Data Representation with No-Code Augmented Reality Authoring Tools

Ana Beatriz Marques, Vasco Branco, Rui Costa, and Nina Costa

Abstract This study is framed within information design and focuses on data representation using augmented reality (AR) and its appropriation by designers. The need to investigate no-code AR authoring tools was motivated by the conclusions drawn during the development of "Floating Companies" AR prototype, which highlight the limited access of designers without programming skills to this technology. We intend to identify the limitations of no-code tools regarding development platforms, such as Unity, and represent data with no-code tools. The methodology used entails three phases—the collection and characterization of no-code AR tools; the review of its limitations regarding the development milestones in FLOC; and the proposal of data representation based on no-code tools. The AR tools landscape offers several free platforms which do not require programming skills. It was found that the analyzed tools do not support algorithmic data representation, which forces any representation to be designed manually in a customized way, presenting limitations regarding the amount of data, but also opportunities. The type of project that no-code platforms support falls within the concept of communicative visualization—a type of visualization that does not intend to deeply analyze the data, but rather to communicate and engage public.

Keywords Augmented reality · No-code · Data representation

1 Introduction

This study concerns data representation using AR and its appropriation by designers without programming background. The need to investigate AR authoring tools was prompted by conclusions drawn during the development of the AR prototype "Floating Companies" (FLOC), created for the exhibition of the Design OBS Project (FBAUL, 2021) [1]. FLOC is the prototype of an AR application for mobile devices

A. B. Marques (✉) · V. Branco · R. Costa · N. Costa
Department of Communication and Art, Research Institute for Design, Media and Culture [ID+], University of Aveiro, University Campus of Santiago, 3810-193 Aveiro, Portugal
e-mail: abmarques@ua.pt

N. Martins et al. (eds.), *Perspectives on Design and Digital Communication IV*, Springer Series in Design and Innovation 33, https://doi.org/10.1007/978-3-031-41770-2_19

that visualizes in hybrid space the design companies registered in Portugal in 2019, in the form of a set of spheres that float in space, justifying the name of the application. The project was framed by *Design OBS—Towards a Design Observatory: models, instruments, representations and strategies* [2], which collects and interprets data on the Portuguese design ecosystem to promote its knowledge and influence public policy.

FLOC was implemented using the Unity, one of the most widely used platforms for creating AR experiences, often involving the creation of scripts (code instructions). The path followed in the prototype development, presented below in a summarized way, represented a great effort, being the biggest barrier the need to learn and use programming concepts in C# without a specific background in that area [1]. Other authors have identified the problem of difficult access to the creation of AR experiences by non-technologists, namely [3–6]. Given this limitation, we identified the need to collect and investigate no-code AR authoring tools currently available, which can be used in a more constraint-free way, by designers who typically do not have a programming background and want to focus their attention on the application design.

Providing an access that presents fewer barriers for non-technologists to enter the creation of AR content is essential to develop the language in this medium. This study is relevant from two points of view—by presenting designers with tools aiming to simplify their introduction into AR content creation; and relevant in the sense that it contributes to the gradual exploration of AR by creators who will eventually find ways to adapt content to the medium, giving it its own identity. As MacIntyre et al. [7] refer, the history of new media has shown that any medium will only reach its full potential when it is made available to designers who, through their action, define the popular forms of the medium. Gutenberg invented the printed medium but not the novel, Edison invented the moving image but not film. Berners-Lee invented HTTP and HTML, but not the web. Sutherland invented AR technology, and the forms that AR content will take are still unknown. Design therefore involves imagining the new rather than finding a solution within a set, being concerned not with the way things are (deductive reasoning) but with the way they could be (abductive reasoning) [8]. While discussing the implications of recent technological advance for innovation and design theory, Verganti et al. [8] frame creativity as a process of problem-finding rather than problem-solving, which is synonymous of ‘making sense of things’. The human role becomes that of understanding which problems/opportunities should be addressed and driving the continuous evolution of algorithms towards a meaningful direction. The core of this activity is therefore not problem solving, but problem discovery (pp.224–225). This notion of design for technological innovation applies in the context of this study, as the implications of making AR more accessible to designers do not focus on solving problems, but rather on formulating new problems that can drive data representation with AR in a meaningful direction.

The objectives of this study include mapping and characterizing the set of no-code tools currently available online; ascertaining the limitations and possible strengths in the design of data representations with the collected tools, compared to the work done during the development of the “Floating Companies” prototype in Unity (Sect. 2).

2 Floating Companies (FLOC) AR Prototype

FLOC involved the design, development, and user testing within the scope of the exhibition "Design Observatory in Portugal - Situation" (FBAUL, 2021). The project followed a practice-based methodology which aimed to investigate, from the design point of view, the strengths, and limitations that hybrid space presents for data visualization, particularly when there is no semantic relationship between that data and the space where it will be inserted.

FLOC represents information about the set of design companies in Portugal from a database curated by Design OBS [9]. In the application, each company in the sample corresponds to a sphere, whose color and diameter reflects its size (number of employees). Their positioning in space gives an approximate idea of the distribution of the sample across the Portuguese territory, but also of their relative performance (the vertical position of each company is an indicator of its profit per employee) (Fig. 1). On the ground plan there is a district map of Portugal under which the companies are located. By tapping on each sphere further information on that company can be accessed—name, district, and profit per employee. The visualization offers a second scene—the resume view—in which the number of companies per employee class in each district is represented based on a 3D sphere chart.

FLOC explored the inherent potential of AR to render abstract numerical values concrete by simulating them in the real environment. In the main view the aim was to give a concrete existence to the number, distribution, and attributes of design companies through the physical configuration of the virtual objects representing them. FLOC was developed using Unity game engine combined with Vuforia SDK, which allowed to use an area target that launches the experience through the recognition

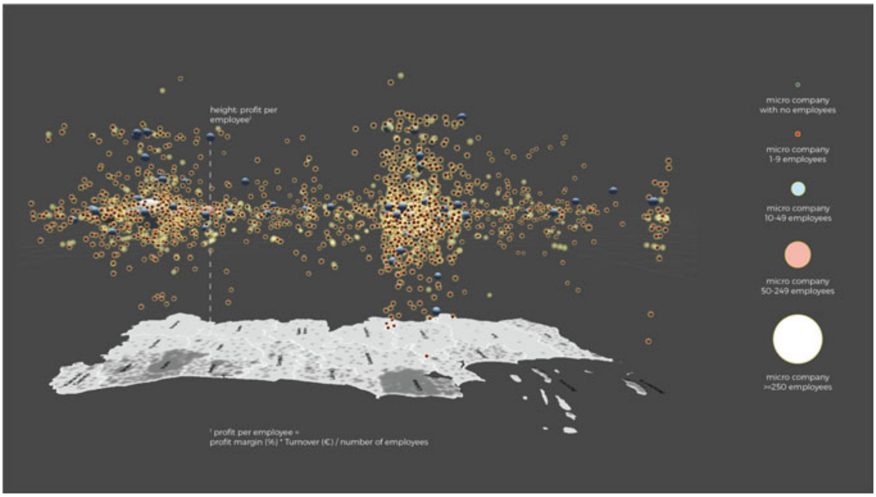

height: profit per employee¹
micro company with no employees
micro company 1-9 employees
micro company 10-49 employees
micro company 50-249 employees
micro company >=250 employees
¹ profit per employee = profit margin (%) * Turnover (€) / number of employees

