dragging the boundaries across hexagons. This is a slightly more complex solution but it offers greater flexibility in presenting a more realistic visualisation. However, the nature of constituency manipulation would remain limited under this method.

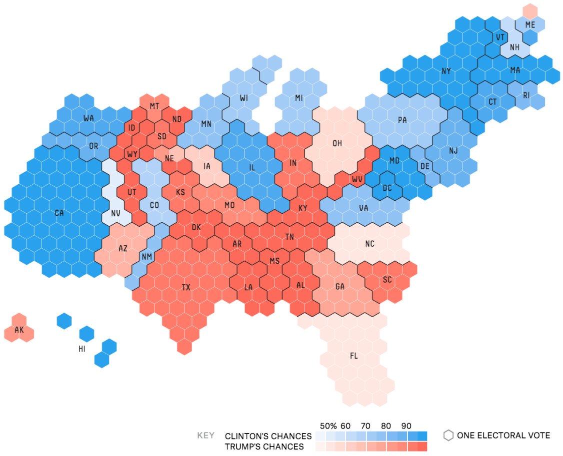


Figure 2.7: Hexagon tile-grid example
https://projects.fivethirtyeight.com/2016-election-forecast/

Voronoi diagrams

A Voronoi diagram is a simple method of dividing an area into cells. Given a set of objects on a plane, partitions are created so that each cell contains all points on the plane closer to one object than to any other object. It is frequently to choose a location for a new service or to locate things e.g. the nearest hospital.

In the context of this tool, this concept could be used to divide a region up into constituencies using polling centres as an analogy for objects. This provides an interesting and fluid way to manipulate constituency boundaries, by having the user move a polling centre to manipulate that constituency's boundaries. One drawback of this technique, however, is that it can be difficult to achieve exactly the desired change. The example below shows a Voronoi diagram of the US using states as objects, superimposed on a geographical map of the US.

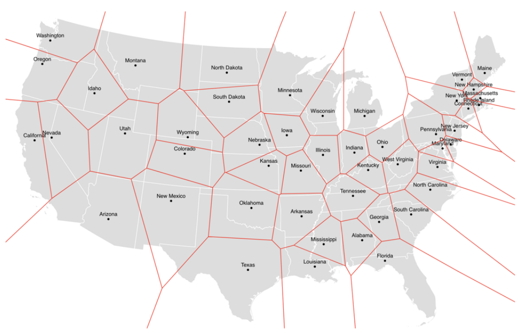


Figure 2.8: Voronoi Diagram example
https://observablehq.com/@mbostock/u-s-voronoi-map-o-matic

Proportional symbols

Coloured symbols (e.g. dots), can be used to illustrate how people vote, with population size represented by the size of the symbol. This is an effective technique, as it captures both geographic location and population density, rather than just area. However, it may appear overly complex and difficult to interpret when used on a large scale.

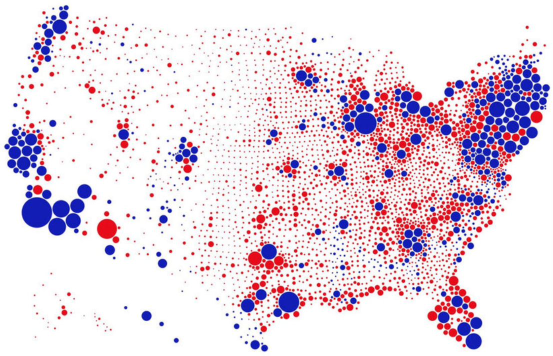


Figure 2.9: Proportional symbols map example
https://www.core77.com/posts/90771/A-Great-Example-of-Better-Data-visualisation-This-Voting-Map-GIF

Dot density:

Dot density maps use individual dots to represent information such as the number of votes for a given party in a given geographical area. This is a simple and effective way of illustrating the breakdown of votes for all parties in a given election. In the map of the US below, each dot in the map represents 5,000 votes for a candidate.

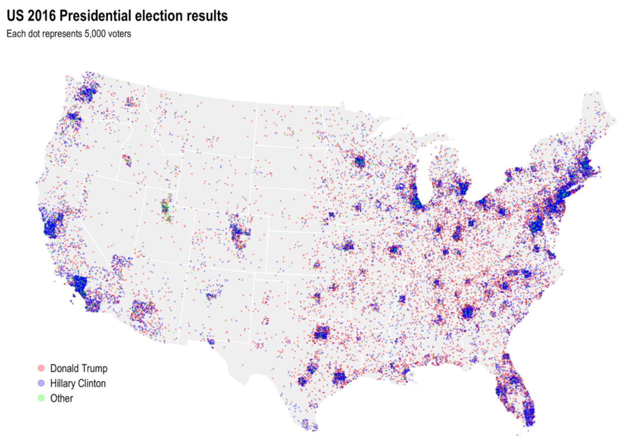


Figure 2.10: Dot density map example
https://www.andybeger.com/2018/05/11/us-2016-dot-density/

Map visualisation enhancements

Data visualisation enhancements are generally extensions to basic visualisations used to improve the user experience. In visualizing maps, or any kind of electoral data there is typically a lot of information to be conveyed - data visualisation enhancements are often concerned with finding ways to display this information effectively without overwhelming the user.

Coloured rings

In the example from the Times UK below, coloured rings are used to display results of a UK constituency over the past 4 elections. This effectively uses channel coding to display a lot of information compactly. This is an interesting technique, as it could be applied to a dot density map to represent a voter's 1st, 2nd, 3rd preference and so on in a PR-STV election. In this scenario, a possible improvement to this technique could be to utilise pop-out channels; setting the 1st preference circle to be larger in order to make it the fixation point.

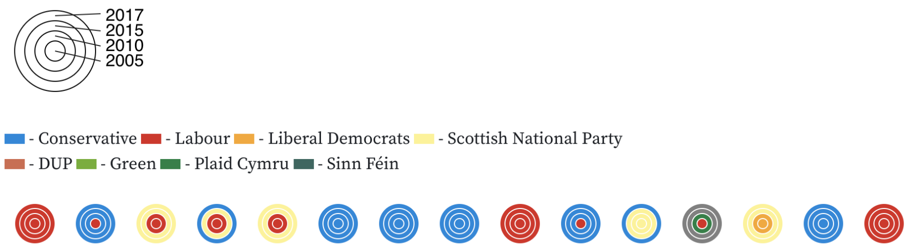


Figure 2.11: Coloured rings example
https://www.thetimes.co.uk/edition/news/elections-results-mapped-60-seats-that-always-back-the-winner-l3zjqjlb9

Pop-ups

Pop-ups are commonly used in map data visualisation to present a wide variety of information. They provide a very simple and effective way to present more detailed information on a map upon user interaction, allowing the user to "drill down". In the example from FiveThirtyEight below, hovering over a US state on the map presents a pop-up bar chart, illustrating the chances of Clinton and Trump winning.

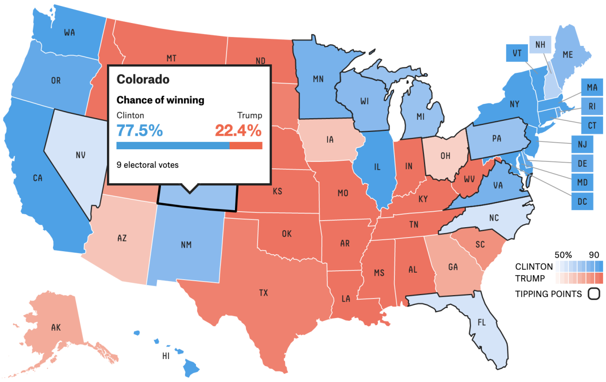


Figure 2.12: Map pop up example
https://projects.fivethirtyeight.com/2016-election-forecast/

Techniques to visualise vote counting

Visualising vote count data (particularly PR-STV) is an interesting challenge as unless you're familiar with the count system it can often be hard to interpret related visualisations. Animation is a helpful technique in this regard as it can be used as a device for storytelling.

Tables

Tables are commonly used to show an overall snapshot of a count in PR-STV elections in Ireland. This example, from the Irish Times election 2020, is an effective example of how a simple table can be used to show how a particular candidate came to be elected. One downside to this method is that all the information is presented at once, and doesn't guide the user through the story.

Candidate		% 1st Pref	Count 1	Count 2	Count 3	Count 4
SF Funchion, Kathleen	✓	23.75%	17,493			
FF McGuinness, John	✓	14.34%	10,558	11,021	11,298	11,622
FF Murnane O'Connor, Jennifer	✓	12.70%	9,351	9,723	9,895	10,093
FF Aylward, Bobby	–	10.25%	7,550	7,791	7,920	8,124
FG Phelan, John Paul	✓	8.69%	6,396	6,498	6,547	6,752
FG Deering, Pat	✗	8.05%	5,929	6,037	6,137	6,323
GP Noonan, Malcolm	✓	6.71%	4,942	5,504	5,647	6,207
FG O'Neill, Patrick	✗	4.99%	3,674	3,723	3,776	3,943
IO Hayes, Alan	✗	3.19%	2,347	2,951	3,383	3,710
LAB Hynes, Denis	✗	3.00%	2,208	2,610	2,712	
PBP Wallace, Adrienne	✗	2.12%	1,558	3,350	3,769	4,059
RI Byrne, Helena	✗	1.35%	992	1,164		
IFP O'Neill, Melissa	✗	0.59%	431	666		
IO Ray, Angela	✗	0.29%	214	331		

Figure 2.13: PR-STV result table example
https://www.irishtimes.com/election2020/carlow-kilkenny

Animated bar chart

Animated bar charts are used to visualise changes in trends over time, particularly where there is a lot of information which has the potential to overload the user if presented all at once. In the below example, The Irish Times effectively utilises this technique to present a count by count visualisation of transfers, eliminations and elections. This presents a much more holistic data story than the table above, and helps guide the user through the counting process.

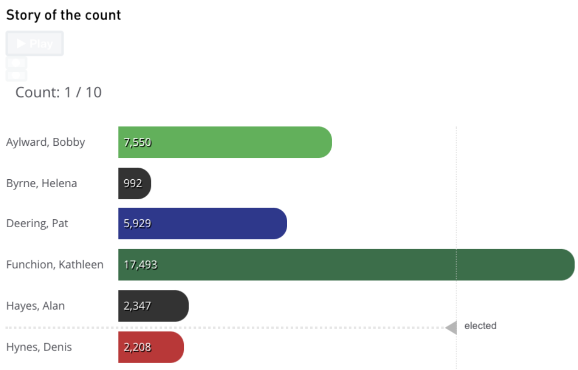


Figure 2.14: Animated bar chart example
https://www.irishtimes.com/election2020/carlow-kilkenny

Techniques to Visualise Election Results

Charts, diagrams and other types of visualisations are commonly used to depict election results. While many different types of visualisations are capable of presenting the same information, the choice of visualisation can have a significant impact on how the information is interpreted by conveying patterns and trends in different contexts.

Charts

Charts such as bar charts, stacked column charts, and pie charts are commonly used to present information such as percentage vote share or seats won. Many can be used to present similar types of information, however, some are favourable in certain contexts. For example, stacked column charts provide a simple way to illustrate the proportions of multiple totals, while a bar chart with a negative and positive axis can be used to illustrate change.

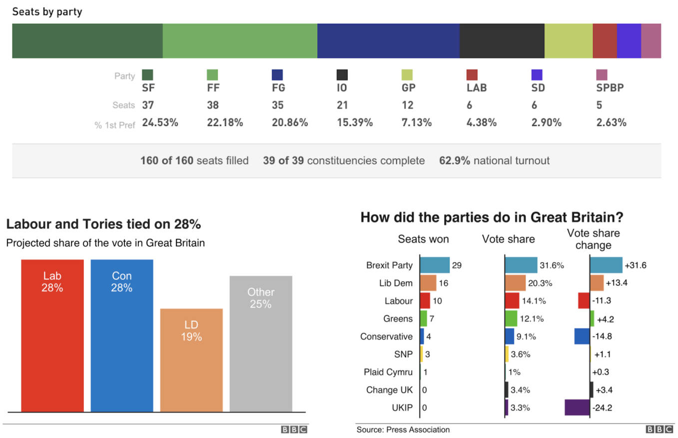


Figure 2.15: Stacked column chart & bar chart examples
https://www.bbc.com/news/uk-politics-48091592
https://www.irishtimes.com/election2020/carlow-kilkenny

Hemicycle

Hemicycle charts are a popular way to visualize election results. The appearance of the chart is similar to a parliament layout, and consequently makes it a great way to visualize the seats held by each political party.

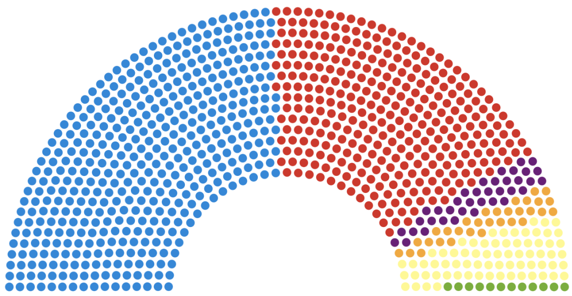


Figure 2.16: Hemi-cycle (parliament) chart example
https://www.bbc.com/news/world-europe-48417191

Sankey diagram

A Sankey diagram is a visualisation used to depict flow from one set of values to another. In the below example, from YouGov, a Sankey diagram neatly visualises where parties gained and lost votes between elections. In the context of this tool, a Sankey diagram could be used to give a quick and intuitive snapshot of PR-STV transfers.

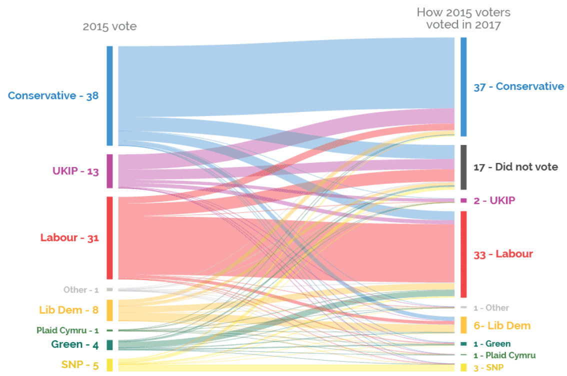


Figure 2.17: Sankey diagram example
https://yougov.co.uk/topics/politics/articles-reports/2017/06/22/how-did-2015-voters-cast-their-ballot-2017-general

2.4 Similar Applications

There is a limited number of similar applications available for teaching this topic, but none that this author identified provided the ability to both draw constituency boundaries and switch between different electoral systems - key tenants of this project. However, there was a range of applications and sites that individually provided useful direction and ideas that would later inform features.

The ReDistricting Game

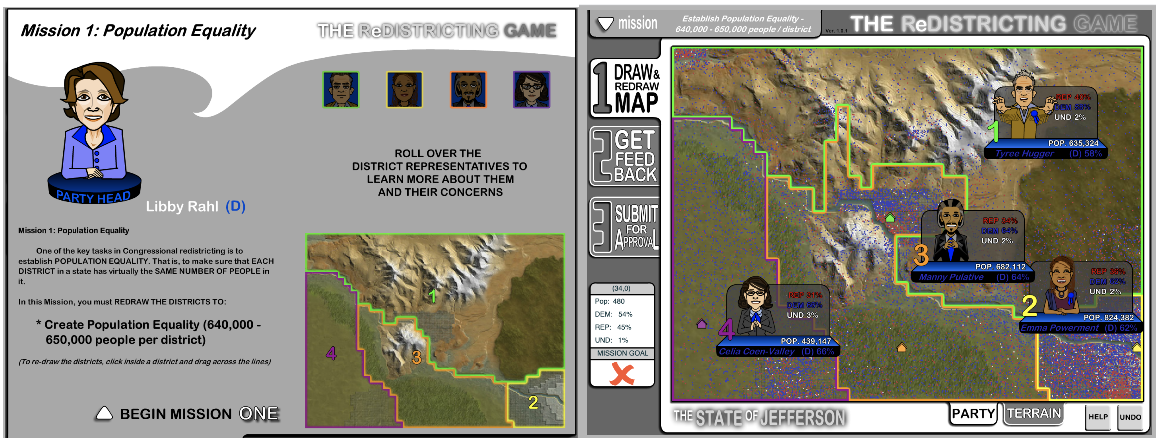


Figure 2.18: Redistricting game start screen & boundary manipulation

URL: http://www.redistrictinggame.org/

Platform: Online Game

Target: US Citizens (No specific age cohort specified)

Overview: The Redistricting Game was designed for USC Annenberg Center for Communication by a large research and development team at USC Game Innovation Lab. Its core purpose is to enable US citizens to engage and educate themselves about political gerrymandering (redistricting) which is prominent in the United States.

Interface: The game is presented as a series of challenges - with varying levels of difficulty - that allow players to explore different forms of gerrymandering. The game is based on the American system, where there are two main parties. Once users select a challenge they are prompted to choose which party to work for. The main element of the game is broken down into 3 steps:

1. Manipulating boundaries: Users are presented with a square-grid map of voters (dot density distribution), which is divided into four areas by district boundaries. These boundaries can be changed by clicking and dragging, and an overview of the percentage share of each party in each district is given as the boundaries are manipulated.
2. Evaluation: This provides users with feedback on how what they have done compares with the desired result for that challenge.
3. Approval: The final step is an extension on the main process, which indicates whether it might be approved by the state courts or legislature.

Evaluation: The application presents the concept of redistricting in a novel way; with challenges, characters and the user encouraged to pick a side to work for. The feedback and the integration of information about reform initiatives are also interesting. However, the user interface takes some time to get to grips with, and the user may become overloaded with information at times. The application could benefit from utilising data visualisation techniques such as the "drill down" framework, to allow the user to explore in great detail on demand. The application is limited in the context of this author's project, given it focuses solely on gerrymandering within a two-party, first past the post system. The application UI and technologies used such as flash, are somewhat outdated and may not be accessible by all in modern browsers.

NYU Gerrymandering Game

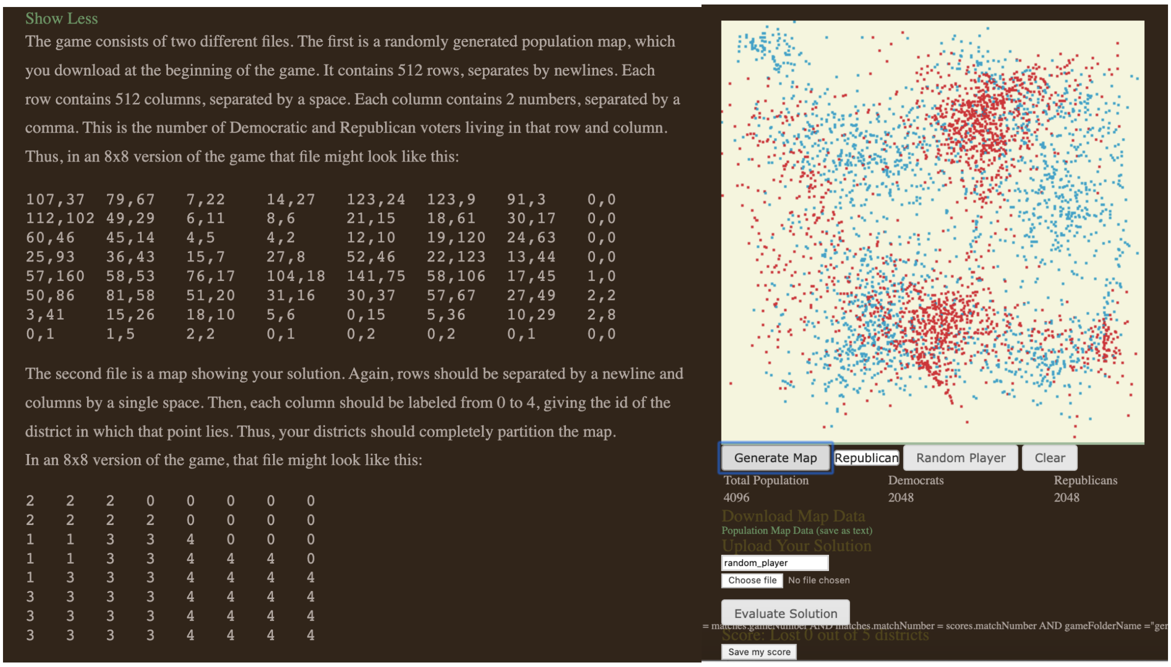


Figure 2.19: NYU Gerrymandering game instructions & game map

URL: https://cims.nyu.edu/drecco/games/gerry/index.php

Platform: Online Game

Target: US Citizens (No specific age cohort specified)

Overview: This game was designed in New York University, with the goal of allowing users to redraw district boundaries and understand how they can maximize a particular party's influence in the US House of Representatives.

Interface: The game is played on a square grid representing a state. Each point on the grid is represented by a blue or red dot, which indicates the number of registered Democrats and Republicans living on that point. Learners are challenged to partition the state into 5 districts to maximise the number of seats their selected party wins. This is achieved by users uploading a text file containing an array of numbers, each corresponding to the desired district assignment. Once the user submits their solution, they are presented with a score indicating the number of districts which vote for the users' desired party.

Evaluation: While the game provides a basic insight into how a state can be gerrymandered, the tool is quite limited in its functionality. The text-based user input to assign district boundaries is arduous, prone to error and not accessible for most users.

Irish Times' Election 2020 Results Hub

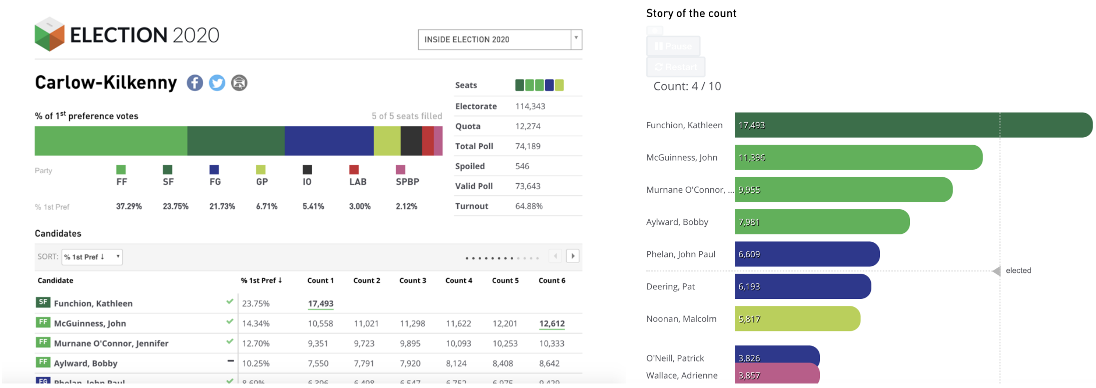


Figure 2.20: Irish Times' Election 2020 Hub

URL: https://www.irishtimes.com/election2020/

Platform: Newspaper website

Target: General Public

Overview: The Irish Times' Results Hub was designed to give its readers an overview of the results for each constituency in the Irish general election in 2020. It aims to provide readers with an insight into the story behind the result with detailed breakdowns of each count under the Irish PR-STV system.

Interface: The homepage of the hub gives an overview of the national picture of the election. Users can then select their desired constituency from a drop-down menu in order to get detailed information about that particular result. Within an individual constituency overview, a user is presented with a range of interactive visualisations such as a bar chart indicating the 1st preference vote breakdown or a table indicating the results for each candidate in each count. Users are also given the option to build a coalition government based on the results in order to allow them to see what the various possible configurations are.

Evaluation: While this tool is based on actual election results and does not allow users to try out varying configurations, this author believes it's useful to examine from a visualisation perspective. The system is very intuitive and allows users to 'dive in' with very minimal effort. The visual aspects of the tool are very appealing, with colour employed effectively to associate candidates and parties throughout. The use of an animated bar chart to illustrate transfers under the PR-STV system between counts is very effective.

New York Times' 2018 House Election Results

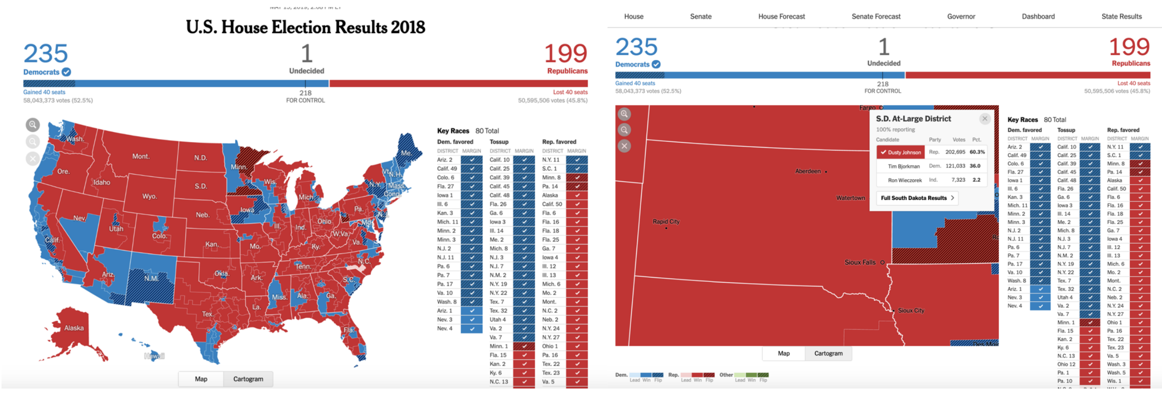


Figure 2.21 New York Times' 2018 House Election Results

URL: www.nytimes.com/interactive/2018/11/06/us/elections/results-house-elections.html

Platform: Newspaper website

Target: General Public

Overview: This New York Times' page was designed to give its readers an overview of the results for each state in the 2018 House Election Results in the United States. It allows readers to understand the story behind the result through detailed breakdowns for each district.

Interface: A stacked bar chart on the homepage illustrates the overall national result between Democrats and Republicans. While a coloured map of the United States is used to illustrate the result within each district, red for Republican victories, blue for Democrat victories. When a user clicks on an individual district they are presented with a popup/tooltip to indicate the candidate and the number of votes they received. From there a user can choose to get a further, more detailed breakdown on the result for that district. Throughout bar charts, maps and tables are used to portray the results.

Evaluation: Again, while this tool is based on actual election results and does not allow users to try out varying configurations, this author believes it's useful to examine from a visualisation perspective. The use of a coloured map is very effective in allowing the user to clearly visualise the outcome of the election in each state and district at a glance. The ability to click on a district to view its breakdown is a very intuitive way for the user to identify a district and understand its results within the context of its state. The use of tooltips when hovering over a district is also effective in providing a summary of the result without having to click into the full breakdown.

2.5 Summary

It is clear that electoral systems can be complex and often difficult to fully understand at first, but that the use of technology, e.g. microworlds, has immense potential to encourage thinking and learning in a manner that is beneficial towards this understanding. There are a limited number of existing applications working towards this goal, and of the existing applications, there are few that could be considered up-to-date and incorporating modern data visualisation techniques. It is also clear that pedagogical theory and data visualisation techniques have the potential for great synergy - microworlds espouse the idea of creating powerful thinking tools, while data visualisations techniques are essential to creating the cognitive loop that makes applications become part of the thinking process.

3. Project Specification

3.1 Considerations

The tool is designed for use by Transition Year (15/16-year-old) secondary school students in Ireland. In that respect, there are a number of considerations which this author has taken into account in the design process.

- Prior knowledge: Transition Year students' will typically have engaged in approximately 100 hours of student hours of Civic, Social and Political Education over a three year period as part of the Junior Cycle curriculum (NCCA, 2016). As part of this curriculum, students get an introduction to democracy, elections and voting in Ireland (PR-STV system) (NCCA, 2016).

- Learning environment: The tool will typically be used within the context of a technology-mediated group project learning environment, however, it may occasionally be accessed by students on an individual basis.

- Similar applications: There are a limited number of similar applications available within this space, and none (that this author has identified) which approach the subject from an Irish context.

- Accessibility: There may be students of varying ability and technological literacy so it is important that the tool is accessible, usable, and convenient for everyone regardless of these factors. Learners should be able to participate fully and the tool should help them to realise their full educational potential.

- Ethics: The tool should avoid politicising the matter so far as is possible, with all activities planned and designed to be of educational benefit to learners - helping them to become more informed citizens.

3.2 Scope

Given the vast nature of electoral systems as illustrated in the previous chapter, it was necessary for this author to place some limitations on the scope of this project within the context of the project's objectives, and the considerations and background research listed previously.

- To avoid overloading the learner, the system will consider just two electoral systems Proportional Representation - Single Transferable Voter (PR-STV) and First-Past-The-Post (FPTP) as representations of the differences between proportional systems and plurality systems respectively. While the party-list system is the most popular form of proportional electoral system, PR-STV is chosen as it is the system used in Ireland for local, national and European elections, and one which many academics have argued to be the best form of proportional system. First-Past-The-Post is chosen as it is the most popular form of plurality system and is used in the United Kingdom and the United States for which students in Ireland are more likely to be familiar with and interested in over other jurisdictions.

- To simplify the comparisons between electoral systems and constituency configurations the system will allow each party, one candidate in each constituency. While there is some benefit in understanding how the number of candidates a party runs affects the outcome of an election, having multiple candidates from the same party in the same constituency is not the norm in single-seat constituencies or FPTP systems. This extra layer of complexity - where learners would have to distinguish between votes for parties as well as votes for candidates within parties - would have the potential to detract from the core learning objectives of the tool by overloading the user. This simplification also greatly assists in the types of visualisations which can be used.

- Similarly to avoid overcomplicating the system, the issue of spoiled votes will be ignored as it would contribute very little to a user's overall understanding of electoral systems.

3.3 Functional Requirements

Constituency and Electoral System Configuration:

- The user must be able to view a set of constituencies.
- The user must be able to view a distribution of voters across the set of constituencies.
- The user must be able to switch between single-seat and multi-seat constituency configurations.
- The user must be able to manipulate constituency boundary configurations.
- The user should be able to access an overview of each constituency which provides information about the number of seats, quota and breakdown of votes in that constituency.
- The user must be able to switch between PR-STV and FPTP electoral systems.

Election Results:

- The user must be able to run an election based on a given constituency and electoral system configuration.
- The user must be able to view the number of seats each party receives in a given election (the result).
- The user should be able to view the Gallagher index for each result.
- The user should be able to view a breakdown of the result at a constituency level.
- The user should be able to visualise the counting process at a high level.

3.4 Non-functional Requirements

Availability and reliability:

- The user must be able to access the system from a standard PC or laptop device.
- The user should be able to access the system from a touch screen tablet device with a suitable screen resolution.
- The system should support at least the 4 most used modern browsers (Chrome, Firefox, Safari, Edge).
- The system should be scalable and capable of supporting multiple groups using it concurrently.

Documentation:

- The system should provide enough basic information and context throughout to allow students to explore and construct their understanding of electoral systems by diving in.
- The system should provide users with a quick-start guide to help them get started within minutes.

Usability:

- The system should utilise the principles of universal design to support a wide range of users.
- The system should be of an appropriate level for Transition Year students in secondary school.
- The system should allow the user to quickly and easily visualise changes they have made to a given election configuration in real-time.
- The system should allow users to effectively visualise how a given electoral system works.
- The system should be effective in a cooperative learning environment.
- The system should allow users to quickly and effectively construct an artefact which can be shared and discussed with peers.
- The system should be intuitive and provide users with feedback on interaction to assist them in understanding how to use the tool.
- The system should effectively utilise scaffolding techniques to increase the zone of proximal development.
- The system should effectively utilise dual coding theory to help learners understand the information better.

3.5 Initial technology choices

Taking into account the functional and non-functional requirements of the tool, the development of a web application was considered to be the most appropriate form for the tool to take. A web application makes it convenient for the users to access the tool from any location with very little commitment, and as such can be easily deployed in a classroom setting across multiple devices and platforms. It also allows learners to access the tool from home. The main requirement or barrier to access is a device with an internet connection. Indeed, the possibility of a poor internet connection. The trend toward web applications and away from desktop applications in the last decade has been positive as it allows for the use of rapid development in applications such as this considerably more. With this growth, much of the development of new data visualisation and graphical user interface technologies have focused on web applications, therefore making it the obvious choice for this project.

In developing a web application, there were a number of technologies required.

HTML, CSS & Bootstrap

HTML (Hypertext Markup Language) & CSS (Cascading Style Sheets) are ubiquitous technologies used in web development. HTML underpins the content and structure of a web-page, while CSS is responsible for the design or styling of the web application.

Bootstrap is an open-source toolkit for responsive, front-end web development with HTML, CSS, and JavaScript. It was chosen for its reusable CSS and JavaScript components, which allow for the rapid development of mundane but critical functionality such as buttons, forms etc. allowing for work to be focused on the unique functionality of the web application. The responsive nature and popularity of these components also increases the web application's compatibility and familiarity across devices and platforms, as per the non-functional requirements. Additionally, as Bootstrap is open source, all of its out-of-the-box components can be freely adapted as needed. Alternative CSS frameworks such as PureCSS and Foundation were also considered, however, Bootstrap has a much more extensive list of components and better community support given its ubiquity.

Javascript

The third layer of a standard web topology is JavaScript. JavaScript is an object-orientated programming language, and enables the implementation of complex features and interactivity on web pages. Javascript was chosen for a number of reasons:

- It is the most common option for user interface interactivity used across the web today, meaning it is supported across platforms.
- It provides common programming features that allow for the integrated implementation of functionality, such as vote counting, with rapid client-side execution.
- It has a robust range of libraries and frameworks available for visualisation and interactivity.

Firebase

Google's Firebase is an all-in one-platform with hosting, storage, and analytics. Firebase provides production-grade web content hosting, with the application cached on solid state drives at content delivery network edges around the world, ensuring it can scale as necessary and is easily and reliably accessible.

Firebase is also ideal in a rapid development scenario, such as this project, as it enables new versions to be deployed or rolled back in one command, and new versions to be pushed out based on user feedback. Additionally, Firebase also seamlessly integrates with Google analytics and provides useful information on devices and browsers used to access the tool, which allows for the prioritisation of testing.

Alternatives such as Amazon's AWS and Microsoft's Azure provide similar solutions although Firebase was chosen due to this author's familiarity with it and the integration of a range of Google services which allowed this author to focus on the development of the app, without having to manage the infrastructure.

3.6 Data visualisation technology choices

This section builds on the initial technology choices and presents an analysis of a range of JavaScript libraries and frameworks for data visualisation. Each of the libraries is examined on a number of factors, in particular their ability to provide the visualisation techniques presented previously. The selection criteria also includes the flexibility of the library to create bespoke visualisations, its compatibility across devices in order to ensure accessible use, and the practicality of using it within the project constraints (i.e. its learnability & community support).

D3.js

D3.js is a JavaScript framework for manipulating documents based on data. It is built on modern web standards, using HTML, SVG and CSS. There are a number of similarities between D3 and other DOM frameworks such as jQuery, however rather than the developer directly manipulating elements, the developer provides data to D3 and defines callbacks, and D3 manipulates the elements. D3 dynamically generates visualisations by binding data to elements (such as shapes) in the Document Object Model (DOM). If data or elements are left over after computing the data join, these are available in the enter (no corresponding nodes) and exit (no corresponding data) sub-selections. This enables exact data-element correspondence which is ideal for efficient dynamic visualisations (Bostock et al., 2011).

- Almost any type of visualisation is possible, which is good for bespoke applications
- It is easy to add a vast range of fluid interactivity
- Scalable and flexible framework with a diverse collection of official and community-developed modules allow for different types of visualisations.
- Active open-source community with a large repository of examples.
- A preferred choice amongst large publishers such as the New York Times.
- Steep learning curve.

P5.js

P5.js is a JavaScript library based on the principles of the popular Processing (Java-based) language to make coding accessible for artists, designers, educators, and beginners. Each 'sketch' or HTML canvas in p5.js is made up of one setup function, where the basic outline of a sketch is set out, and a single draw function, where you draw to the canvas (McCarthy, et al, 2016).

- Wide range of visualisations possible.
- Possible to add interactivity.

- Utilises general programming concepts from Processing which allow the programmer to go beyond drawing (such as using sounds or the webcam etc).
- Supported by the already existent Processing community.
- No built-in rendering, you have to write all the rendering code for any objects you include, integrating it appropriately within the drawing loop.
- Relatively easy to get started.

Google Chart Tools

Google Chart Tools is a library that creates graphical charts from user-supplied data. The service is powerful and simple to use within Javascript, and is great for rapidly generating common visualisations.

- Common visualisations can be generated quickly and easily.
- Possible to add a limited amount of interactivity.
- Supported and maintained regularly by Google so good compatibility across the web.
- Less suitable for bespoke visualisations that require a great deal of flexibility.
- Very little background knowledge needed.