Fig. 1 Floating Companies App prototype. Arrangement of the virtual elements in space

of the exhibition room. Overall, the prototype development involved the following milestones:

1. Create and assign the prototype an area target to launch the AR experience with the recognition of the exhibition space.
2. Import the database on companies into Unity in CSV format and make it analyzable by Unity using a script.
3. Implement a script to analyze the database, generating a second database with the information on size, color, and location to be assigned to each sphere (prefab), based on the information on size, location, and profit of each company.
4. Implement a script to read the latest database and instantiate a sphere for each of the companies according to the location, size, and color information, within a bounding box.
5. Create a second scene - Resume View - where the total number of companies per district is represented. Here, instead of assigning a sphere to each company, the data are integrated, and each sphere represents the number of companies in a certain district.
6. Add User Interface elements (buttons to (1) the project description; (2) link to the original database; (3) the resume view and back).

   i. Attribute behavior to the spheres when they are tapped on - light effect and display of an information board (details-on-demand) with information extracted from the generated database.

Our team were composed mainly by designers without programming knowledge and the development process was initiated based on a tutorial on how to build a scatterplot graph from a database in Unity [10], being gradually adapted to the goals of the project over multiple development iterations. During this process we recurred to Unity forums, to its large community of users and often used excerpts from other existing programs. At a later stage, it was also necessary to call upon the expertise of a programming professional with knowledge in C#. This process was time consuming, laborious and at times restricted design options due to the limited programming knowledge.

3 Literature Review

Although visual representation of data has a long historical tradition, the vocabulary of the field is constantly evolving. For some authors data visualization is an area perfectly differentiated from infographics [11], for other authors infographics is a sub-area of data visualization [12, 13]. Iliinsky and Steele [11] distinguish infographics from data visualization in a synthetic way—infographics refers to a visual representation of data that is (1) manually-drawn (with a personalized treatment of the information); (2) specific to the data in question (and therefore difficult to recreate with different data); (3) aesthetically rich; (4) limited in terms of data (as each piece of information is manually coded). Data visualization, on the other hand, refers to a representation of data that is (1) algorithmically designed; (2) easy to regenerate with different data; (3) often aesthetically poor and (4) data-rich (as integration of large amounts of data is viable). On the other hand, Masud et al. [12] consider infographics as communicative visualization, ie. a type of visualization that is not used to convey a detailed data analysis, but rather to communicate a narrative. This type of visualization is used to raise awareness of a particular topic even if it is not intended to provide data analysis. Visual metaphors and illustration are visual strategies often used in communicative visualization.

Numerous examples of this type of visualization combined with AR come from journalism [14, 15], such as “Inflation Shrinking Ray” [16]; “EV Battery Breakdown” [17] or “Water footprint ” [18]. In general, when creating AR applications, most content developers use the so-called game engines: software applications that allow teams composed of professionals from different contexts to collaborate in the creation of an application [19]. Many of the most well-known game engines, such as Unity or Unreal, require programming to create interaction, which hampers the exploration of these tools by non-programming designers [4]. Unlike game engines, high-fidelity prototyping tools, which will be covered in this article, are digital tools used by designers and software developers to address interface details without a full implementation [20].

In literature, authoring tools are classified according to a variety of criteria. Regarding the way content is previewed, they may be considered as ‘digital tools’, ‘immersive tools’ or both. Typically, digital authoring tools (e.g. Lens Studio) (Fig. 2) support a preview of the project using an emulator (a software that imitates another computer system) that needs to be deployed on the device for testing, while immersive authoring tools (e.g. Apple Reality Composer) (Fig. 3) allow editing while previewing the AR experience in the user environment [21]. Lee et al. [22] address the advantages of immersive authoring tools, which allow users to preview, experience and check first-hand the virtual content and its integration into the real environment.

Billinghurst et al. [23] collect and classify AR tools according to the required programming skills—from ‘low level software libraries’ requiring good development skills, to simple authoring tools for novice users with no programming knowledge. While libraries and low-level software provide a high level of flexibility while requiring programming skills, standalone authoring tools allow end users to easily create AR content with a minimum of programming knowledge, although the content created is quite simple. Due to the rapid evolution of this research field, the collection made in 2015 by Billinghurst et al. is no longer up to date with some of the software mentioned discontinued.

Dengel et al. [5] conduct a systematic literature review on AR Authoring Toolkits for educators without programming skills. Authors identify 69 different toolkits which they classify according to their accessibility, level of programming skills required, and supported interactivity. The methodology used is well documented, the time interval is wide—25 years—and the collection is recent and updated. However, due to the collection method based on bibliography only, the absence of some no-code

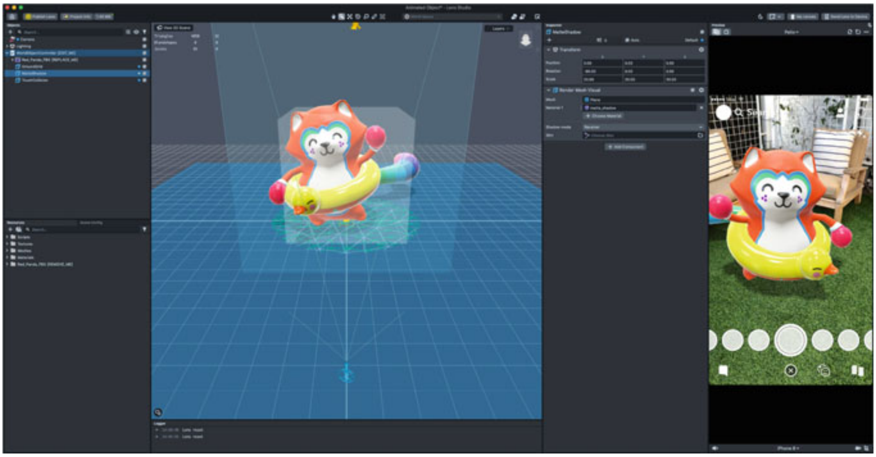


Fig. 2 Lens Studio AR emulator

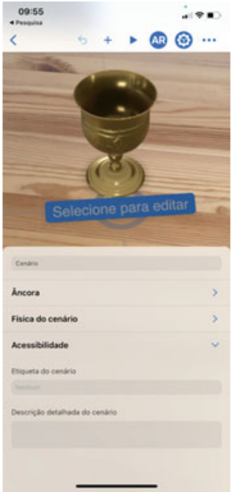


Fig. 3 Reality Composer immersive environment

software available online, such as Reality Composer or Adobe Aero, was immediately noted. Furthermore, many of the collected tools are suitable for educators but are not sufficiently flexible for designers as they provide closed uneditable solutions. In systematic reviews on AR tools, other authors [24] considered the review supported only by academic literature potentially restrictive as it may exclude tools that despite not being academically validated are used by a community of practitioners.