The table in figure 3.1 below provides a summary comparison of the libraries presented across the range of selection criteria.

Library Comparison

Javascript visualisation libraries

Library	Map visualisations	Vote count & Result visualisations	Extensibility	Compatibility	Rendering	Open-source	Learning curve
D3.js	All techniques explored are possible with lot's of community examples within election contexts. Easy to extend visualisations to include bespoke features and interactions.	All techniques explored are possible and highly flexible with a vast range of modules and community examples.	High	Modern browsers (everything except IE8 and below)	SVG elements, greater flexibility in manipulating each element.	Yes	Steep, but quick and easy to use once comfortable with the approach.
P5.js	Most techniques explored are conceivably possible but there is lack of previous examples to build on.	All techniques explored are possible, however there is a limited number of modules or previous community examples available so they would need to be built from scratch.	Medium	Modern browsers (everything except IE8 and below)	HTML5 canvas elements, canvas is a lot faster than SVG (although less flexible).	Yes	Average, quick to get started.
Google Charts	Limited range of map visualisation techniques available, all centered around markers layered on a Google Map.	Vast range of pre-built charts available, however there is limited flexibility. Visualisations such as hemi-circles or animated bar charts are not possible.	Low	Cross-browser (including VML for older IE versions)	SVG elements, greater flexibility in manipulating each element.	No, API is provided free and minified.	Easy, very little background knowledge needed.

Figure 3.1: Visualisation libraries comparison table

Choice of framework

D3, P5 and Google Charts are some of the most popular libraries for data visualisation in javascript, and more broadly across the web. Based on the analysis of these libraries presented above, D3.js was chosen for the development of this tool for a number of reasons:

- D3 has a large community, with a wide range of examples from its author and others - far outnumbering what is available for the other libraries analysed. Moreover, D3 is the media industry's standard for high-end data visualisation - there is a strong foundation of electoral visualisations to build on.
- D3 avoids proprietary representation and affords great flexibility and compatibility, exposing the full capabilities of web standards such as HTML, SVG, and CSS.
- While D3 can be tricky initially, its data manipulation and binding approach can easily generate complex visualisations from large data sets.
- D3.js offers limitless possibilities for interactivity, with various listeners and its exact data-element correspondence.

- As D3 is data-driven, visualisations it creates are extensible; this allows for the addition of other electoral systems in the future by simply changing the dataset.

D3 in more depth

A typical approach to visualising data with SVG using Javascript or jQuery would be to iterate through a data set and generate an SVG element to be inserted into the DOM. To update a visualisation then, old SVG elements would have to be compared to remove, update or insert changes. This is relatively complex, as old data would need to be stored in the DOM, with ids, in order to compare it with new data.

The data() method binds the data to document elements; for example to create circle elements, you can specify that the selection "circle" should correspond to data. By default the first element is bound to the first data point, and so on. when the data is updated, D3 determines which SVG elements need to be updated. If data or elements are leftover after computing the new data-join, these are available in the enter (no corresponding nodes) and exit (no corresponding data) sub-selections allowing the specification of future use. D3 then handles the complicated process of updating the DOM (Bostock et al., 2011).

D3 also has a number of other core methods such as Selections and Append which simplify the imperative approach of W3C DOM API. This enables exact data-element correspondence, which is crucial for efficient dynamic visualisations, as required in this project. Additionally, there are numerous built-in methods for setting attributes or styles, registering event listeners, adding, removing or sorting nodes, and changing HTML or text content directly in the DOM (Bostock et al., 2011).

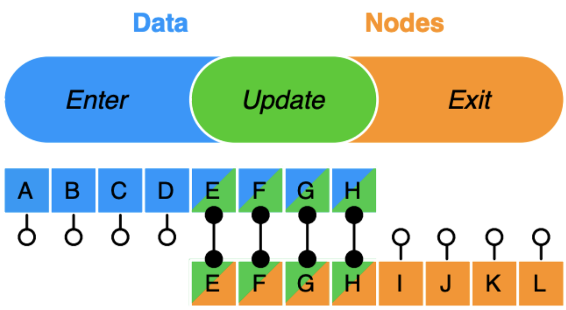


Figure 3.2: Enter, update, exit
When new data (blue) are joined with old nodes (orange), three subselections result: enter, update and exit.
(Bostock et al., 2011)

colour	xco	yco
blue	20	25
red	40	25
green	60	25

```
//Create SVG container
var svg = d3.select("body").append("svg")
.attr("width", 200)
.attr("height", 50);

//Bind data
svg.selectAll("circle")
.data(circledata)
.enter()
.append("circle")
.attr("cx",function (d) { return d.xco; })
.attr("cy", function (d) { return d.yco; })
.attr("r", 10)
.style("fill", function (d) { return d.colour;});
```

```
//HMTL Output
<svg width="200" height="50">

<circle cx="20" cy="25" r="10" fill="blue" />
<circle cx="40" cy="25" r="10" fill="red" />
<circle cx="60" cy="25" r="10" fill="green" />

</svg>
```

Figure 3.3: D3 bind data & HTML output sample
Circle attribute data from the table is bound to a circle selection using
D3 which produces the corresponding HTML to be injected into the DOM.

3.7 System architecture

This section considers a range of possible system architecture approaches for rendering the application. The criteria for selection include responsiveness, load-times, and compatibility with existing technology choices such as D3.

Client-side rendering (CSR)
In a client-side approach, a server provides a simple HTML document (with linked styles and scripts) to the browser, and the rest of the web application is then dynamically generated by JavaScript.

- Allows for a responsive application with rich interactions.
- Applications are generally fast, after the initial load time.
- A static front-end can be hosted via a content delivery network to improve load times.
- Creates a smaller load on the server.
- Easy to deploy.

Traditional server-side rendering (SSR)

In traditional server-side rendering, the server responds to a browser request by generating and then returning a fully-rendered page.

- Fully-rendered HTML is returned as soon as the initial request returns a response.
- The server responds by generating and returning a completely new page for every interaction which can be slow but is not an issue for static-sites.
- Creates a bigger load on the server.
- Allows for the user of various programming languages to be executed on the server-side.
- Better search engine optimization.

Hybrid rendering

Another popular technique for rendering is a hybrid of the client and server-side approaches, where the server sends a fully rendered page to the client and then lets the client's javascript bundle take over.

- Fully rendered HTML as soon as the initial request returns a response. Then leverages the capabilities of CSR.
- Additional requests to the server return fully rendered HTML as required.
- Complicated setup.
- Creates a bigger load on the server.

Choice of architecture

Following analysis of a number of approaches, a simple client-side rendering approach was chosen for a number of reasons:

- Interactivity is a core component of this application, and a client-side approach allows for a more responsive application.

- In D3, all the data transformations and renderings are done client-side in the browser to support rich user interaction, and therefore works best within a client-side architecture
- The hybrid rendering approach, while beneficial in terms of initial load time, comes with the large and complex overhead of weaving D3 into a server-side framework
- A static front-end can be hosted via the Firebase content delivery network (CDN) which delivers improved performance
- As client-side rendering removes load from the webserver this also results in improved scalability

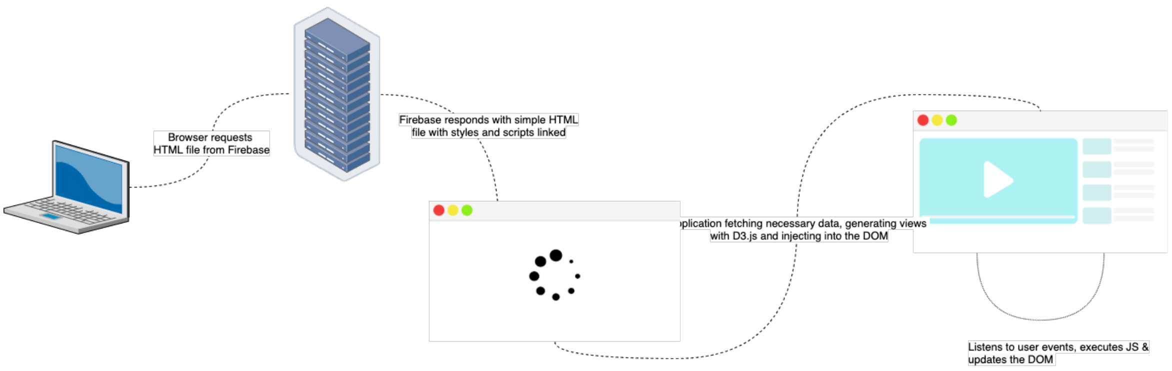


Figure 3.4: Basic application architecture diagram

4. Implementation

4.1 Problem approach

From an implementation perspective, it was helpful to decompose the problem into four core components; data, data analysis & transformation, graphical interpretation, and user interaction. This strategy draws on common steps in data visualisation (Fry, 2007) and D3's data-driven approach. By breaking the system implementation down in this way, it provided a useful framework to approach each problem and make it more manageable. Utilising this strategy also helps to build in a degree of extensibility, by enabling the solutions to be generalised.

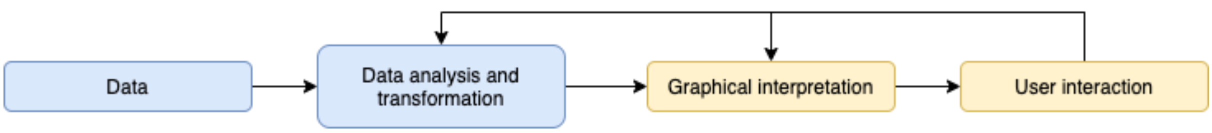


Figure 4.1: Problem decomposition diagram

4.2 Constituency Manipulation

Constituency manipulation is one of the core components of the tool, and was one of the most challenging aspects of the project, as many of the existing solutions such as the tile methods (where a boundary is manipulated tile by tile) are slow and tedious. This not only affects the user experience but places limitations on the types of scenarios that can be created, e.g.the number of constituencies that remains feasible to manipulate. As a result, defining this experience was one of the first steps in the design and implementation of the application.

Based on the analysis of the various visualisation techniques previously presented, the Voronoi method was chosen as the basis of this component, as it allows for more visually attractive and fluid interaction than the other techniques analysed.

One of the first benefits of using the Voronoi method is given a set of sites in a space, it partitions that space into cells. This means boundaries lines are not manually specified, which is useful when switching between single and multi-seat constituencies. It allows for the easy addition (in the case of single-seat) and removal of constituencies (in the case of multi-seat) by simply specifying more or fewer sites.

D3 implements Fortune's sweep line algorithm for computing the Voronoi diagram of a set of two-dimensional points (x, y). The first step was to generate the site data, in the form of a JSON object with points and a name. These were initially set arbitrarily and then adjusted following experimentation in order to produce relevant electoral scenarios based on voting data. `d3.scaleLinear()` is used to transform the data values into visual variables based on the width and height of the space. These variables are then inputted into `d3.voronoi()`, which executes the Voronoi tessellation operation and produces an array of polygons.

```
//Data
[{
    "x": 241,
    "y": 300,
    "name": 0
  },
  {
    "x": 585,
    "y": 174,
    "name": 1
  }
...
]
// Scales
x = d3.scaleLinear().domain([0, 100]).range([0, width]);
y = d3.scaleLinear().domain([0, 100]).range([height, 0]);
xr = d3.scaleLinear().domain([0, width]).range([0, 100]);
yr = d3.scaleLinear().domain([height, 0]).range([0, 100]);
```

```
// Voronoi tesselation
var voronoi = d3.voronoi()
  .x(function(d) { return x(d.x); })
  .y(function(d) { return y(d.y); })
  .extent([[0, 0], [width, height]]);
```

Figure 4.2: Voronoi tessellation code snippet

Once the array of polygons has been generated, this dataset and the original points are bound to "path" selections and "circle" selections respectively, using D3 data() method. By default, the first element is bound to the first datum and so on. This generates an SVG element which is then inserted into the DOM by D3.

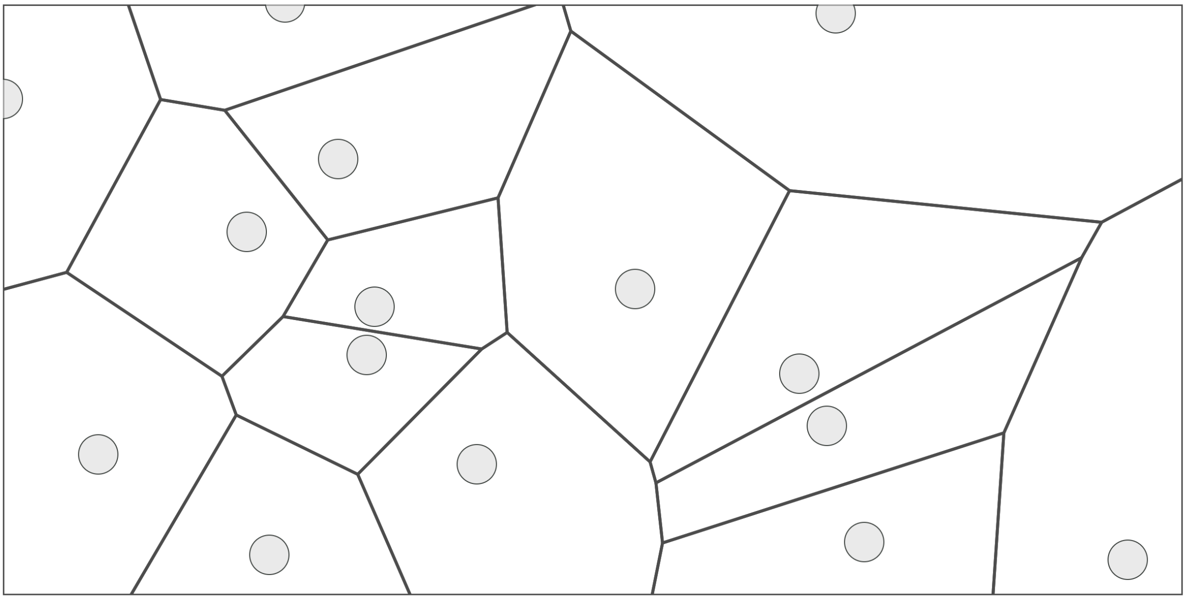


Figure 4.3: Output SVG: Voronoi diagram (constituency map)

This generates a standard static Voronoi diagram, and the next step is to add interactivity to allow the manipulation of boundaries. Each Voronoi site (represented by circles) can be thought of as a polling centre, with each constituency (cell) consisting of all the voters closest to its polling centre. It follows that the boundaries of a constituency can be manipulated by the movement of a polling centre

The translation of this idea into D3 is relatively simple, through the use of built-in drag behaviour which enables drag-and-drop interaction on selections. By adding a `.call(d3.drag())` listening event on the Voronoi site (circle) selection, a function can be triggered to update the coordinate data based on mouse location. The Voronoi diagram is then regenerated based on the updated site. To make this a fluid process, a `.transition()` selection is used on update to smoothly interpolate the DOM from its current state to the desired target state over a given duration.

A key consideration here was perfecting the size and aesthetic of the circle sites; the goal was to not let them dominate the whole map, have them be obvious and easily accessible for the user. This was achieved through experimentation and trial and error, eventually settling on a mid-size circle with reduced opacity, preventing the obstruction of any visualisations underneath. An additional benefit of the D3 drag behaviour from an accessibility point of view is the automatic unification of mouse and touch input, and the avoidance of browser idiosyncrasies.

Figure 4.4: Boundary manipulation flowchart

On drop, after dragging a polling site, an event is triggered to re-calculate the election and provide the user with real-time feedback on the changes they have made. This will be explored further in more detail in the following sections.

4.3 Voter data generation and visualisation

The next phase of designing the map component involved choosing a technique for displaying voter distributions and generating the data set behind it.

The Voronoi method has no clear built-in method for displaying voter distributions, such as the square or hexagon tile methods explored previously. As such, it was necessary to look to the dot density, proportional symbols and other techniques.

While the proportional symbols technique provides an interesting way of illustrating different population densities, this extra layer does not add any real value in the objectives of the application, so the dot density technique method was chosen for its simplicity.

Using this method, each voter is represented by individual dots on the map. To allow interpretation of this visualisation at a glance, each party was associated with a colour. This allows a user to see a snapshot of voting preferences in each constituency. When selecting the colours, it was important to keep accessibility (e.g. red-green deuteranopia) and political considerations (e.g. there is an actual Green Party so best not to use that colour) in mind; the final colours selected were Blue, Red, Yellow, Coral, and Pink.

To simplify this visualisation, variations in population density were excluded as it yielded little additional value towards the application objectives. The result is a design with a roughly homogeneous distribution of dots across the entire map area.

When developing the dataset, it was necessary to generate a JSON set of points (x,y) and preferences for each voter. Similar to the Voronoi diagram, this dataset is then bound to a selection (in this case "circle") which causes D3 to generate an SVG element of circles,based on their points. The first circle is associated with the first data point, and so on; in this way, `d.color;` references the colour specified in the first data point and colours the circle by voter preference.

```
//Data
var voterdata = [
  {
    "id": 0,
    "xco": 12.95,
    "yco": 17.42,
    "color": "#128240",
    "pref": ["coral", "blue","pink","yellow","red"]
...
  },

//Selection and binding
svg.selectAll("circle")
    .data(voterdata)
    .style("stroke", "#000000")
    .style("fill", function(d){ return d.color;})
    .style("stroke-width", 1)
    .attr("r", 7)
    .attr("cx", function(d){ return d.xco; })
    .attr("cy", function(d){ return d.yco; })
    .on("mouseover", toolTip.show )
    .on("mouseout", toolTip.hide);
```

Figure 4.5: Voter dot generation

In the interest of efficiency, the design also needed to encompass a way to view a voter's 2nd, 3rd, 4th preferences etc. The coloured ring technique was considered initially, but this was thought to overwhelm the user with too many colours. Instead, using the `mouseover` event in D3, a tooltip is triggered which lists a voter's preferences based on the data-element association.

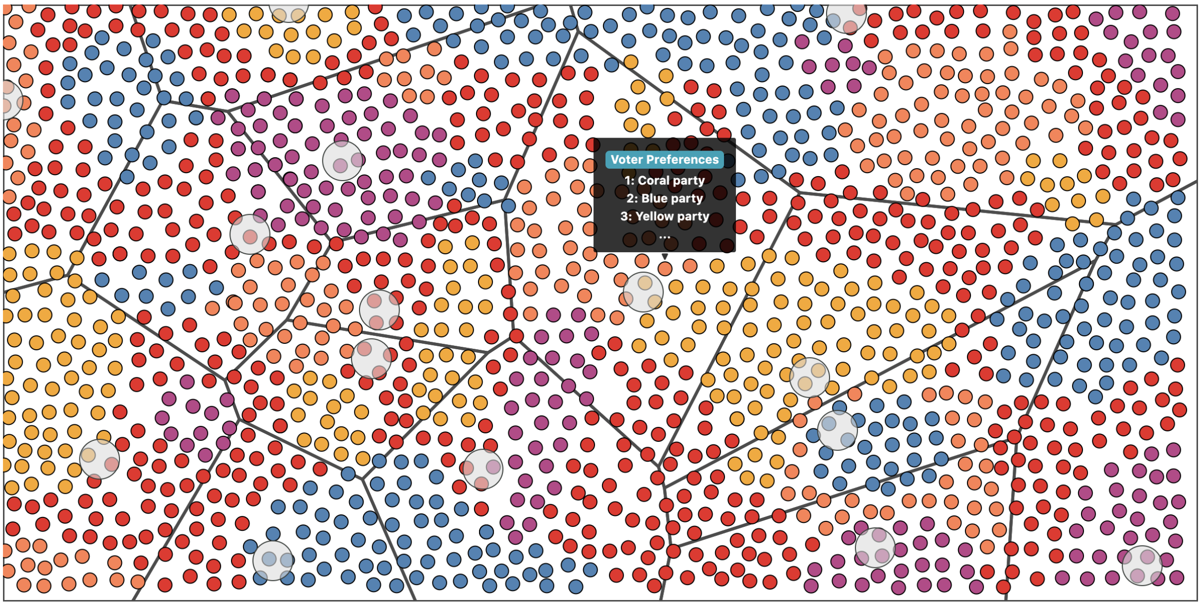


Figure 4.6: Final map

Initially, all of the 1st preference votes were generated randomly, with the specification that each party received a given percentage of the vote, however, it quickly became apparent with experimentation that this method was not ideal, as it led to unrealistic scenarios. For example, there needed to be clear clusters of like-minded voters to guide the user to include or exclude certain voters from a constituency to achieve optimal gerrymandering.

While this helped to address the issue, the bigger challenge came in specifying subsequent preferences. In real elections, some parties are more transfer-friendly, and this can often make the difference between winning or losing a seat. It was therefore necessary to build this phenomenon into the data set to emphasise the differences between PR-STV and FPTP.

This was achieved through several parameters (set based on theory and experimentation):

- In 40% of cases, ballots were randomly selected from all possible permutations of preferences, given a predetermined first preference.
- In 60% of cases, ballots were randomly selected from a predetermined set of preference permutations, given a predetermined first preference.
- The predetermined permutations in the second case build in the following transfer patterns:
    - All blue voters give their second preference to coral and vice versa.
    - All yellow voters give their second preference to pink and vice versa.
    - Red voters give their second preferences randomly and will not receive any second preferences. (Red has the most first preferences, however, by making it less transfer-friendly this can help demonstrate the effects of PR-STV)
    - Subsequent preferences (3, 4, & 5) are specified at random.
- An assumption that voters vote all the way down the ballot was made.

Finally, before the implementation of voter counting methods, it was necessary to associate voters with constituencies based on what polygon cell each voter (x,y) is contained within. This is implemented using `d3.polygonContains` which returns true if and only if the specified point (voter) is inside the specified polygon. In this way, an array of voter ids is generated for each constituency, and these voter ids can be used to access a voter's preferences or any other relevant information associated with a voter.

```
  //Associate voters with constituencies
   for(var i = 0; i < polygonCells.length; i+=1){
      var temp = [];
      for(var j = 0; j < voterdata.length; j+=1){

      var bool = d3.polygonContains(polygonCells[i], [voterdata[j].xco,
                 voterdata[j].yco]);
      if(bool == true){
          temp.push(voterdata[j].id)
      }
    }
          constituencyVoters.push(temp);
}
```

Figure 4.7: Code snippet - associating voters with constituencies

4.4 Vote Counting

When counting votes, the two methods chosen were First-Past-The-Post (FPTP) and Proportional Representation Single Transferable Vote (PR-STV), as specified within the scope of the project. Vote counting is implemented on a constituency-by-constituency basis - each time the method is called, it is passed an array of voter IDs which make up a particular constituency.

Executing FPTP on this dataset is a relatively simple operation. Using the set of voter IDs, the first preference of each voter can be accessed and a count for each party maintained in a JSON array.

```
    //First count
    for (var i = 0; i < votes.length; i++) {
```

```
    if (!candidates.some(cand => cand.name === voterdata[votes[i]].pref[0])) {

      candidates.push({
        name: voterdata[votes[i]].pref[0],
        count: 1,
      });
    }

    else { candidates[candidates.findIndex(cand => cand.name ===
           voterdata[votes[i]].pref[0])].count++;
    }
  }
```

Figure 4.8: Code snippet - first count

Once the FPTP count is finished, the top candidates are elected according to the number of seats available within the constituency. In a single-seat constituency, the candidate with the most votes is elected, in a two-seat constituency the candidate with the most votes and the candidate with the second-most votes is elected, and so on. The elected candidates are recorded in a JSON array.

```
//Record elected candidate
      elected.push({
        name: candidates[i].name,
        seats: 1
      });
```

Figure 4.9: Code snippet - recording elected candidates

The implementation of PR-STV is significantly more complex, as it involves multiple counts, distributions, eliminations, and surplus reallocations. In this respect, an important factor to consider before implementing this method was how each of these steps could be captured to provide a user with an understanding of how the PR-STV system works. For example(in contrast to FPTP) a candidate may be elected for different reasons, such as reaching the quota or by being the highest remaining candidate when there are no more candidates to be excluded. As a result, it was essential to structure the counting method into stages based on the Department of Housing, Planning & Local Government's "A Guide to Ireland's PR-STV Voting System".

In the first count, all votes are counted according to first preferences, as described under FPTP. A quota is then calculated using the Droop method based on the total number of votes ($\frac{Total\ Valid\ Poll}{Total\ Seats + 1} + 1$). Any candidate receiving first preferences equal to or exceeding the quota is deemed elected, and this is recorded in a similar manner to FPTP but with the addition of the reason for election e.g. reaching the quota.

The second and all subsequent counts of a PR-STV election involve either the distribution of the surplus of an elected candidate or the exclusion of the lowest candidate(s) and the redistribution of their votes.

This presents several challenges: in real-life counts, ballot papers are usually separated initially into bundles based on the candidates' first preferences. When a candidate is eliminated, each of their ballots is transferred to another candidate's bundle based on the second-preference on the ballot. This results in bundles where a candidate may have a "1" or "2" beside their name on the ballot but the candidate whose bundle they are in is considered the owner of the votes. This process continues through subsequent preferences.

Direct implementation of this method was attempted initially, as in real-life scenarios it is quite efficient. However, in translation the method rapidly becomes convoluted, requiring many additional methods to manage the distribution of votes in later counts. In programming, with the luxury of being able to edit a ballot and re-count all ballots in negligible time, the implementation can be simplified by translating the method indirectly.

The idea centres on adjusting a voter's recorded list of preferences so that their second preference becomes their first preference when their current first preference is eliminated, thereby 'transferring' the vote. For example, in this array representing a voter's preference `"pref": ["coral", "blue","pink","yellow","red"]`, if Coral was deemed eliminated, then all arrays (voter preferences) containing Coral at zeroth index would be shifted. Shifting removes the element at the zeroth index and shifts the values at consecutive indexes down, thus blue would now become the first preference `"pref": ["blue","pink","yellow","red"]`. This allows the next count to be run again using the exact same method as the first count. The key difference is that rather than adding the number of transfers to a candidate's previous count, all votes are recounted as if it was the first count. To calculate the total number of transfers a candidate receives in a count, the previous count can be subtracted from the current count. Although this may seem inefficient, it reduces the overall complexity of the algorithm and allows methods to be reused.

```
voterdata[votes[j]].pref.splice((voterdata[votes[j]].pref.findIndex(voter => voter ===
candidates[i].name)), 1);
```

Figure 4.10: Code snippet - excluding candidates

This handles transfers of votes when a candidate is eliminated, but a slightly varied approach is required when a candidate is deemed elected. When a vote is used to elect a candidate, it cannot be reused to elect another candidate unless a candidate receives a surplus of votes. In that case a proportion of a candidate's votes equal to the surplus can be re-distributed and used by another candidate. Rather than shifting the current first preference out of the array, the current preference is replaced with an `"elected"` or `"surplus"` string. Ballots with the first preference `"elected"` are ignored and thus not reused. While ballots marked as surplus are redistributed when the condition for redistribution of a surplus is reached. In this case `"pref":`

`["surplus", "blue","pink","yellow","red"]`, `"surplus"` is shifted in the same manner as the elimination of a candidate, thus becoming `"pref": ["blue","pink","yellow","red"]`.

Once this foundation was laid, each of the PR-STV counting stages were implemented on that basis:

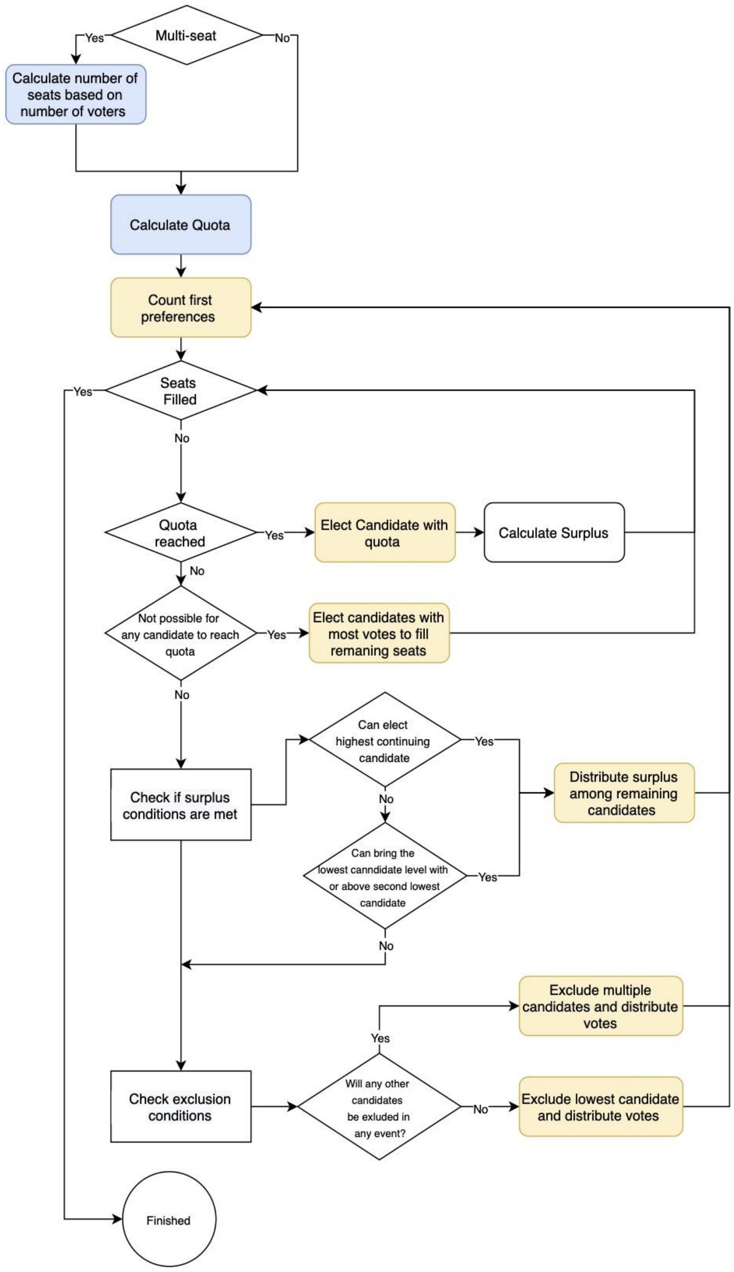


Figure 4.11: PR-STV count - implementation flow chart

The yellow rectangles in the above flow diagram, indicate data or actions that recorded every count where applicable; the total votes for each party, why and how many candidates are excluded, if a surplus is distributed, if a candidate is elected with a quota and if a candidate is elected with quota. Additionally, the blue rectangles indicate data that is recorded for each constituency during the count. All of this data is compiled into JSON arrays, which can then be used to power visualisations that capture the story of the election.

```
//Count Sample Output
[
  {
   "constituency": 1,
   "countNo": 1,
   "name": "Blue Party",
    "value": 37,
    "lastValue": 0,
    "color": "#313190",
    "seats": 2,
    "quota": 54
  },
  {
    "constituency": 1,
    "countNo": 1,
    "name": "Red Party",
    "value": 41,
    "lastValue": 0,
    "color": "#fe0300",
    "seats": 2,
    "quota": 54
  },
...
]

//Count Explanation Sample Output
[
{
    "countNo": 1,
    "constituency": 3,
    "comment": "Yellow is the lowest candidate and is deemed excluded.",
  },
  {
    "countNo": 2,
    "constituency": 3,
    "comment": "Coral is the lowest candidate and is deemed excluded.",
  },
...
]
```

Figure 4.12: Code snippet - Count method sample output

4.5 Gallagher Index

Once an election is complete, the Gallagher index of the election is calculated by getting the square root of half the sum of the squares of the difference between the percentage of votes received and percentage of seats for each party (Gallagher, 1991).

This is later presented to the user as part of the election result in order to help them understand the relative disproportionality of the election.

```
function calculategallagherindex(popVotePercentage, seatsPercentage){

var sum = 0;

for (var i = 0; i < noParties; i++) {

  var difference = popVotePercentage[i].percent - seatsPercentage[i].percent

  difference = difference * difference

  sum += difference

}

 gallagherIndex = Math.sqrt((0.5 * sum))

return gallagherIndex
}
```

Figure 4.13: Code snippet - Gallagher index calculation

4.6 Live count (Animated Bar Chart)

Visualising count data was one of the more difficult tasks, as there is a lot of it in the case of PR-STV. Count data is commonly presented in tables in order to illustrate the result of each count. However, this approach requires the user to have an understanding of the information being presented. A more pedagogical approach to this issue is to visualise changes to a candidates vote over time through the affordance of animated bar charts. This method gives the user a scaffolded guide to the counting process with a count by count visualisation of transfers, eliminations and elections.

To begin, Mike Bostock's (D3 founder) bar chart race implementation - animating the value of the top global brands from 2000 to 2019 - was used as a foundation. In this example, there are four main components: the bars, the x-axis, the labels, and the ticker showing the current date. The animation iterates over each of the keyframes, delegating updates to each of the chart components and awaiting the transition's end (Bostock, 2019).

Several basic changes were made to the implementation to adapt it to illustrate the count; the ticker changed to display the current count (i.e. Count 1, Count 2 and so on) and the bars and labels adjusted to maintain the correct party colour association. On each iteration of the ticker, data output from the Counting method is filtered to retrieve each party's vote for a particular count, and this is bound to relevant components. Similar to previous visualisations, a `.transition()` selection is used on update to smoothly interpolate the DOM from its current state to the desired target state over a given duration.

```
//Iterate through results for each count of a constituency
  let ticker = d3.interval(e => {
```

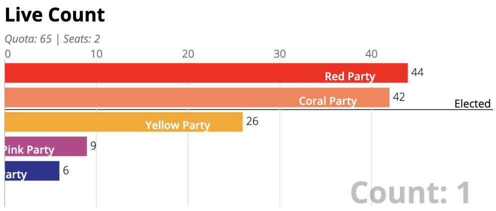
```
        //get data of the same count
        countSlice = data.filter(d => d.countNo == countNo && !isNaN(d.value))
           .sort((a,b) => b.value - a.value)
           .slice(0,top_n);
      … }
```

```
//Count Sample Output
[{
   "constituency": 1,
   "countNo": 1,
   "name": "Blue Party",
    "value": 37,
    "lastValue": 0,
    "color": "#313190",
    "seats": 2,
    "quota": 54
  },

…

  {
    "constituency": 1,
    "countNo": 2,
    "name": "Blue Party",
    "value": 45,
    "lastValue": 37,
    "color": "#fe0300",
    "seats": 2,
    "quota": 54
  },
…
]
```

Figure 4.14: Code snippet - animating count data

This provides a basic implementation of an animated bar chart, giving count by count visualisation of a candidate's votes. However, it falls short of providing the user with an understanding of what is happening and why. As such a number of data visualisation enhancements were needed to increase the Zone of Proximal Development and encourage the learner to construct their own understanding of the vote-counting system.

In the original visualisation, when a candidate is eliminated the bar representing it stays static. This means the chart doesn't convey to the user whether that candidate was eliminated or simply received no transfers in that count, it also doesn't indicate to the user where a candidate is winning votes from. To address this a colour attribute was added to the output of the Counting method, and on the elimination of a candidate that candidates' colour attribute is set to grey. A colour selection is then made on each ticker iteration to update the bar colour as appropriate. This provides a visual cue to the user that a candidate has been eliminated, however, it still doesn’t fully explain to the user why that candidate has been eliminated.



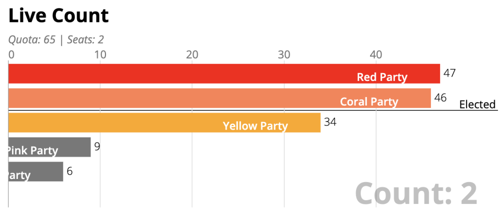


Figure 4.15: Live count - animated candidate elimination

For example in the figure above, two candidates are eliminated, not simply the lowest candidate. A novel way to address this is to provide “live” commentary on the count, with explanations of each step. This idea was touched on in the implementation of the Counting method, as addressing the issue was reliant on how well actions in that method could be translated to count commentary. As the Counting method was scrupulously constructed with the generation of rich data in mind, the addition of this feature was relatively easy. For each count, an explanation string was included in the JSON output, which was then appended to the count visualisation.

```
//Count Explanation Sample Output
[{ "countNo": 1,
    "constituency": 3,
    "comment": "Blue & pink excluded together as their current votes mean that they will
     be excluded in any event. Votes to be redistributed to remaining candidates."
  },{
    "countNo": 2,
    "constituency": 3,
    "comment": "Yellow is the lowest candidate and is deemed excluded. Red deemed
     elected without reaching quota. Coral deemed elected without reaching quota."
  }
```

Figure 4.16: Sample count commentary

```
//Isolate commentary for this count
let commentarySlice = currentCountCommentary.filter(d => d.countNo == countNo-1 &&
d.constituency == currentConstituency)

//Update commentary
  function updateCommentaryText(){
        commentaryText.html(`${commentarySlice}`);
    }
```

Figure 4.17: Code snippet - updating count commentary

Additionally, the quota, number of seats and a horizontal line demarcating the winning parties (with the number of parties above the line being the number of available seats) were added to the visualisation to provide further context.