Some authors develop and propose their own authoring tools [4, 6, 7, 25]. One of the best-known references is DART (Designer's Augmented Reality Toolkit) [7]—a tool launched in 2003 aimed at new media designers which has been widely used by a diverse population of creators during the following years. Gandy and MacIntyre collected the reflections of a group of DART users regarding the software's use and developed guidelines for building AR authoring tools for non-technologists [26]. The guidelines fall into the following categories: (1) access, (2) layered authorship, (3) collaboration with diverse teams, (4) making the process less painful and (5) the importance of community. Access concerns the technological barriers that non-technologists encounter when using AR authoring tools. Layered authoring is about providing a layered development environment that allows for an appropriate level of complexity, for instance a no-code tool that also provides access to scripting. Collaboration with diverse teams considers the need to integrate several specialized collaborators in the same project. Point 4 relates to the need to make the creation process easier. Finally, the importance of an active online community that conveys a sense of permanence is highlighted. Other authors present guidelines for the creation of an AR tool for non-technologists, such as Leiva et al. [4] who point out four design goals for a rapid prototyping tool: 1. create low fidelity assets through sketching; 2. position assets in space through direct manipulation; 3. capture user position changes through video; 4. create complex animations through direct manipulation. This is also the case of Schmalstieg & Höllerer [3] who argue that an AR authoring solution should address the following points: allow different possibilities of relating the application content to the real world (tracking modes); foster cross-platform compatibility; support a collaborative workflow and enable future reuse.

3.1 Description Categories

Categories for describing the authoring tools were outlined based on the literature review, combining points enunciated by different authors, but also based on the empirical knowledge accumulated throughout the development of previous AR experiences [1, 27]. **Accessibility**—The relevance of accessibility in 'new media' tools is referred to by Gandy and MacIntyre [26] in the technological sense, but also in the sense of presenting fewer barriers to entry in general. Technological accessibility relates to the need to provide a layered authoring environment that allows the user to access an appropriate level of complexity. Here, the criterion of accessibility has been extended to affordability, which can also be a barrier for designers looking to start exploring AR. **Tracking modes** and device compatibility—The support of different ways of relating the application content to the real world (tracking modes) and the need to foster cross-platform compatibility are two criteria pointed out by Schmalstieg and Höllerer [3] for a successful AR authoring solution. **Interactivity**—According to Billinghurst et al. [23] no-code authoring tools allow to easily create AR content with minimal programming knowledge but offer little flexibility. Therefore, it is important to ascertain whether the tools collected support some level of interaction and if this support depends on programming. **Type of authoring tool**—One of the important features in AR tools pointed out by Leiva et al. [4] is the possibility of positioning assets in space through direct manipulation, which is achieved through immersive or mixed authoring tools. **Publication**—The place of publication relates to accessibility, not accessibility for content creators, but user accessibility.

4 Research Question

Based on the literature review, we identified the absence of a comprehensive systematic review of AR no-code tools currently available online. This collection exists, but is limited to the academic literature, excluding many solutions that are currently available online for free. We also identified the inexistence of literature on the strengths and limitations of using these types of no-code tools for data representation.

The following research question is posed: What are the currently available no-code AR tools that are most suitable for designers without programming knowledge in data representation? As a secondary question, we intend to find out the advantages and limitations of using these tools in comparison with game engines, such as Unity. This study aims to fulfil three objectives: (1) To map and characterize the set of no-code tools currently available online. (2) To outline the limitations and strengths of working with the collected tools, using for comparison the work done previously during the development of "Floating Companies" in Unity. (3) To illustrate the type of visualization that can be designed with no-code tools, using data from Design OBS, similarly to the strategy previously followed [1, 27].

5 Methodology

This study can be divided in three parts—the first part (Methodology and Results) aims at collecting and characterizing the AR no-code platforms; the second part (Results) consists in verifying the limitations of those platforms regarding the tasks performed during FLOC development; the third part (Results) presents a data representation project built from a no-code tool, using Design OBS data regarding Portuguese design companies. The systematic collection of no-code tools and the identification of their limitations regarding development platforms are intended not only to map this universe, but also to qualify the use of these tools, identifying their advantages, but also the type of assignments for which they are not prepared.

The third part (Results) presents a data representation project named "Pencils" that showcases how data can be conformed through these tools. It also explores some challenges and complementarities between no-code AR apps, with full implementation tools, bringing AR closer to designers with no advanced coding competences.

The survey of no-code tools was carried out using the systematic review collection method. The collection process, as well as the characterization, are documented in the database "No-Code AR Platforms" [28]. To construct a comprehensive survey, this study uses a systematic literature review through web search as described by

Stansfield et al. [29], who suggest a three-stage process when constructing a systematic review through website search: (1) planning the search, (2) executing the search and (3) screening records for relevance and managing the results. 1. Planning the research involves justifying and informing decisions about: where to research, who is doing the research, and the timeframe and resources available for the review. 2. In the research, the main objective is to use each resource in a consistent and individually appropriate way. 3. finally, the results obtained must be filtered according to their relevance and according to previously established eligibility criteria.

5.1 Planning the Search

The search for no-code tools via web search, using Google's advanced search, was motivated by the need to gather a comprehensive collection that is often not documented in the academic literature. Given the topicality of the subject, no start was defined for the research time interval, which only has end date (all results until 26/10/2022 and all results until 27/10/2022). The collection entailed three advanced Google searches (Table 1). The aim of performing three searches from synonymous terms—'no-code augmented reality', 'augmented reality without coding' and 'augmented reality without programming'—was to encompass the maximum number of results and not to rely all the search on a single term, which could not be the most representative. One of the main challenges of research planning is to know which sites are most appropriate for the research [29]. In this context, it is intended to conduct a survey as complete as possible, so only ads, Facebook pages and LinkedIn profiles were excluded.