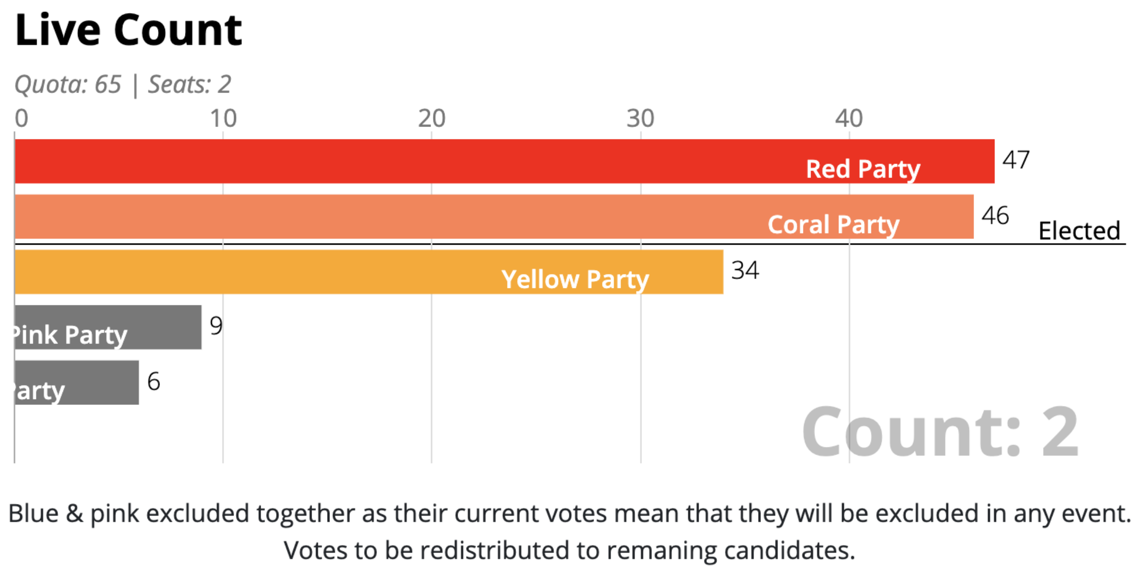


Figure 4.18: Final live count (animated chart)

4.7 Other visualisations

A series of other more standard-visualisations were implemented using D3 modules. These were selected on the basis of the analysis presented previously and how they satisfy the functional and non-functional requirements of the application. In each case, data from the Counting method is extracted, bound to elements and injected into DOM.

Stacked-column charts

Stacked-column charts are used in the application to display breakdown of votes received and seats won for each party. Bar charts and pie charts were also considered for this, however, stacked column charts were chosen as they more effectively enable direct comparison of the parts of multiple totals. This was implemented using a D3 single-stack module (Singh', 2018), and adjusted to maintain the correct party colour associations. The number of votes received and the number of seats received is presented in percentage format to further enhance this comparison, while it is also expressed in number format in order to allow the user to more directly understand how many constituencies a party may need to win or lose in order to, for example, increase the proportionality of a given election. The colour associated with each party is maintained in these charts to tie together the experience for the user.

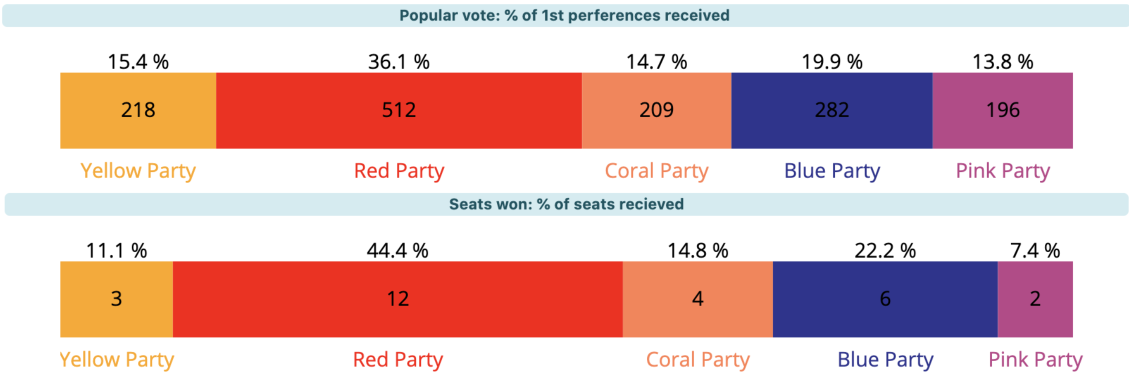


Figure 4.19: Final stacked-column charts

Hemicycle

It was important to further emphasize the result within the context of a parliament, so a hemicycle is also used to show the number of seats received by each political party as it mimics the layout of a parliament. This is implemented using the `d3-parliament` method, and adapted to incorporate the Gallagher index and maintain the correct colour association.

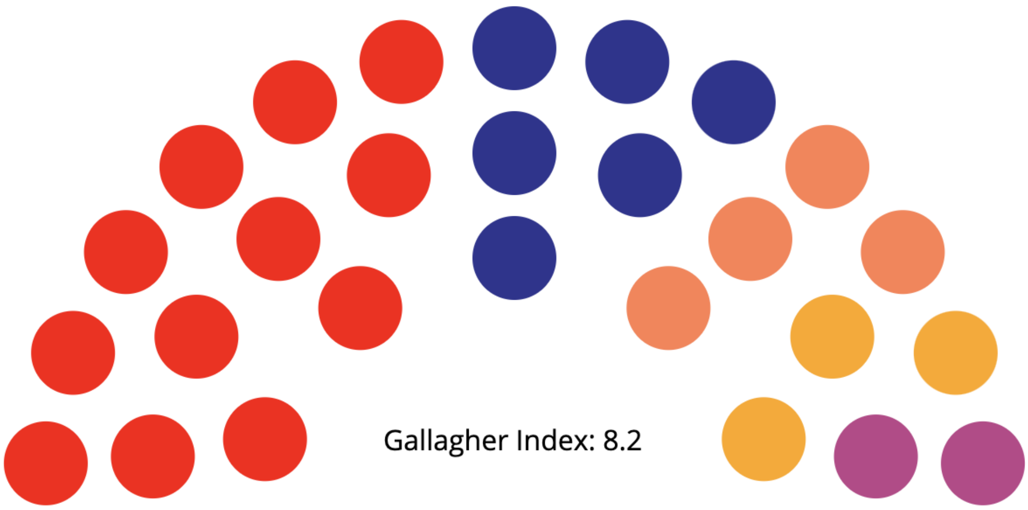


Figure 4.20: Final parliament seats visualisation

Sankey

Some parties are considered to be more transfer 'friendly' than others; this simply means that more voters are more likely to give them their second or subsequent preferences. Transfer patterns can often make the difference between winning or losing a seat. In the application, transfer patterns are modelled into voter data as described previously. As such it was important to capture this phenomenon and help users to visualise it. A Sankey diagram was chosen as it provides a quick and intuitive snapshot of PR-STV transfers by depicting the flow of votes from one party to another. This is implemented using the `d3-sankey` module and enhanced with tooltips to give an exact breakdown of voters following from party to party.

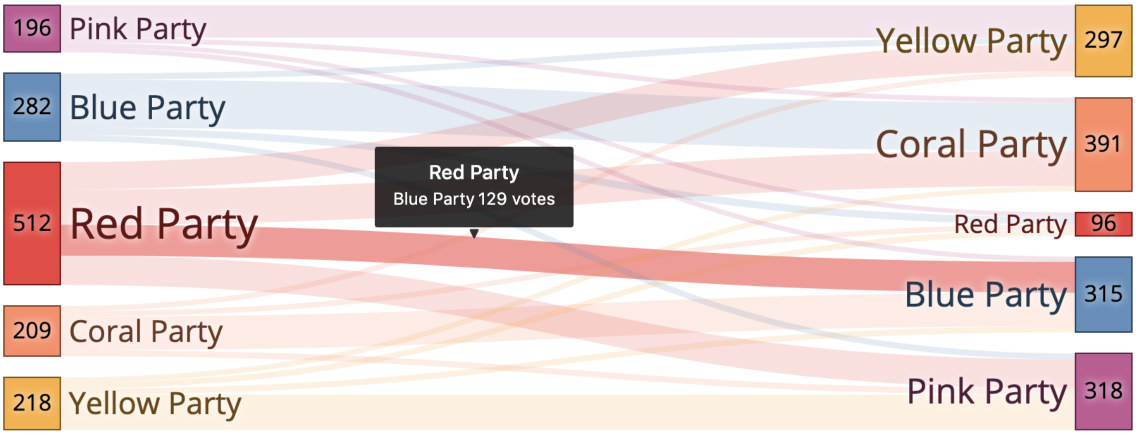


Figure 4.21: Final sankey diagram - vote transfers

4.8 Setting challenges

Active learning requires students to engage in meaningful learning activities which help them think about what they are doing. As such it was also important to implement a range of challenges that embody many of the desired learning outcomes and help the user to extract value from the application. Three challenges were set:

Challenge 1: Design a fair election
Design an election to establish the most proportional result possible by manipulating constituency boundaries and using different electoral systems. See what's the lowest Gallagher index you can achieve.

Challenge 2: Design an unfair election
Gerrymander an election to establish an unfair political advantage for a particular party or group by manipulating constituency boundaries. See what's the highest Gallagher index you can achieve.

Challenge 3: Design a seat majority
The Blue Party receives the second-largest amount of votes. Manipulate the election configuration so that it receives the largest seat majority.

In order to make each of these challenges effective, the initial electoral configuration was differentiated across the challenges. For example in Challenge 1, the default electoral configuration was set to have a relatively high Gallagher index. By exaggerating the result, in opposition to the goal, it initially makes the task easier and draws the user into the challenge. In this example, the difference between PR-STV and FPTP is also set to be relatively large in order to help the user get an initial sense of the difference the electoral system makes to the outcome of an election. Each configuration was set based on experimentation with constituency boundaries and voter distribution.

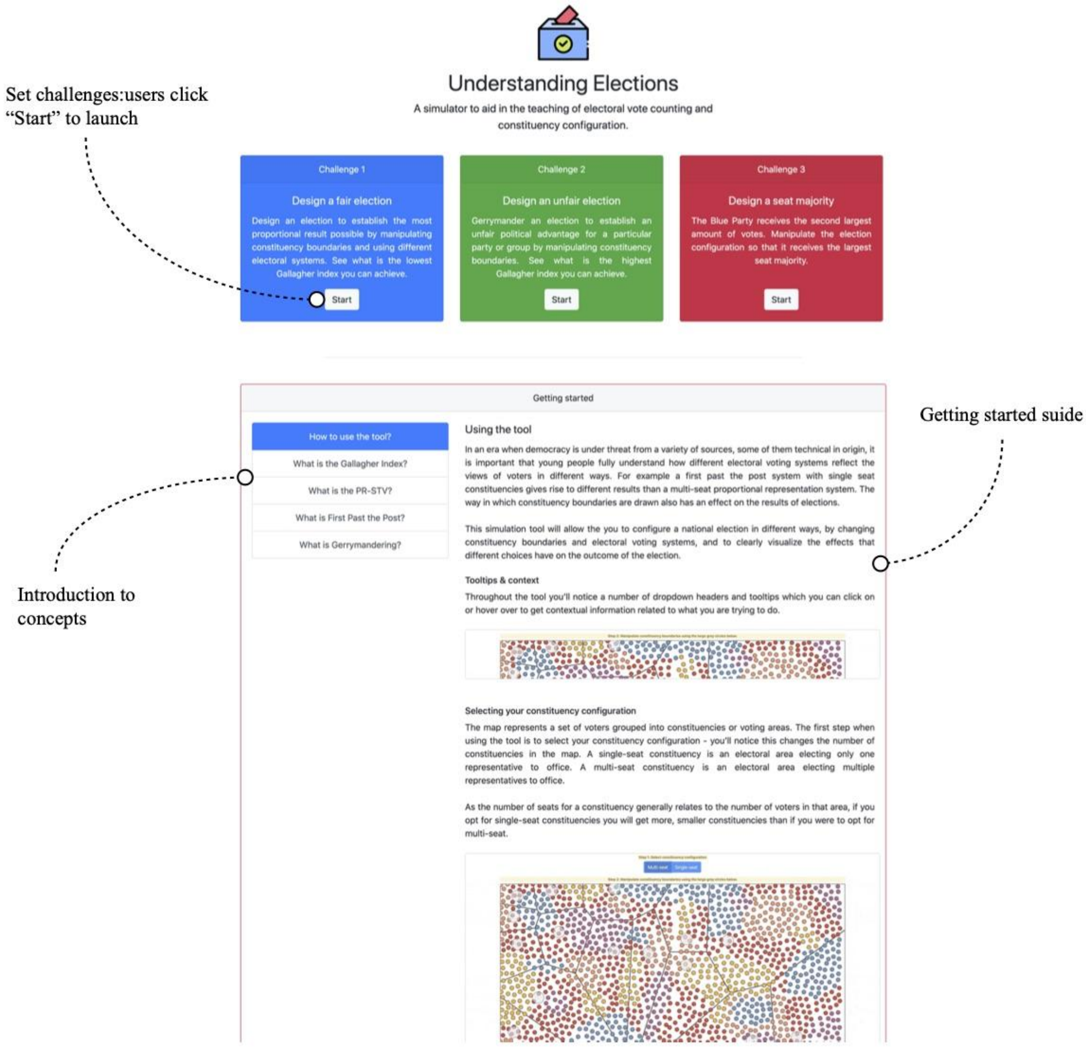
Figure 4.22: Final simulator challenges

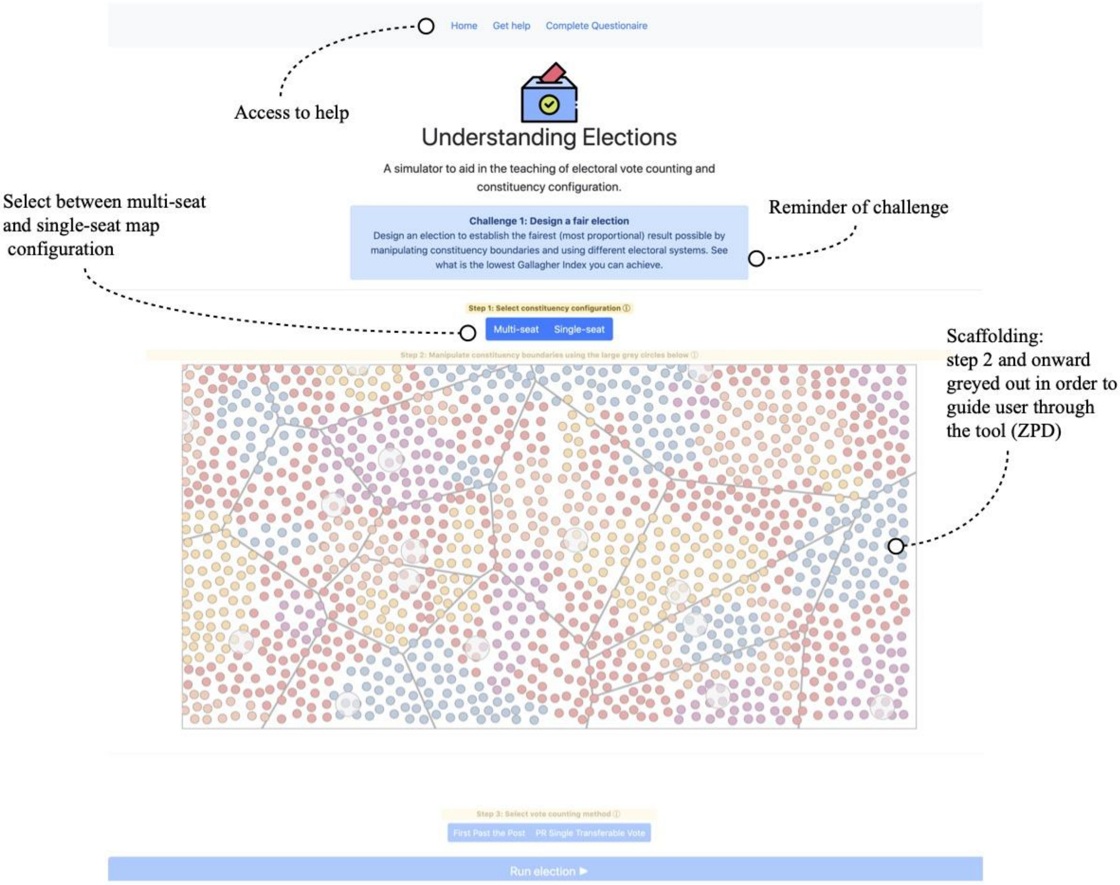
4.9 User interface flow and enhancements

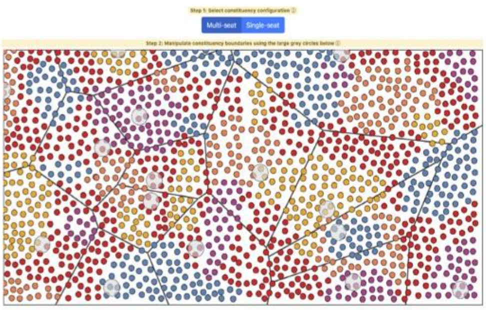
This section presents an overview of the final flow of the user interface and other enhancements that are best explained in the context of the flow.

The final flow draws heavily on human-computer interaction theory and centres on the concept of drilling down. The user interface focuses on providing only the most relevant information exactly at the time it is needed, alongside tools to allow the user to drill down on demand. In this way, the tool tries to become part of the thinking process and attempts not to get in the way of the learner. The application also builds in scaffolding techniques, such as process constraints (e.g. next step available on completion of the previous step) and prompts (e.g. tool-tips to encourage the user to carry out an action) to guide the user through the tool. This endeavours to increase the Zone of Proximal Development (ZPD) i.e. what the learner can do independently. In the same fashion, it also tries to make the interface accessible to a variety of users, both those new to the application and those more experienced users.