5.2 Executing the Search

Applied the exclusion criteria, advanced searches originated 30 web pages ('source' column of the 'Executing the search' sheet [28]), from which the AR platform names were manually extracted. In most web pages, platform names were extracted through a navigation along the URL. In the case of YouTube links, the platforms were extracted from the video content. In the case of publications on Reddit website or Twitter, only the search result post and its comments were considered. Of the total websites analyzed, two pages did not mention AR platforms, the remaining 28 pages pointed 38 names of supposed no-code AR platforms: Bundlar, Brand XR Studio, Geenee AR, Zapworks, Adobe Aero, AR Media, ARway, Aryel, Augmania, Minsar, PlugXR, Scapic, Spark Studio, XR + , Amazon Sumerian, Appypie, AR Code, Augm it! Blipparbuilder, Byldr, CanvasLogic, Envisage AR, Epigraph, Imersian, JigSpace, Lens Studio, MetaVRse, MyWebAR, Reality Composer, Scopear, ThreeKit, UniteAR, Vuforia Studio, Wintor AR, WorldCAST, XR Today, Hololink, Quartz Composer.

Table 1 Collection of no-code AR platforms based on three advanced searches using Google

Search nr	1	2	3
Search terms	'no-code augmented reality'	'augmented reality without coding'	'augmented reality without programming'
Time frame	All results until 26/10/2022	All results until 27/10/2022	All results until 27/10/2022
Google advanced search criteria	All these words: no-code augmented reality; this exact word or phrase: "no-code augmented reality"; language: english; region: any region; last update: anytime; terms appearing: in the title of the page; file type: any format	All these words: augmented reality without coding; this exact word or phrase: "augmented reality without coding"; language: english; region: any region; last update: anytime; terms appearing: in the title of the page; file type: any format	All these words: augmented reality without programming; this exact word or phrase: "augmented reality without programming"; language: english; region: any region; last update: anytime; terms appearing: in the title of the page; file type: any format
Results counting	23 results	6 results	4 results
Results counting (after excluding Facebook and LinkedIn pages)	20 results	6 results	4 results

5.3 Screening Records

For this study, only software that meets the following criteria were considered:

Type of tool: The platform must be an authoring tool. Therefore, it is necessary to verify that names extracted from each webpage correspond to actual AR tools - high fidelity and low fidelity tools are included. Excluded are platforms whose primary purpose is e-commerce and marketing. **Availability**: The platform must be available online, excluding platforms that have been discontinued or whose website cannot be accessed at the time of the search. **Accessibility**: The platform should be free or provide a free license (with no need for meeting with the team). Platforms with commercial licenses providing a free version, include the following typologies: Free limited version; Free use until x views; Free version for non-commercial use; Free version for personal and commercial use; Free version for testing and personal use. Excluded are platforms offering a trial version only for a limited period or offering a trial by appointment. **Required programming skills**: The platform should always provide forms of interaction without code. This includes tools which, in addition to the no-code feature, support coding for more complex behaviors. Tools relying entirely on programming skills are excluded.

The described criteria originated 12 platforms (Table A in annexes): Geenee AR; Adobe Aero; PlugXR Creator; Spark Studio; XR +; Blipparbuilder; Byldr; JigSpace; Lens Studio; Reality Composer; WorldCAST and Hololink.

6 Results

6.1 Comparison Between No-Code Tools

Affordability—Four platforms are completely free—Adobe Aero, Spark Studio, Lens Studio and Reality Composer. Remaining platforms are commercial but provide a free version that may: limit non-commercial use, limit features compared to the paid version, or require the use of a watermark. **Static/Interactive**—The following criteria analyze the capabilities of each tool without programming (for example, if a tool offers interactivity but only by scripting, it is considered static). Geenee AR, Jigspace and WorldCast platforms are considered static as they support a single predefined form of interaction. The remaining platforms offer varying forms of interaction—both in flexibility and in interaction implementation. Seven platforms support GUI-based interaction editing—PlugXR Creator; XR +; Blipparbuilder; Byldr; Lens Studio; Reality Composer and Hololink. Adobe Aero and Spark Studio base their interaction on a system known as 'drag and drop'. Geenee AR is the only platform requiring code to implement interaction. Interaction complexity among tools is highly variable—although Lens Studio and Blipparbuilder implement GUI-based interaction, Lens Studio presents a much wider set of interaction options, being highly flexible. Although Lens Studio and Spark Studio platforms allow creating AR experiences without writing code, they require some technical knowledge and familiarity with the development environment. These tools stand out as being those that present greater complexity but also greater flexibility. **Layered Authoring**—Six tools offer a layered authoring environment, simultaneously supporting coding: Geenee AR; Spark Studio; XR +; Blipparbuilder; Lens Studio and Reality Composer. All the analyzed cases use Javascript, except for Reality Composer, which is based on Swift. **Type of Authoring Tool**—Most tools are digital (9); Byldr is immersive; Adobe Aero and Reality Composer are considered digital and immersive simultaneously. **Tracking Mode**—Analyzed tools support a great variety of tracking modes (Table 2). The modes supported by most tools are surface tracking (10 cases); image tracking (9 cases); and face tracking (5 cases). The most versatile tool is Lens Studio followed by Geenee AR.

Device Compatibility—Most tools (Table 3) support mobile devices (11 out of 12), but there are also tools that support eyewear devices (3). Most tools support iOS operating system (11) and secondly Android (8). Most versatile tool in terms of device compatibility is XR + , covering Android and iOS mobile devices, but also eyewear devices (Oculus Quest, Oculus Go, Oculus Rift, and HTC Vive).

Publishing—Forms of publication diverge among platforms (Table 4). Most part (7 cases) allow for Web publishing—known as Web AR, which does not require the installation of an application and AR content is accessed through a link, QR Code or AR Code. Four tools allow the publication in the same platform used for the prototype creation; one tool allows for the publication in its own app; and two other tools allow for the integration of the prototype in an existing app. One tool

Table 2 Tracking modes enabled by each platform. * The software does not use tracking because it runs on AR/VR headsets

	Marker/ QR code	Image	Full body	Face	Shoulder	Upper body	Hands	Foot	Cats/ Dogs	Object/ Model	Content around user	Surface	World Mesh	Geo Anchor	Area Target	N/ A *
Geenee AR	X	X	X	X			X					X		X	X	
Adobe Aero		X										X				
PlugXR Creator		X										X				
Spark Studio	X		X	X			X					X				
XR +		X		X			X	X		X		X				
Blipparbuilder		X									X	X				
Byldr																X
JigSpace												X				
Lens Studio		X	X	X	X	X	X		X			X	X			
Reality Composer		X		X						X		X				
WorldCAST	X	X										X				
Hololink		X														

Table 3 Device compatibility by platform

	Android	iOS	Windows	Oculus Quest	Oculus Go	Oculus Rift	HTC Vive	Magic Leap	Hololens 2
Geenee AR	X	X							
Adobe Aero		X							
PlugXR Creator	X	X							
Spark Studio	X	X							
XR +	X	X		X	X	X	X		
Blipparbuilder	X	X						X	
Byldr				X					X
JigSpace		X	X						
Lens Studio	X	X							
Reality Composer		X							
WorldCAST	X	X							
Hololink	X	X							

allows publishing in Web VR and another in AR Quick Look. There are even tools dedicated to social media content, such as Lens Studio and Spark Studio.

6.2 Tools Limitations

After collecting and characterizing the available tools, it was necessary to determine their limitations compared to the Unity game engine. To this end, a matrix was constructed summarizing the main milestones involved in the development of FLOC in Unity (Table 5) and each platform was then tested in the execution of each listed task. The following table presents the milestones of FLOC development (left) and the no-code tools supporting each task (right).

The milestones (M) below mentioned represent only the most important tasks included in the FLOC development:

(M1) Launch an AR experience in space using as area target—concerns the possibility of launching the virtual content through the recognition of an area, an interior space which in the case of FLOC was the exhibition gallery.

(M2) Regarding the creation of multiple scenes and the navigation between them was essential to integrate in FLOC a main view—composed by all the units of the database, and a secondary view (summary view) in which the same information was represented but in an integrated way. The existence of two views enabled the coexistence of two different representation paradigms and their comparison.

(M3) Refers to the software's ability to read and analyze a database, allowing objects to be instantiated directly from that database. This feature was essential, and

Table 4 Publishing/viewing mode by platform

	Web AR	Web VR	AR Quick Look	Platform used to create	Viewing Application	Integration into existing app	Instagram	Snapchat	Facebook	Facebook Lite	Messenger
Geenee AR	X										
Adobe Aero				X							
PlugXR Creator	X			X		X					
Spark Studio							X		X	X	X
XR +	X	X									
Blipparbuilder	X			X							
Byldr					X						
JigSpace	X										
Lens Studio								X			
Reality Composer			X	X		X					
WorldCAST	X										
Hololink	X										

Table 5 Matrix identifying the no-code tools that support important development milestones identified in FLOC

N	Development milestones with Unity (based on FLOC)	Geenee AR	Adobe Aero	PlugXR Creator	Spark Studio	XR +	Blipparbuilder	Byldr	JigSpace	Lens Studio	Reality Composer	WorldCast	Hololink
1	Launch an AR experience in space using an area target	X											
2	Create multiple scenes and the navigation among them	X		X	X	X	X	X		X	X		X
3	Instantiate objects directly through the reading and analysis of a database, with each database entry corresponding to an instantiated object												

(continued)

Table 5 (continued)

N	Development milestones with Unity (based on FLOC)	Geenee AR	Adobe Aero	PlugXR Creator	Spark Studio	XR +	Blipparbuilder	Byldr	JigSpace	Lens Studio	Reality Composer	WorldCast	Hololink
4	Attribute responsive behaviors to tap or click, including a change in the visual aspect of the object clicked and the display of a caption		X	X	X	X		X		X	X		
5	Import 2D images	X	X	X	X	X	X	X	X	X		X	X
6	Create a graphical user interface (screen canvas)				X					X			

of all the tasks it stands out as being the most relevant for the creation of FLOC, since it automated the representation of 2.714 companies, an action that would not have been feasible manually. Reading and analyzing the database also made it possible to assign a caption to each sphere (through tap) providing details about the selected company.

M4) Consists in assigning responsive behaviors to virtual objects. In FLOC this possibility materialized, for example, in the summary view, where, by clicking on the spheres, the visual aspect of that object changed, and a caption was presented—in this case a caption built manually and not by accessing the database. The importation of 2D images allowed, in the case of FLOC, to import the map of Portugal—an object that although simple, allowed to contextualize the experience geographically. The importation or creation of 3D models was also essential to the creation of FLOC, but is not included in the matrix since it is a task supported by all the tools analyzed.

M5) Finally, the possibility of creating a graphical user interface (screen canvas) supports the existence of buttons in the screen space, instead of virtual buttons. The advantage of the graphical user interface (GUI) in an AR experience is that it allows some buttons or menus to always be within the reach of the user who does not need to find them in the virtual space. In the case of FLOC, the GUI allowed, for example, to access the resume view and the database URL.

None of the tools under analysis can recreate all the tasks involved in FLOC. Milestones 4 and 5 stand out particularly for not being tackled using any of the tools analyzed. None of the tools included in this study allowed the analysis of a database to create or instantiate objects based on the stored data, a crucial step in the development of FLOC. In fact, Reality Composer allows creating two types of traditional AR charts from an imported csv database—bar charts and pie charts—but it does not allow, without coding, to program the instantiation of virtual objects directly from the database. There is a major limitation on the actions that can be performed from reading the database with Reality Composer, which is the only analyzed tool allowing to read a csv file. Without the ability to read, analyze and act based on the data stored in a database, it would not have been possible to obtain a working prototype due to the large amount of data involved. Except for the reading and analysis of a database, essential to algorithmic data representation, no-code tools allow to perform all remaining tasks involved in FLOC. Spark Studio and Lens Studio, both aimed at social media, stand out as the tools allowing to complete more tasks.

During the execution of the tasks described in Table 5, it was possible to observe less flexibility across all the no-code tools included in the study—a large part of the tasks can be executed, but in a predefined or closed way compared to Unity, offering much less creative freedom. On the other hand, they are far less complex tools and make the accomplishment of each task faster. Lower flexibility accompanied by lower complexity makes the whole AR creation process significantly easier and less frustrating or painful, as Gandy and MacIntyre point out, since it is quite straightforward to get a functional AR experience.

6.3 A No-Code Project—Pencils

Although Iliinsky and Steele [11] distinction between infographics and data visualization is too strict to fully comprehend this field, it offers a useful working basis for contextualizing AR data representation that is designed with no-code tools. Data representation in which the final visual representation is not done algorithmically is typically the type of visualization that no-code tools support, as they typically do not allow to read and analyze a database. However, this type of tool allows to develop the communication dimension mentioned by Masud et al. [12], which consists in exploring visual metaphors and storytelling to communicate results and to engage audience.

Pencils is a prototype intended to illustrate what an AR data representation using exclusively no-code tools can be. The project was developed to relate the information about the Portuguese design companies, specifically information about the revenue providing from sales and services (in K€) – the turnover—and information about the export percentage per NUT II (regions of Portugal: North, Center, MLA, Alentejo, Algarve, Madeira and Açores) in 2019. Table 6 synthetizes the data represented in this AR prototype, which was retrieved from the database about design companies registered with the code 7410 (design activity) in Portugal in the year 2019 [9], the same used in FLOC.