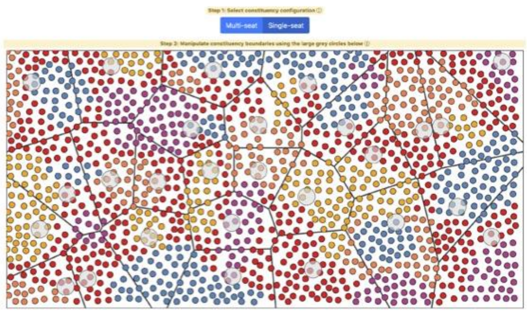
Throughout the interface, the user is provided with constant feedback; utilising tooltips, different mouse pointers, and highlighting (e.g. illustrating which constituency is elected) to provide a more intuitive experience. Feedback is also provided when a user attempts an action that is not permitted in the present scenario, such as manipulating a boundary in result mode. These enhancements were all utilised to improve the flow of the application and to allow the user to concentrate on discovering.


Set challenges:users click
“Start” to launch
Understanding Elections
A simulator to aid in the teaching of electoral vote counting and constituency configuration.
Challenge 1
Design a fair election
Design an election to establish the most proportional result possible by manipulating constituency boundaries and using different electoral systems. See what is the lowest Gallagher index you can achieve.
Start
Challenge 2
Design an unfair election
Gerrymander an election to establish an unfair political advantage for a particular party or group by manipulating constituency boundaries. See what is the highest Gallagher index you can achieve.
Start
Challenge 3
Design a seat majority
The Blue Party receives the second largest amount of votes. Manipulate the election configuration so that it receives the largest seat majority.
Start
Getting started
Getting started suide
How to use the tool?
What is the Gallagher Index?
What is the PR-STV?
What is First Past the Post?
What is Gerrymandering?
Introduction to
concepts
Using the tool
In an era when democracy is under threat from a variety of sources, some of them technical in origin, it is important that young people fully understand how different electoral voting systems reflect the views of voters in different ways. For example a first past the post system with single seat constituencies gives rise to different results than a multi-seat proportional representation system. The way in which constituency boundaries are drawn also has an effect on the results of elections.
This simulation tool will allow the you to configure a national election in different ways, by changing constituency boundaries and electoral voting systems, and to clearly visualize the effects that different choices have on the outcome of the election.
Tooltips & context
Throughout the tool you'll notice a number of dropdown headers and tooltips which you can click on or hover over to get contextual information related to what you are trying to do.
Selecting your constituency configuration
The map represents a set of voters grouped into constituencies or voting areas. The first step when using the tool is to select your constituency configuration - you'll notice this changes the number of constituencies in the map. A single-seat constituency is an electoral area electing only one representative to office. A multi-seat constituency is an electoral area electing multiple representatives to office.
As the number of seats for a constituency generally relates to the number of voters in that area, if you opt for single-seat constituencies you will get more, smaller constituencies than if you were to opt for multi-seat.

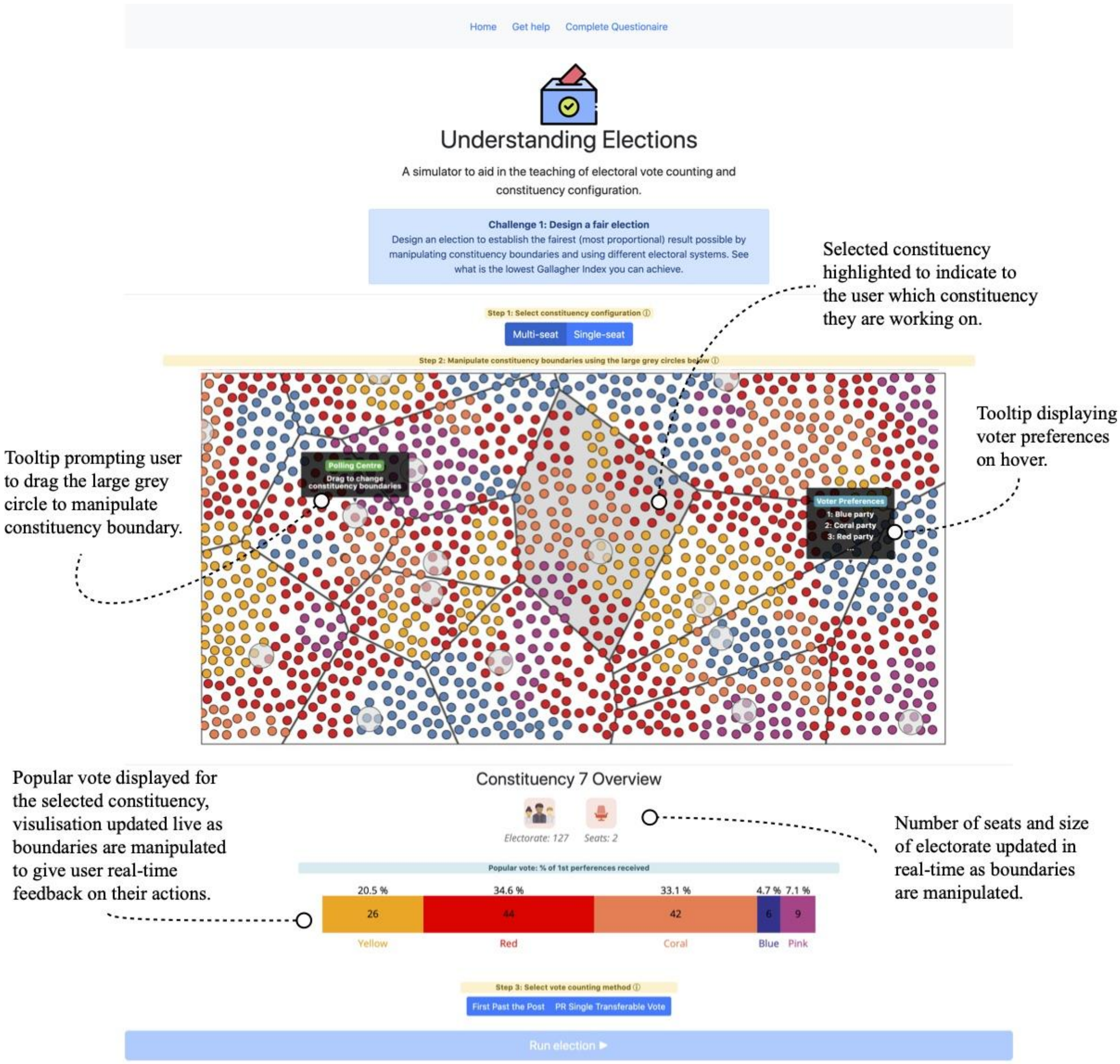




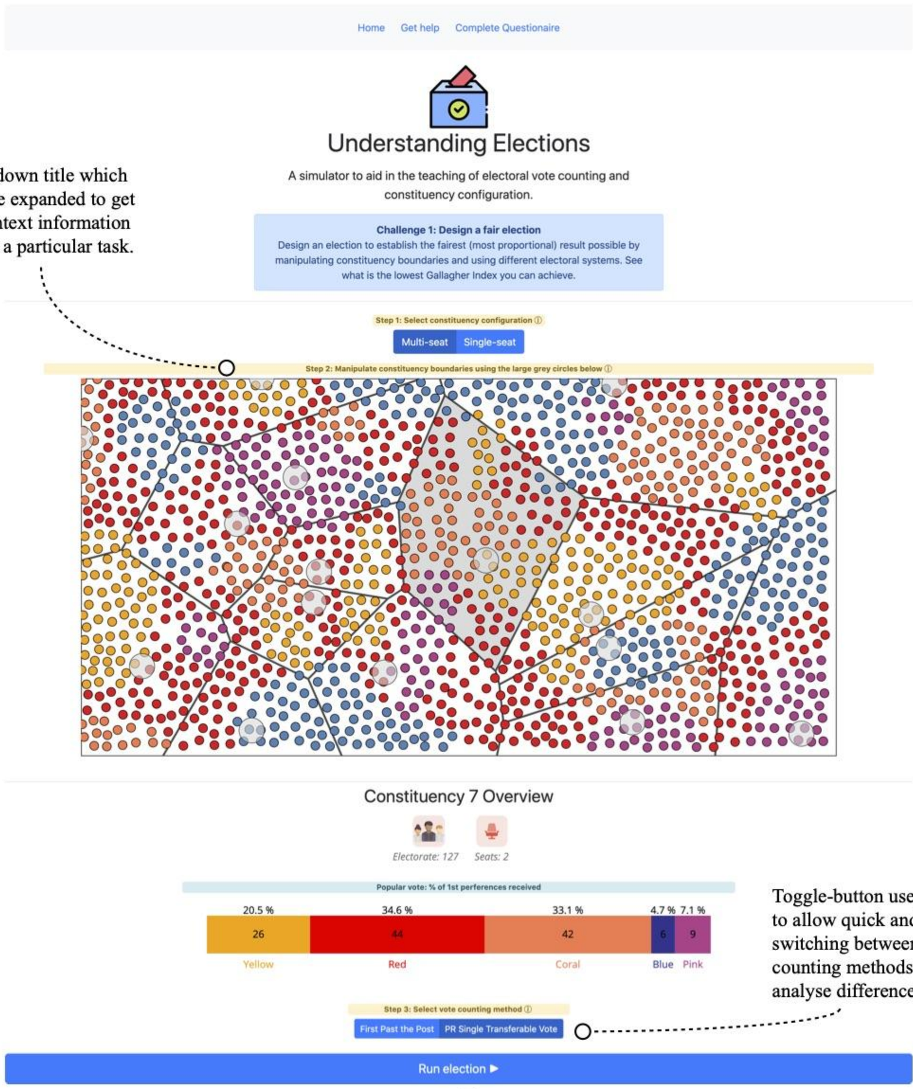
Multi-seat layout



Single-seat layout

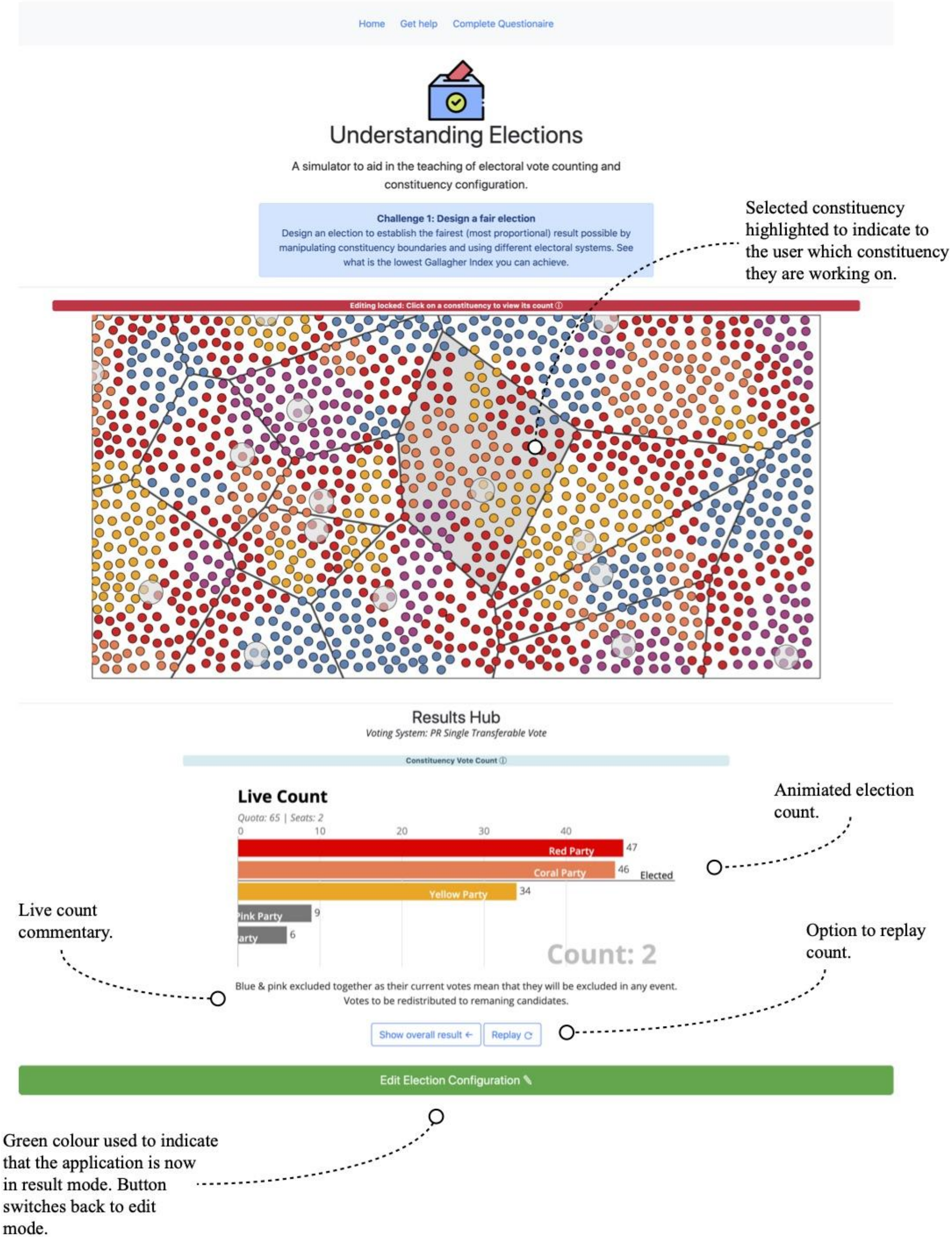

Home Get help Complete Questionaire
Understanding Elections
A simulator to aid in the teaching of electoral vote counting and constituency configuration.
Challenge 1: Design a fair election
Design an election to establish the fairest (most proportional) result possible by manipulating constituency boundaries and using different electoral systems. See what is the lowest Gallagher Index you can achieve.
Step 1: Select constituency configuration
Multi-seat
Single-seat
Step 2: Manipulate constituency boundaries using the large grey circles below
Polling Centre
Drag to change constituency boundaries
Voter Preferences
1: Blue party
2: Coral party
3: Red party
Constituency 7 Overview
Electorate: 127
Seats: 2
Popular vote: % of 1st perferences received
20.5 %
34.6 %
33.1 %
4.7 %
7.1 %
26
44
42
6
9
Yellow
Red
Coral
Blue
Pink
Step 3: Select vote counting method
First Past the Post
PR Single Transferable Vote
Run election ▶
Selected constituency highlighted to indicate to the user which constituency they are working on.
Tooltip prompting user to drag the large grey circle to manipulate constituency boundary.
Tooltip displaying voter preferences on hover.
Popular vote displayed for the selected constituency, visulisation updated live as boundaries are manipulated to give user real-time feedback on their actions.
Number of seats and size of electorate updated in real-time as boundaries are manipulated.

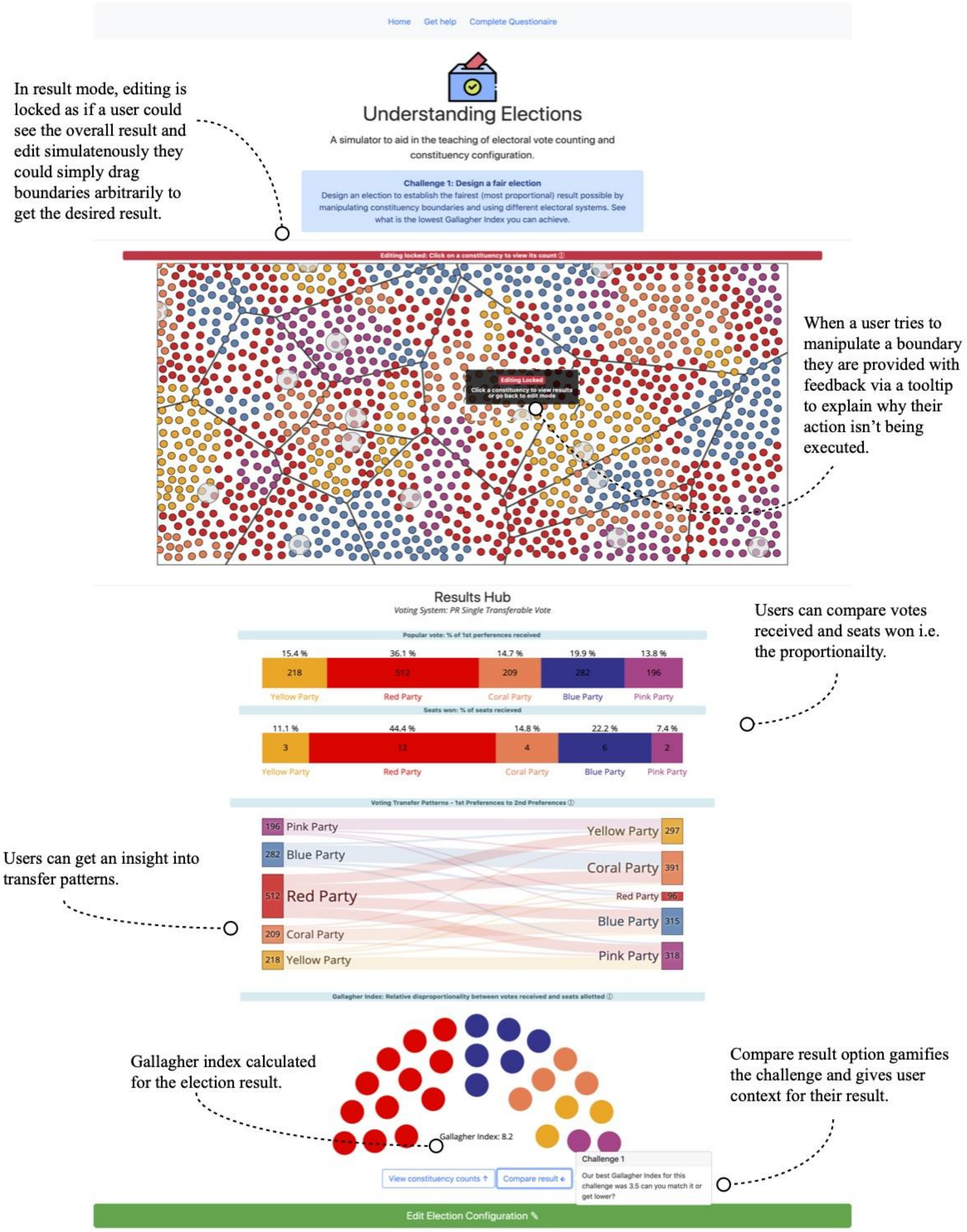
Dropdown title which can be expanded to get in context information about a particular task.



Toggle-button used to allow quick and easy switching between vote counting methods in order to analyse differences.

When an election is run, the application is switched to result mode. In result mode the colour is switched to Green as a visual cue to the user to indicate a mode change. As such the button uses a large form factor.


Home Get help Complete Questionaire
Understanding Elections
A simulator to aid in the teaching of electoral vote counting and constituency configuration.
Challenge 1: Design a fair election
Design an election to establish the fairest (most proportional) result possible by manipulating constituency boundaries and using different electoral systems. See what is the lowest Gallagher Index you can achieve.
Selected constituency highlighted to indicate to the user which constituency they are working on.
Editing locked: Click on a constituency to view its count
Results Hub
Voting System: PR Single Transferable Vote
Constituency Vote Count
Live Count
Quota: 65 | Seats: 2
Red Party 47
Coral Party 46 Elected
Yellow Party 34
Pink Party 9
6
Count: 2
Animiated election count.
Live count commentary.
Blue & pink excluded together as their current votes mean that they will be excluded in any event. Votes to be redistributed to remaning candidates.
Option to replay count.
Show overall result
Replay
Edit Election Configuration
Green colour used to indicate that the application is now in result mode. Button switches back to edit mode.



Home
Get help
Complete Questionaire
Understanding Elections
A simulator to aid in the teaching of electoral vote counting and constituency configuration.
Challenge 1: Design a fair election
Design an election to establish the fairest (most proportional) result possible by manipulating constituency boundaries and using different electoral systems. See what is the lowest Gallagher Index you can achieve.
In result mode, editing is locked as if a user could see the overall result and edit simulatenously they could simply drag boundaries arbitrarily to get the desired result.
Editing Locked
Click a constituency to view results or go back to edit mode
When a user tries to manipulate a boundary they are provided with feedback via a tooltip to explain why their action isn't being executed.
Results Hub
Voting System: PR Single Transferable Vote
Popular vote: % of 1st perferences received
15.4 %
36.1 %
14.7 %
19.9 %
13.8 %
218
512
209
282
196
Yellow Party
Red Party
Coral Party
Blue Party
Pink Party
Seats won: % of seats recieved
11.1 %
44.4 %
14.8 %
22.2 %
7.4 %
3
12
4
6
2
Users can compare votes received and seats won i.e. the proportionailty.
Voting Transfer Patterns - 1st Preferences to 2nd Preferences
196 Pink Party
282 Blue Party
512 Red Party
209 Coral Party
218 Yellow Party
Yellow Party 297
Coral Party 391
Red Party 96
Blue Party 315
Pink Party 318
Users can get an insight into transfer patterns.
Gallagher Index: Relative disproportionality between votes received and seats allotted
Gallagher index calculated for the election result.
Gallagher Index: 8.2
Challenge 1
Our best Gallagher Index for this challenge was 3.5 can you match it or get lower?
View constituency counts
Compare result
Compare result option gamifies the challenge and gives user context for their result.
Edit Election Configuration

5. Testing and evaluation

5.1 Planned workshop and evaluation session

For testing and evaluation, a series of workshop sessions were devised in order to test the application with Transition Year students. The workshop planned a series of activities anchored around the simulation tool, wherein learners are given room to explore various electoral systems and the tools to construct their own election. On completion of the workshops, participants would be asked to complete a post-session questionnaire to collect information about their experiences with the application. A sample of participants would also be asked to take part in semi-structured focus group interviews lasting between 20 - 30 minutes for further qualitative analysis of the questionnaire results.

These activities were due to take place as part of the Bridge21 Transition Year programme - an innovative 21st-century education programme based at this author’s university - in March led by a member of the Bridge21 team in collaboration with this author. Ethics approval was granted, however, due to the COVID-19 pandemic, and the subsequent closure of this author’s university and schools across Ireland, the workshops had to be cancelled.

In light of this scenario, it was necessary to pivot quite late to online questionnaires. Additionally, due to ethical considerations, it was not possible to do such online questionnaires with Transition Year students as they are typically under the age of 18. The original questionnaire was adapted to capture feedback from peers - students and staff - at this author’s university. While an additional questionnaire was also devised for educators - secondary school teachers, lecturers, education researchers etc. - to capture views on the effectiveness of the tool from a pedagogical perspective. This presented a range of challenges in recruiting participants as the application was designed to be used within an education setting and completing the questionnaire requires an individual to invest time both in using the tool and completing the questionnaire.

5.2 Usability testing

Usability testing remains a central way of ascertaining whether users are achieving their goals (Sauro et al., 2012) as such this formed an important pillar of the testing of this application. In the field of Human-Computer Interaction (HCI), there is a range of methods for evaluating and comparing interfaces with respect to their usability and other desirable properties such as the non-functional requirements set for this application. A popular method is usability questionnaires, for which there are a number of standardized questionnaires specifically designed to assess participants’ perceived usability and satisfaction of applications. Two of the most popular are the System Usability Scale (SUS) and the Post-Study System Usability Questionnaire (PSSUQ).

System Usability Scale (SUS)

The System Usability Scale is a 10 item questionnaire, it typically tests the usability and learnability of a system. SUS uses a 5 point Linkert scale and generates scores ranging from 0-100 (Brooke, 1996). SUS is useful as it provides a standardized method to gather feedback quickly and easily it is often referred to as the 'Quick and Dirty' usability scale. The popularity of SUS also makes it easy to contextualise results within industry benchmarks or results for similar systems.

	The System Usability Scale Standard Version		Strongly disagree 1	2	3	4	Strongly agree 5
1	I think that I would like to use this system.		O	O	O	O	O
2	I found the system unnecessarily complex.		O	O	O	O	O
3	I thought the system was easy to use.		O	O	O	O	O
4	I think that I would need the support of a technical person to be able to use this system.		O	O	O	O	O
5	I found the various functions in the system were well integrated.		O	O	O	O	O
6	I thought there was too much inconsistency in this system.		O	O	O	O	O
7	I would imagine that most people would learn to use this system very quickly.		O	O	O	O	O
8	I found the system very cumbersome to use.		O	O	O	O	O
9	I felt very confident using the system.		O	O	O	O	O
10	I needed to learn a lot of things before I could get going with this system.		O	O	O	O	O

Figure 5.1: System Usability Scale (SUS)

Post-Study Usability Questionnaire (PSSUQ)

The Post-Study Usability Questionnaire is a 16 item questionnaire, it typically measures system usefulness, information quality, interface quality, and has sub-scores for each. PSSUQ uses a 7 point Linkert scale and generates an overall satisfaction score based on the averages of the three sub-scores (Sauro and Lewis, 2016). Similarly to SUS, PSSUQ is useful as it provides a standardized method to gather feedback which can be compared with industry benchmarks.

	The Post-Study Usability Questionnaire Version 3	Strongly agree 1	2	3	4	5	6	Strongly disagree 7	NA
1	Overall, I am satisfied with how easy it is to use this system.	O	O	O	O	O	O	O	O
2	It was simple to use this system.	O	O	O	O	O	O	O	O
3	I was able to complete the tasks and scenarios quickly using this system.	O	O	O	O	O	O	O	O
4	I felt comfortable using this system.	O	O	O	O	O	O	O	O
5	It was easy to learn to use this system.	O	O	O	O	O	O	O	O
6	I believe I could become productive quickly using this system.	O	O	O	O	O	O	O	O
7	The system gave error messages that clearly told me how to fix problems.	O	O	O	O	O	O	O	O
8	Whenever I made a mistake using the system, I could recover easily and quickly.	O	O	O	O	O	O	O	O
9	The information (such as online help, on-screen messages and other documentation) provided with this system was clear.	O	O	O	O	O	O	O	O
10	It was easy to find the information I needed.	O	O	O	O	O	O	O	O
11	The information was effective in helping me complete the tasks and scenarios.	O	O	O	O	O	O	O	O
12	The organization of information on the system screens was clear.	O	O	O	O	O	O	O	O
13	The interface* of this system was pleasant.	O	O	O	O	O	O	O	O
14	I liked using the interface of this system.	O	O	O	O	O	O	O	O
15	This system has all the functions and capabilities I expect it to have.	O	O	O	O	O	O	O	O
16	Overall, I am satisfied with this system.	O	O	O	O	O	O	O	O

Figure 5.2: Post-Study Usability Questionnaire (PSSUQ)

Choosing a questionnaire

Based on analysis of the two questionnaires, PSSUQ was chosen for inclusion in this study. While PPSUQ is somewhat longer and more complex than SUS, with 16 questions compared to 10, it's questions are more targeted and are designed specifically for a scenario-based type of usability test. As users are asked to complete a series of challenges this makes PSSUQ more effective. Additionally, as the application includes both interactive elements and background information on topics, the breakdown of scores for system usefulness, information quality, interface quality is useful to better understand which parts of the application the user is referring to.

Results

The PSSUQ was included in both the peer questionnaire and the educator questionnaire, with a total of 55 participants completing this element of the questionnaire. As this is the first time the tool was evaluated, the results can be interpreted within a Sauro and Lewis study which calculates PSSUQ Norms from 21 studies and 210 participants (Sauro and Lewis, 2016). The resulting scores take values between 1 and 7, with lower scores being favourable.

The feedback was positive across all four scoring components, suggesting a high degree of perceived satisfaction with the application. Information Quality was the most critical result, although still far exceeding PSSUQ Norms. Dissecting this result in the context of the qualitative feedback received, it would seem that this is chiefly due to the absence of a mediated learning environment for which the application was designed for. Some participants suggested the inclusion of a video introduction or the ability to hover over words to get a definition would be useful in addressing this. On the other hand, Interface Quality received the most favourable result, consistent with the qualitative feedback received. A large number of participants complimented the high level of interactivity, use of visualisations and the use of colour throughout the tool. In terms of system usefulness, users appeared to feel comfortable, and satisfied using the system and were able to complete the set challenges without frustration. Overall the results suggest that the application satisfies its goals in terms of usefulness and accessibility, and provides a strong foundation for future iterations to build on with continued testing and evaluation.

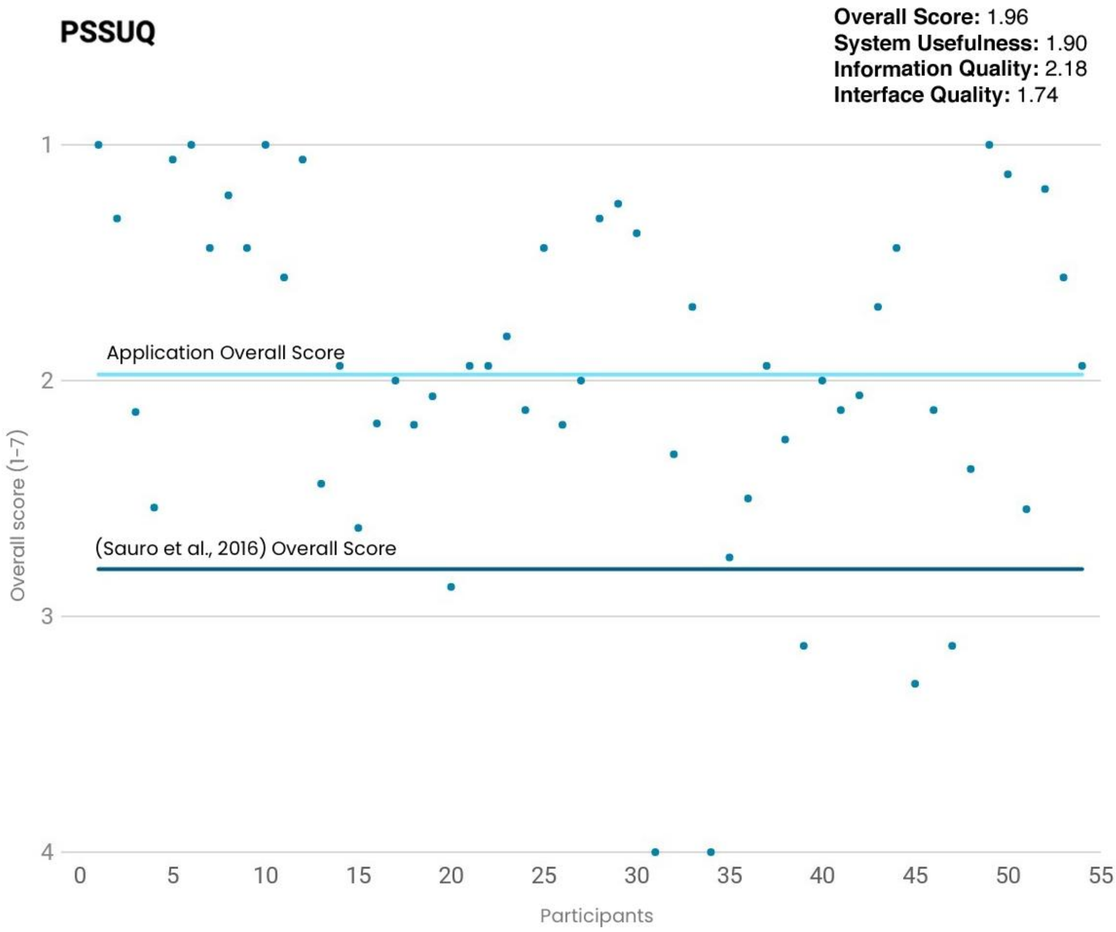


Figure 5.3: PSSUQ result (scatter-plot)

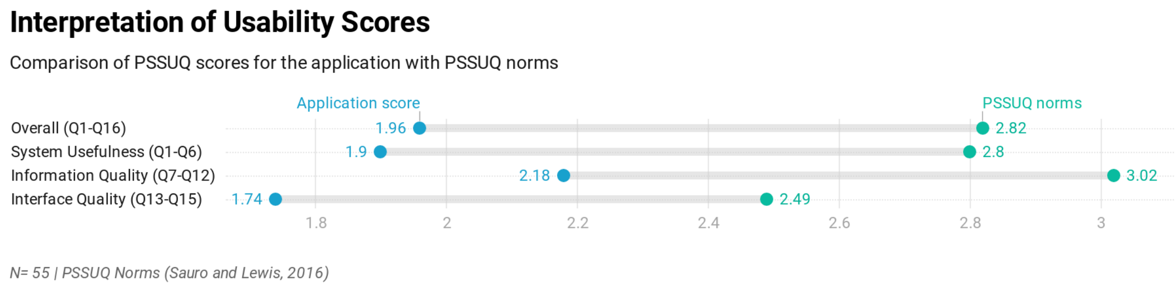


Figure 5.4: Application PSSUQ results compared with norms (range-plot)

5.3 Peer feedback

In the absence of the ability to evaluate the tool with real learners, feedback was sought from this author's peers including both students and staff at the University. While this wasn't the ideal scenario, it nonetheless provided interesting results regarding the effectiveness of the application in embedding learning outcomes.

An interesting way to examine the effectiveness of the application was to look at opinion changes pre and post use. Participants were asked to what extent they agreed with several statements regarding electoral systems prior to having used the application, and to what extent they now agree with the same statements.

The table below illustrates changes in the mode of opinion for each statement. This is interesting as it illustrates the constructionist notion of learners coming with pre-existing beliefs and adapting them based on experimentation. For example, when asked about the fairness of different electoral systems, there was a clear divergence between PR-STV and FPTP, with many viewing PR-STV as fairer post-use and FPTP as less fair post-use.

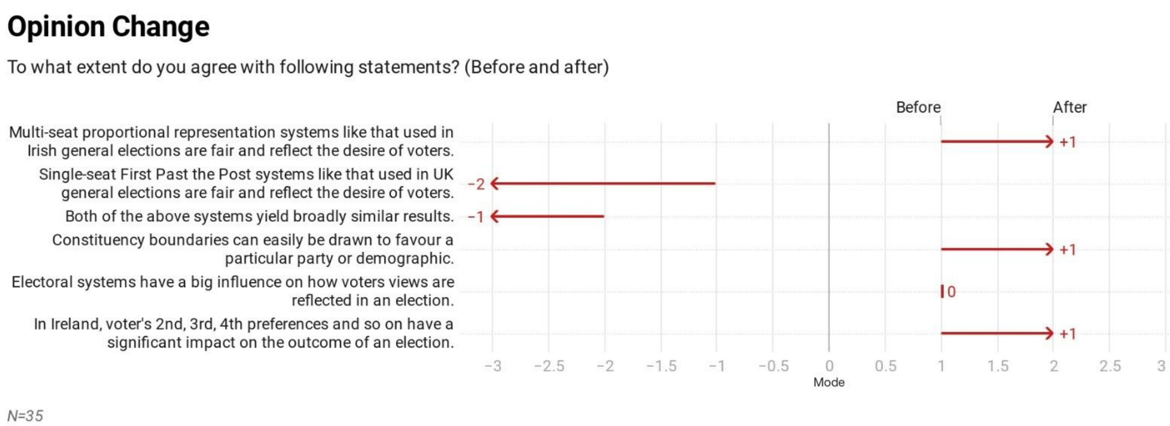


Figure 5.5: Change in mode of opinion post-use (arrow-plot)

The results clearly indicate that the application is successful in embedding a range of concepts including proportionality and gerrymandering, for which there were shifts in opinion. This is further emphasised by the fact 97% of participants stated that they were either more informed or much more informed after using the application, and again this is borne out in qualitative feedback across the board:

"I was unaware that current elected parties could influence the district boundaries (Gerrymandering) I assumed these district changes came purely from towns land population changes"

"I learned a lot about the different voting systems, different ways the results can be counted, and how these can have big impacts on the winners of the election."

"I learned that using my 2nd, 3rd & 4th votes can deeply impact the outcome of an election."

"Very interesting to see relatively abstract concepts represented in such a concrete manner."