The software employed in the development of the project was Adobe Aero to create the AR experience and Cinema 4D to model and animate the 3D objects. Although Adobe Aero is not among the most versatile platforms in terms of tracking modes, device compatibility or publishing location, it is among the few platforms supporting both digital and immersive modes. Reality Composer also covers both modes, but its use on a computer is now integrated in Xcode—an iOS developer tool. By supporting digital and immersive modes simultaneously, Adobe Aero makes the AR design process very fluid, enabling the user to design in the digital environment and preview content in the immersive environment, switching between viewing modes very quickly. This feature resulted in a quick execution of the tasks listed in Table 5, which was decisive for its selection. The purpose of developing a project

Table 6 Data on Portuguese design companies represented in Pencils project

NUTS II	Year	Sales (K€)	Services (K€)	Turnover (K€)	Export (%)
North	2019	61,714	82,820	144,534	11,43
Center	2019	10,768	29,703	40,471	9,10
AML	2019	38,693	112,255	150,947	8,64
Alentejo	2019	997	7394	8391	10,85
Algarve	2019	5380	10,204	15,584	7,21
Açores	2019	384	939	1323	10,12
Madeira	2019	822	3589	4411	7,06

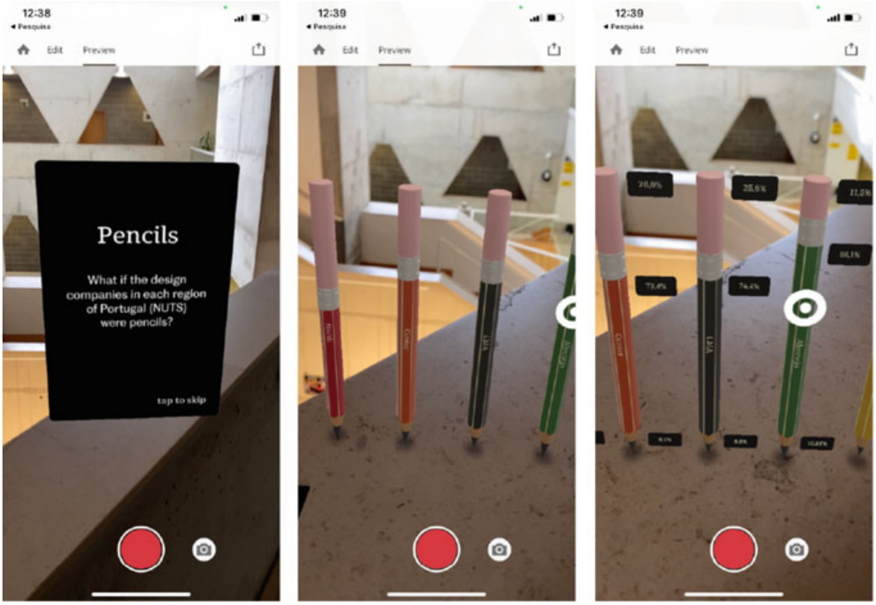
from data previously analyzed was to illustrate only the communicational phase, in which a visual metaphor is created.

In this case, the exercise was to find a way of compiling data into a fluid narrative using a significant visual metaphor, considering the subject matter. The iterative analysis and reflection were carried out together with the co-authors in periodic meetings, to design and progressively improve the experience, define possible paths from the communication point of view, such as the use of other visual metaphors as well as aspects from the design domain—size of the virtual objects, three-dimensional modelling, color, and aspects related to interaction. These meetings also served to reflect on the limitations of the AR experience in context and potential aspects to correct and improve, such as the distinction between services and products and the change of color, the use of other forms of target, the scale of the experience and other interaction possibilities.

6.3.1 Project Description

Pencils (Fig. 4) portrays the landscape of design companies in the seven regions (NUTS II) of Portugal—Norte, Centro, AML, Alentejo, Algarve, Açores and Madeira—concerning their turnover—similarly to FLOC application.

Each region is represented by a pencil whose physical configuration communicates information about the companies in that region (Fig. 5)—rubber represents



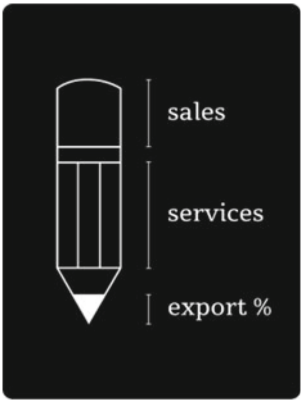
Pencils

What if the design companies in each region of Portugal (NUTS) were pencils?

tap to skip

Fig. 4 Pencils screen captures

Fig. 5 Pencils caption



sales, wood represents services, and the charcoal bill represents the export percentage. The use of pencils as a visual metaphor to represent design companies was prompted by a second phase of user tests on the latest version of FLOC—which we named FLOC II, where one participant mentioned that visually the experience was a bit ‘boring’ but that she understood as it was an informational piece. Although it is really an informational piece, we tried to integrate the visual metaphor in Pencils as a way to emphasize the ludic and communicational character of the experience, since information and entertainment don’t necessarily have to be incompatible, as the Immersive Journalism field has already demonstrated.

The AR experience can be accessed by QR code (Fig. 6) using an iOS device with Adobe Aero installed (free), from any location, as the target is a horizontal plane that can be a tabletop or even the floor. In Pencils, the user progresses through a short narrative that unfolds in several events (Fig. 7): first the project is introduced in a brief succession of panels; then pencils appear in the scene representing the percentage of product sales, services, and exports. In the third phase, percentage values are replaced by absolute values (€), showing the significant differences in turnover by region. In the following scene, pencils corresponding to the lowest turnover regions are combined into a single pencil, enabling the comparison with the two regions exhibiting the highest turnover. At last, pencils return to their initial configuration and are packed away in a box, closing the narrative, and enabling the user to access the URL where the databases supporting the experience are stored.

By aggregating multiple values into a smaller number of visual marks, Pencils fits into traditional data visualization. Differently, FLOC belongs to what [30] designate as immersive unit visualization—visualizations in which every data point is represented by a separate visual mark—while simultaneously offering a traditional, aggregated view of the data—the resume view. In Pencils we synthesize information and provide a data overview, while in FLOC there is no synthesizing exercise. Both projects aim to make a numerical abstraction concrete, but they do so in radically different ways and the tools per se seem to offer different forms of thinking and therefore visualizing. If we think about the distinction between infographics and

Fig. 6 QR code to access Pencils

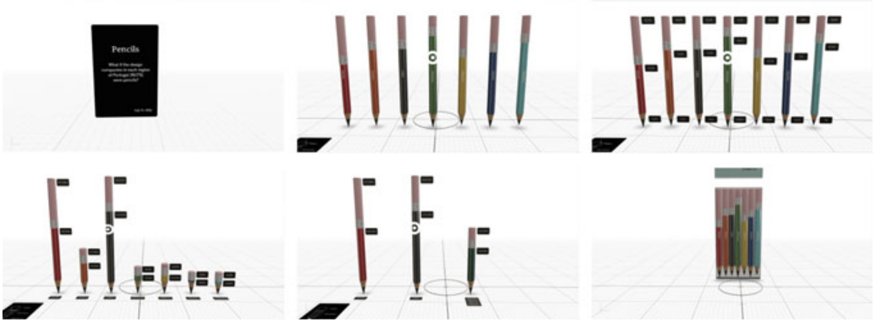


Fig. 7 Screen captures (Adobe Aero on desktop) showing Pencils narrative sequence

visualization criteria [11], FLOC fits better in data visualization and Pencils in infographics. FLOC supports a large amount of data, was designed algorithmically, and can be automatically reproduced with different data. Pencils was designed manually, with a personalized treatment of the information, and it would be laborious to recreate with different data. Due to the personalized treatment of information, the amount of data included is limited. On the other hand, in Pencils there was a greater investment in the design of the experience and narrative. While developing FLOC, the concerns with issues related to programming took away time and attention that could have been devoted to the design of the experience.