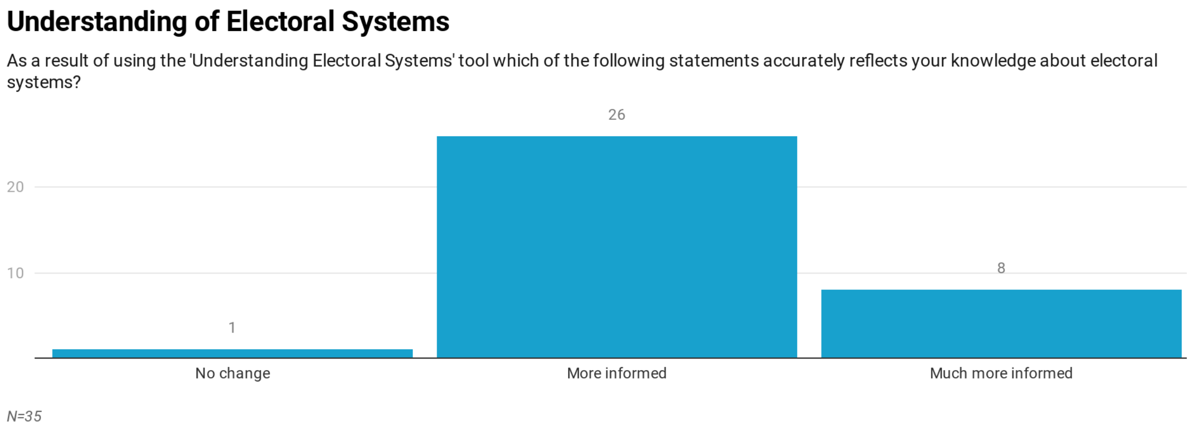


Figure 5.6: Change in understanding of electoral systems (column-chart)

5.4 Educator feedback

In the absence of the ability to evaluate the tool with real learners, it was also necessary to capture the views of educators on the effectiveness of the tool from a pedagogical perspective. Feedback was sought from secondary school teachers, education researchers, and lecturers with a total of 22 respondents.

Participants were asked to what extent they agreed with several statements regarding the pedagogical effectiveness of the tool. The figure below gives a snapshot of sentiments expressed by educators which were largely positive across the board.

The majority of respondents strongly agreed that the application would be effective in a cooperative learning environment, and this is firmly borne out within the qualitative feedback. Interestingly, despite this, a small number of respondents felt the tool didn't provide enough background information, which suggests the application would work best in an appropriate learning environment as intended. This provides further explanation for the information quality score received in the usability component of the questionnaire.

The responses also clearly indicate that educators perceive the tool to embody the core components of a microworld, with many citing an effective use of interactive visualisations which allow learners to test their ideas. Again this is consistent with the interface quality score received in the usability component of the questionnaire, indicating there is a strong correlation between both components of the questionnaire.

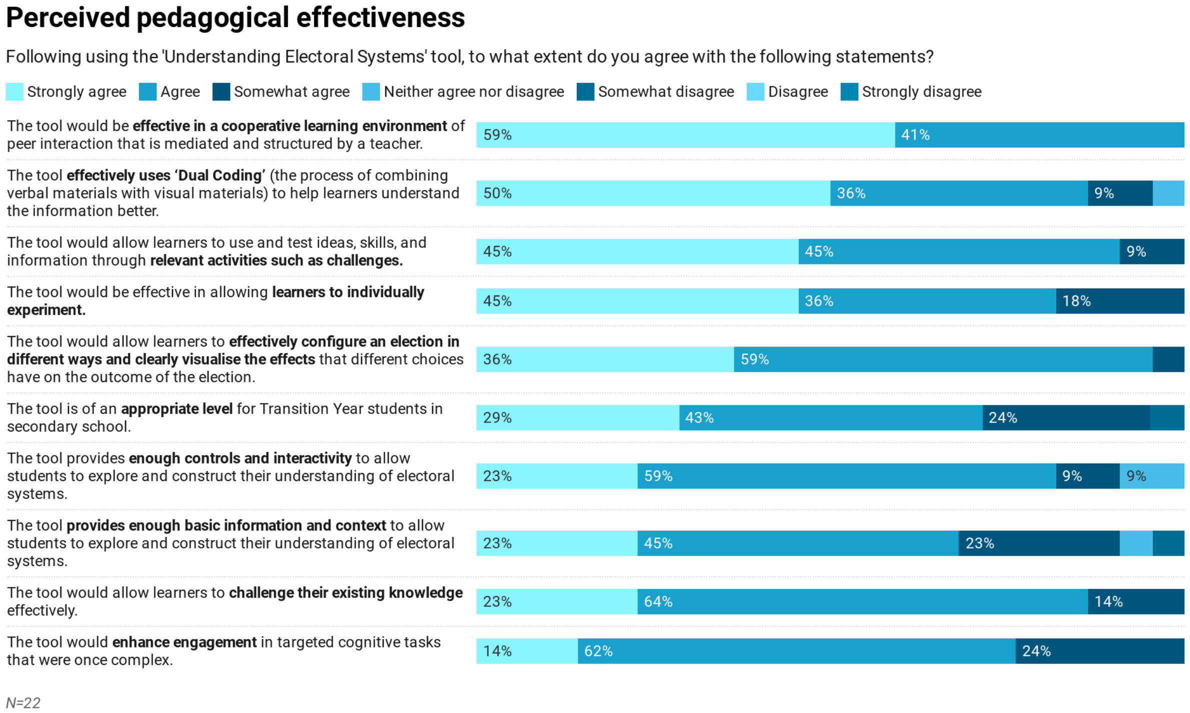


Figure 5.7: Perceived pedagogical effectiveness (stacked-bar chart)

Overall the sentiment expressed by educators was overwhelmingly positive, with all respondents saying they would be likely, very likely or extremely likely to use or recommend the use of the application. There was also a strong indication that learners would be more likely to engage with this tool over existing methods.

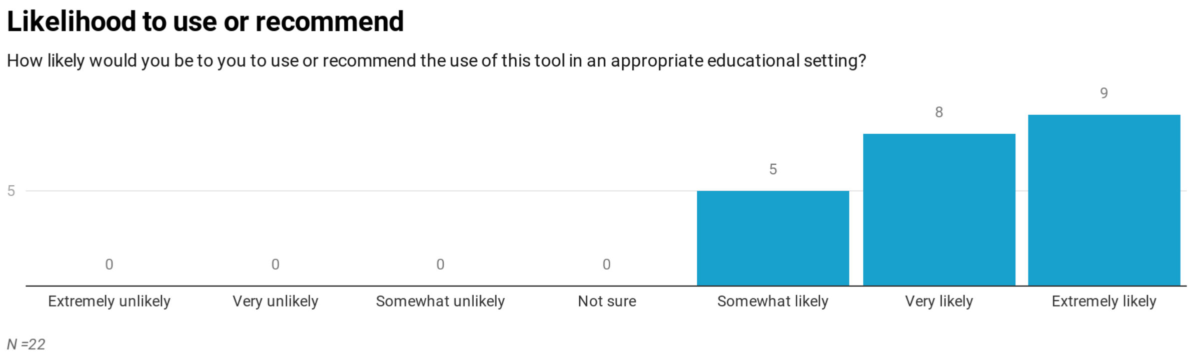


Figure 5.8: Likelihood to use or recommend (column-chart)

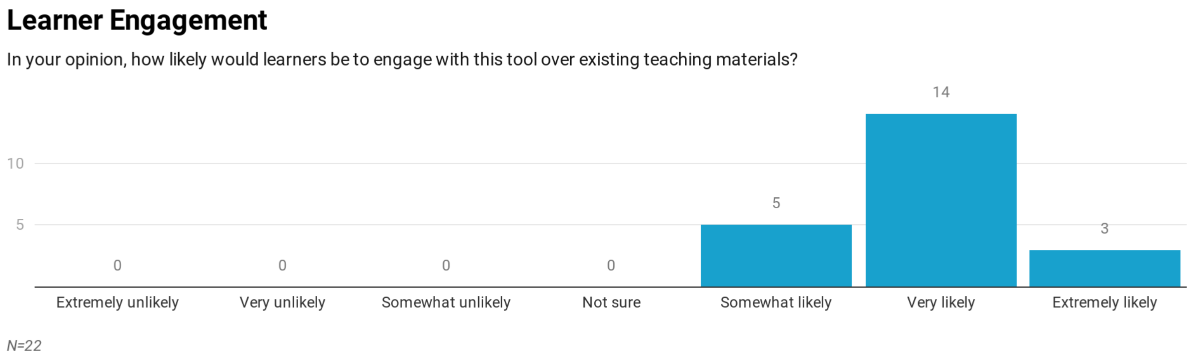


Figure 5.9: Learner engagement (column-chart)

This is further borne about qualitative feedback received. Interestingly a number of respondents suggested the application would also be a useful tool across a variety of secondary school subjects including Computer Science, History and Math due to the nature of the data and simulations used throughout the tool.

"Not only great for TY CSPE classes but great for LC Computer Science students (ALT3 - modelling & simulation example) and History students (look at gerrymandering in Derry in early 1960s). Very visual. Love the transfers."
– Secondary School Teacher

"The tool is useful from an educators perspective, however, it has to be embedded correctly in a range of warm-up activities. The tool and website is well built, user friendly and accessible. There is a huge amount of data that can be used to explain and tie into other areas of maths." – Secondary School Teacher

"The visual and interactive capabilities are very inviting for students."
– Educational Researcher

6. Conclusions

6.1 Summary

This study aimed to utilise modern data visualisation techniques and technologies in order to design a simulator underpinned by pedagogical and human-computer interaction theory, which would aid in the teaching of electoral systems. Based on a quantitative and qualitative analysis of the application, it is clear that the simulator successfully utilises many of these aspects and has shown an ability to positively impact individuals' understanding of electoral systems.

Despite the initial steep learning curve, the D3 JavaScript visualisation library proved to be an advantageous choice. D3's innate flexibility encouraged a level of creative freedom that was essential to this project. The data generation aspect of the application focused on embedding concepts of real-world electoral systems (e.g. building in voting patterns), while the data visualisation aspect of the tool focused on helping users explore and construct an understanding of these concepts. This turned out to be a successful model with the data-driven approach of D3, affording a high degree of extensibility to the application - while the simulator was designed to compare and contrast two electoral systems, it could easily be extended to other systems by binding existing visualisations to new datasets.

A range of data visualisation techniques were drawn on for the design of the application, and were adapted to improve user experience and learning outcomes. In particular, the use of Voronoi diagrams for constituency manipulation is a novel but efficient way to allow users to quickly and easily alter boundaries without tedious tile-by-tile manipulation. The use of animation to illustrate the counting process proved a popular storytelling device amongst respondents to the questionnaire. One of the major challenges in the design of the application was to not overwhelm the user with different visualisations. This was overcome by presenting relevant information as it is needed, e.g. a vote breakdown for only the constituency being manipulated at that time. Additionally, the popular design pattern in human-computer interaction, of presenting high-level overviews with the ability to further analyse or "drill down" as necessary, was utilised and appeared to be successful in making the tool more accessible.

Based on the educator feedback, the application also appears to successfully encompass many of the desired constructionist and pedagogical attributes required. The incorporation of the Gallagher index was a core implementation, as it allowed for relevant challenge-setting, result comparison, and contextualisation of those results, all of which are essential for the learning outcomes of the application. In particular, the ability to produce a metric which can be compared by peers is fundamental in constructionist learning.

While the application was perceived by educators to incorporate the hallmarks of an effective microworld, and has demonstrated an ability to positively affect learning outcomes, there is still a need to evaluate the application with real learners in a collaborative learning environment, to accurately examine its effectiveness.

6.2 Future work

This section offers possible extensions to the presented work.

- **Testing and evaluation with real learners:** Following on from the conclusions made, testing the effectiveness of the application as originally intended would undoubtedly be a priority in any future work. In doing so, it would be possible to obtain much richer qualitative and quantitative insights to test the hypothesis of success assumed from the peer and educator evaluation.

- **Modelling of real election data and scenarios:** Datasets representing electoral results from across the world are widely available and could be used to model real election scenarios within the simulator. This has the potential to enhance engagement with the simulator and enable learners to critically access real elections. However, there are possible ethical concerns surrounding the potential misuse of such modelling to manipulate real elections. Additionally, some datasets don't provide complete information, for example in Ireland due to the nature of the counting process, if two candidates or more are eliminated at once it is not possible to know where exactly candidates subsequently receive votes from.

- **Modelling of additional electoral systems:** As alluded to previously, while the application was designed to model just two electoral systems, PR-STV and FPTP, the application could easily be extended to include other electoral systems due to the data-driven nature of the visualisations. PR party-list would be an interesting additional system to model given its popularity around the world.

Appendix

A live version of the application is available at https://electoral-simulator.firebaseapp.com/

Bibliography

ACKERMANN, E. 2001. Piaget's Constructivism, Papert's Constructionism: What's the difference?

BENYON, D., TURNER, P. & TURNER, S. 2005. *Designing interactive systems: People, activities, contexts, technologies*.

BLUMENFELD, P., SOLOWAY, E., MARX, R. & KRAJCIK, J. 2011. Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. *Educational Psychologist,* 26**,** 369-398.

BONWELL, C. C. & EISON, J. A. Active learning : creating excitement in the classroom. 1991.

BOSTOCK, M., OGIEVETSKY, V. & HEER, J. 2011. D3 Data-Driven Documents. *IEEE Transactions on visualisation and Computer Graphics,* 17**,** 2301–2309 [Online]. Available: http://vis.stanford.edu/files/2011-D3-InfoVis.pdf [Accessed].

BRUNELL, T. L. 2009. The Realities of Redistricting: Following the Rules and Limiting Gerrymandering in State Legislative Redistricting. *Perspectives on Politics,* 7.

CARD, S., MACKINLAY, J. & SHNEIDERMAN, B. 1999. *Readings in Information visualisation: Using Vision To Think*.

CSO. 2011. *Quarterly National Household Survey: Voter Participation* [Online]. Central Statistics Office. Available: https://www.cso.ie/en/media/csoie/releasespublications/documents/labourmarket/2011/voterq22011.pdf [Accessed].

DASSONNEVILLE, R., QUINTELIER, E., HOOGHE, M. & CLAES, E. 2012. The Relation Between Civic Education and Political Attitudes and Behavior: A Two-Year Panel Study Among Belgian Late Adolescents. *Applied Developmental Science*.

DE JONG, T., SOTIRIOU, S. & GILLET, D. 2014. Innovations in STEM education: the Go-Lab federation of online labs. *Smart Learning Environments*.

DEPARTMENT OF HOUSING, GOVERNMENT OF IRELAND. 2020. *A Guide to Ireland's PR-STV Voting System* [Online]. Available: https://www.housing.gov.ie/sites/default/files/publications/files/pr-stv_guide.pdf [Accessed].

DISESSA, A. A. Changing Minds: Computers, Learning, and Literacy. 2000.

FARRELL, D. 2001. Electoral Systems: A Comparative Introduction.

GALLAGHER, M. 1992. Comparing Proportional Representation Electoral Systems: Quotas, Thresholds, Paradoxes and Majorities. *British Journal of Political Science,* 22**,** 469-495.

GALLAGHER, M. 2019. *Election indices dataset* [Online]. Available: https://www.tcd.ie/Political_Science/people/michael_gallagher/ElSystems/Docts/ElectionIndices.pdf [Accessed].

GALLAGHER, M. & MITCHELL, P. 2006. The Politics of Electoral Systems. *The Politics of Electoral Systems***,** 1-688.

GOLLOB, R., KRAPF, P., ÓLAFSDÓTTIR, Ó. & WEIDINGER, W. 2010a. *Educating for democracy: Background materials on democratic citizenship and human rights education for teachers*, Council of Europe.

GOLLOB, R., KRAPF, P., ÓLAFSDÓTTIR, O. L. F., WEIDINGER, W. & EUROPARAT 2010b. *Background materials on democratic citizenship and human rights education for teachers*.

HAKE, R. R. 1998. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics,* 66**,** 64-74.

KELLY, D. 2007. *Who Votes? An Exploration of the Factors Influencing Turnout Propensity in Fingal County.* Masters, National University of Ireland Maynooth.

MA, Y. 2017. Civic Returns to Education: Voter Turnout in Ireland. *The Economic and Social Review,* 48**,** 145-169.

MAYER, R. 2005. Cognitive theory of multimedia learning.

MCCARTHY, L., REAS, C. & FRY, B. 2016. *Make: getting started with p5.js : making interactive graphics in JavaScript and Processing,* San Francisco, CA, Maker Media, Inc.

MCGANN, A. J., SMITH, C. A., LATNER, M. & KEENA, A. 2016. *Gerrymandering in America: The House of Representatives, the Supreme Court, and the Future of Popular Sovereignty,* Cambridge, Cambridge University Press.

MINDELL, D. 2000. LEGO Mindstorms: The Structure of an Engineering (R)evolution [Online]. Available: http://web.mit.edu/6.933/www/Fall2000/LegoMindstorms.pdf [Accessed].

MURRAY, S. 2017. Interactive data visualisation for the web : an introduction to designing with D3.

NCCA 2016. Civic, Social & Political Education A Citizenship Course - Specification for Junior Cycle [Online]. Available: https://www.curriculumonline.ie/getmedia/4370bb23-00a0-4a72-8463-d935065de268/NCCA-JC-Short-Course-CSPE.pdf [Accessed].

NISHINO, S., SONG, J., TANI, K. & YANAI, Y. 2016. *Teaching How Electoral Systems Change Political Outcomes Using a Role-Playing Simulation Game*.

PAPERT, S. 1980. *Mindstorms: children, computers, and powerful ideas*, Basic Books, Inc.

PAPERT, S. 1993. *The children's machine: rethinking school in the age of the computer*, Basic Books, Inc.

PEKKANEN, R., SHUGART, M. & HERRON, E. 2017. *Oxford Handbook of Electoral Systems*.

PIAGET, J. & INHELDER, B. R. 1969. *The psychology of the child*.

PRINCE, M. 2004. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education,* 93**,** 223-231.

SADOSKI, M. & PAIVIO, A. 2000. *Imagery and Text: A Dual Coding Theory of Reading and Writing*.

SHUGART, M. S. & TAAGEPERA, R. 2017. *Votes from Seats: Logical Models of Electoral Systems,* Cambridge, Cambridge University Press.

SMITH, B. & MACGREGOR, J. 1992. What is Collaborative Learning?

TREISMAN, A. & GORMICAN, S. 1988. Feature analysis in early vision: evidence from search asymmetries. *Psychol Rev,* 95**,** 15-48.

UK, E. R. S. 2020. *Party List Proportional Representation* [Online]. Available: https://www.electoral-reform.org.uk/voting-systems/types-of-voting-system/party-list-pr/ [Accessed].

UNWIN, A. 2020. Why is Data visualisation Important? What is Important in Data visualisation?

VYGOTSKIĬ, L. S. & COLE, M. 1978. - *Mind in society : the development of higher psychological processes*.

WARE, C. 2008. *Visual Thinking: for Design*, Morgan Kaufmann Publishers Inc.