7 Discussion

No-code tools are no substitute for full implementation tools in data visualization as they are limited regarding the reading and analysis of a database, which are essential to algorithmically based visualization, without which the representation of large datasets is not viable. This limitation is more evident in the case of immersive unit visualization—where data is not aggregated, instead each unit is individually represented, making manual representation unfeasible or extremely time-consuming. Furthermore, no-code tools offer much less flexibility compared to more complex tools allowing to code and may also offer limitations in terms of scene size. Despite being less flexible than full implementation tools (for instance, no-code tools offer a limited number of interaction behaviors, which is not the case with development platforms relying on scripting), no-code tools allow to more quickly create visual metaphors capable of communicating a dataset. Their main advantage is the easy access to the creation of AR content, being a gateway for designers with no previous experience designing for AR. In fact, it is important to bring designers closer to all technologies, allowing them to broaden their field of action.

Adobe Aero low complexity, like other no-code tools, allows for quick design and brings fluidity to the development process, but it doesn't exactly add fluidity of thought. No-code tools automate development flows that are not automated in game engines which need to allow greater flexibility. For instance, when creating AR experiences with a no-code tool, the project evolves based on the tracking mode selection. In game engines, designers create their own path without a predefined development flow, which requires additional experience (they can even include several tracking modes simultaneously). No-code tools allow simple ideas to be quickly realized, but within a closed set of possibilities. Development tools and no-code tools are complementary and used at different moments. In any case, the following question arises: is it worth designing with these limited mechanisms or is it better to design using systems already mastered and ask translators (programmers) to then make these applications for a new support or medium? We consider that it is crucial that the design of an AR experience necessarily goes through contact with AR, even if it is with limited tools, because they allow to visualize the project in the hybrid space and this contact raises relevant questions related to the medium, which designing in other formats could not raise.

Still from the perspective of design, other question arises which is whether the designer should position himself aside of coding and entirely depend on someone who translates his thinking into a programming language. *Tradurre è tradire* (To translate is to betray), so it will always be desirable for a designer to think already from a digital grammar, but will no-code apps have a lexicon already broad enough for this design to be capable or creator of the new? Or is this loss greater than the translation of a more robust design proposal later translated by expert technicians? Although they are good gateways to the AR medium, no-code tools might excessively lead the project to a final standard solution. But then again, the more people engage with these tools, more solutions will arise.

Even today, most designers still draw manually, doodling (even if it's on an iPad), because there seems to be an immediate connection between thought and drawn images, and even because chance and serendipity plays an essential role in this process. Some students, try to do it directly on the computer, already with final applications, which somehow dulls their thinking—they think according to the limits of that machine, or let the machine manage their thinking. It is important that the no-code tools are assumed by designers as a gateway, and even as a form of rapid prototyping, but not as the exclusive place where the project emerges and is developed. Such modus operandi would conform any AR project to an always identical matrix, annulling the free exploration of new development paths—on which the construction of a proper language for this medium depends.

8 Conclusions

This paper aimed to identify, describe and compare AR no-code tools currently available online, outline their limitations regarding other development platforms like Unity and propose a project for data representation with AR using exclusively no-code tools.

By presenting alternative tools to game engines and development platforms that require programming knowledge, and by illustrating a possible use in the field of data visualization, we expect to make the appropriation of this technology by information designers more accessible, at least at an early stage of their introduction to the AR field.

This study also has limitations which in turn, point to future research directions. The absence in literature of criteria for an AR authoring tool specifically suited to the activity of designers would have been useful for the description and evaluation phases. As a future study, it would be interesting to compare the interpretation of Pencils with a traditional two-dimensional graph representing the same information, to ascertain on what specific criteria can AR surpass a traditional chart in terms of information communication.

Table A Authoring tools characterization

Platform	Static/ Interactive	Interaction Mode	Type of Authoring Tool	Tracking Modes	Compatibility	Publishing
Geenee AR	Static	N/A	Digital	Marker/ QR Code; Image; Full body Face; Hands; Surface; Geo Anchor (Beta); Area	Android; iOS	Web AR
Adobe Aero	Interactive	Drag and drop	Digital and immersive	Image; Surface	iPhone; iPad; iPad Touch	Adobe aero app
PlugXR Creator	Interactive	Scripts editing via GUI	Digital	Image; Surface	iOS; Android	PlugXR App; WebAR; integration into existing app
Spark Studio	Interactive	Drag and drop	Digital	Marker/ QR Code; Full body; Face; Hands; Surface	Android; iPhone; iPad	Effects for Instagram; Facebook; Facebook Lite and Messenger
XR +	Interactive	Scripts editing via GUI	Digital	Image; Face; Hands; Foot; Object/ Model; Surface	iPhones; iPads; Android; Oculus Quest; Oculus Go; and the combo Firefox/ Chrome + Oculus Rift/ HTC Vive on desktop computers	Web AR and Web VR (for smart glasses or VR headset)
Blipparbuilder	Interactive	Scripts editing via GUI	Digital	Image; Content appearing around the user; Surface	Android; iOS; Magic Leap	Web AR and Blippar App

(continued)

Table A (continued)

Platform	Static/ Interactive	Interaction Mode	Type of Authoring Tool	Tracking Modes	Compatibility	Publishing
Byldr	Interactive	Scripts editing via GUI	Immersive	N/A	Hololens 2; Oculus Quest	Byldr Marketplace
JigSpace	Static	N/A	Digital	Surface	iPhone; iPad, Mac and Windows computers	Web AR
Lens Studio	Interactive	Scripts editing via GUI	Digital	Image; Full body; Face; Shoulder; Upper body; Hands; Cats and dogs	iOS and Android devices	Snapchat filters
Reality Composer	Interactive	Scripts editing via GUI	Digital and immersive	Image; Face; Object/ Model; Surface	iPhone; iPad	Reality Composer; AR Quick Look; Integration into apps
WordCast	Static	N/A	Digital	Marker/ QR Code; Image; Surface	iOS; Android	Web AR
Hololink	Interactive	Scripts editing via GUI. Very limited interaction options	Digital	Image	iOS; Android	Web AR

Acknowledgements This work is financed by national funds from FCT – Fundação para a Ciência e a Tecnologia (Foundation for Science and Technology), I.P., under the PhD grant «SFRH/BD/144937/2019». This article is a result of the project Design Obs. Para um Observatório de Design em Portugal: Modelos, Instrumentos, Representação e Estratégias, (Towards a design Observatory in Portugal: models, instruments representation and strategies) supported by Lisbon Regional Operational Programme (LISBOA 2020) and the Competitiveness and Internationalisation Operational Programme (POCI-01-0145-FEDER-032445), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF) and FCT—Fundação para a Ciência e a Tecnologia (Foundation for Science and Technology).

References

1. Marques AB, Branco V, Costa R, Costa N (2022) Data Visualization in Hybrid Space—Constraints and Opportunities for Design. In: Martins N, Brandão D (eds) Advances in Design and Digital Communication III. DIGICOM 2022. Springer Series in Design and Innovation. Springer, Cham, pp 3–15
2. Branco V, Costa R, Costa N, et al (2018) designobs.pt. https://designobs.pt/. Accessed 21 Mar 2022
3. Schmalstieg D, Höllerer T (2016) Authoring. Augmented Reality: Principles and Practice. Addison Wesley, Crawfordsville, Indiana, pp 471–488
4. Leiva G, Nguyen C, Kazi RH, Asente P (2020) Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp 1–13
5. Dengel A, Iqbal MZ, Grafe S, Mangina E (2022) A Review on Augmented Reality Authoring Toolkits for Education. Front Virtual Real 3:https://doi.org/10.3389/frvir.2022.798032
6. Gasques D, Johnson JG, Sharkey T, Weibel N (2019). What You Sketch Is What You Get: Quick and Easy Augmented Reality Prototyping with PintAR. https://doi.org/10.1145/3290607.3312847
7. MacIntyre B, Gandy M, Dow S, Bolter JD (2005) DART: a toolkit for rapid design exploration of augmented reality experiences. ACM Trans Graph 24:932–932. https://doi.org/10.1145/1073204.1073288
8. Verganti R, Vendraminelli L, Iansiti M (2020) Innovation and Design in the Age of Artificial Intelligence. J Prod Innov Manag 37:212–227. https://doi.org/10.1111/jpim.12523
9. Costa N, Branco V, Costa R, et al (2021) Design Companies in Portugal (detailed view, 2018, SABI)
10. “mbs278” (2017) Beginning Data Visualization in Unity: Scatterplot Creation – Methodology blog. http://sites.psu.edu/bdssblog/2017/04/06/basic-data-visualization-in-unity-scatterplot-creation/. Accessed 27 Mar 2023
11. Iliinsky N, Steele J (2011) Designing Data Visualization. O’Reilly, Sebastopol, CA, USA
12. Masud L, Valsecchi F, Ciuccarelli P, et al (2010) From Data to Knowledge – Visualizations as Transformation Processes within the Data-Information-Knowledge Continuum. In: 2010 14th International Conference Information Visualisation. IEEE, pp 445–449
13. Segel E, Heer J (2010) Narrative visualization: Telling stories with data. IEEE Trans Vis Comput Graph 16:1139–1148. https://doi.org/10.1109/TVCG.2010.179
14. Marques AB, Branco V, Costa R (2022) Narrative Visualization with Augmented Reality. Multimodal Technologies and Interaction 6:105. https://doi.org/10.3390/mti6120105
15. Ikonen P, Uskali T (2021) Augmented reality as news. In: Uskali T, Gynnild A, Jones S, Sirkkunen E (eds) Immersive Journalism as Storytelling. Routledge, Oxon, pp 147–160
16. New York Times (2022) Inflation Shrinking Ray. https://www.nytimes.com/interactive/ 2022/04/20/business/inflation-shrink-ray.html?shared=1. Accessed 27 Sep 2022
17. New York Times (2022) EV Battery Breakdown. https://www.nytimes.com/interactive/ 2022/08/19/business/electric-vehicle-battery-made.html. Accessed 27 Sep 2022
18. Deutsche Welle (2020) Water footprint: Augmented Reality shows water cost of beef. https://www.dw.com/en/water-footprint-augmented-reality-shows-watercost- of-beef/a-52218248. Accessed 29 Sep 2022
19. Cappannari L, Vitillo A (2022) XR and Metaverse Software Platforms. In: Roadmapping Extended Reality. Wiley, pp 135–156
20. Li J, W. Tigwell G, Shinohara K (2021) Accessibility of High-Fidelity Prototyping Tools. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp 1–17
21. Billinghurst M, Nebeling M (2022) Rapid prototyping of XR experiences. ACM SIGGRAPH 2022 Courses. ACM, New York, NY, USA, pp 1–124
22. Lee GA, Kim GJ, Billinghurst M (2005) What you experience is what you get (WYXIWYG). Commun ACM 48:76–81. https://doi.org/10.1145/1070838.1070840

23. Billinghurst M, Clark A, Lee G (2015) A Survey of Augmented Reality. Foundations and Trends® in Human–Computer Interaction 8:73–272. https://doi.org/10.1561/1100000049
24. Freitas G, Pinho MS, Silveira MS, Maurer F (2020) A Systematic Review of Rapid Prototyping Tools for Augmented Reality. In: 2020 22nd Symposium on Virtual and Augmented Reality (SVR). IEEE, pp 199–209
25. Seichter H, Looser J, Billinghurst M (2008) ComposAR: An intuitive tool for authoring AR applications. In: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. IEEE, pp 177–178
26. Gandy M, MacIntyre B (2014) Designer's Augmented Reality Toolkit, Ten Years Later: Implications for New Media Authoring Tools. In: Proceedings of the 27th annual ACM symposium on User interface software and technology. ACM, New York, NY, USA, pp 627–636
27. Marques AB, Branco V, Costa R (2021) Minard Revisited – Exploring Augmented Reality in Information Design. Springer, Cham, pp 79–89
28. Marques AB (2023). No-Code Augmented Reality Platforms. https://doi.org/10.5281/ZENODO.7640671
29. Stansfield C, Dickson K, Bangpan M (2016) Exploring issues in the conduct of website searching and other online sources for systematic reviews: how can we be systematic? Syst Rev 5:191. https://doi.org/10.1186/s13643-016-0371-9
30. Ivanov A, Danyluk K, Jacob C, Willett W (2019) A Walk Among the Data. IEEE Comput Graph Appl 39:19–28. https://doi.org/10.1109/MCG