DynaTags: Low-Cost Fiducial Marker Mechanisms

Cassandra Scheirer
Carnegie Mellon University
Pittsburgh, PA, USA
cscheire@andrew.cmu.edu

Chris Harrison
Carnegie Mellon University
Pittsburgh, PA, USA
chris.harrison@cs.cmu.edu

ABSTRACT

Printed fiducial markers are inexpensive, easy to deploy, robust and deservedly popular. However, their data payload is also static, unable to express any state beyond being present. For this reason, more complex electronic tagging technologies exist, which can sense and change state, but either require special equipment to read or are orders of magnitude more expensive than printed markers. In this work, we explore an approach between these two extremes: one that retains the simple, low-cost nature of printed markers, yet has some of the expressive capabilities of dynamic tags. Our "DynaTags" are simple mechanisms constructed from paper that express multiple payloads, allowing practitioners and researchers to create new and compelling physical-digital experiences. We describe a library of 23 mechanisms that can be read by standard smartphone reader apps. Through a series of demo applications (augmenting reality through e.g., sounds, environmental lighting and graphics) we show how our tags can bring new interactivity to previously static experiences.

CCS CONCEPTS

• **Human-centered computing** → **Human-computer interaction**; • **Interaction paradigms** → *Mixed / augmented reality*.

KEYWORDS

tangible interaction, augmented reality, paper interfaces, shape changing, computer vision, widgets

ACM Reference Format:
Cassandra Scheirer and Chris Harrison. 2022. DynaTags: Low-Cost Fiducial Marker Mechanisms. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '22), November 7–11, 2022, Bengaluru, India.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3536221.3556591

1 INTRODUCTION

Printed fiducial markers — such as QRCodes [9], barcodes [25] and ArUco Markers [13, 30] — are now ubiquitous. They are seen taped to windows, walls and tables, as well as printed on signage, documents and business cards, among many uses. The key to their popularity is *low cost* (one can even print them at home for <1 cent), *ease of deployment* (requiring no special skills, tools or infrastructure to install), and *robustness* (both in terms of physical durability and also reliability across smartphone reader apps).

ICMI '22, November 7–11, 2022, Bengaluru, India

ACM ISBN 978-1-4503-9390-4/22/11.
https://doi.org/10.1145/3536221.3556591

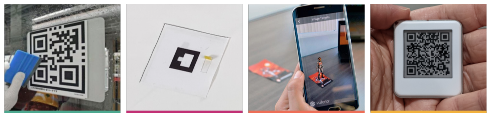
While there are SDKs that can detect objects/images without structured patterns (e.g., Vuforia [39], ARKit [2], ARCore [16]), these are more *complex to use* (requiring a fairly high degree of technical proficiency), *less reliable* (especially at longer ranges, with low-detail references, or in poor lighting conditions) and must be *pre-registered* with the reader device (unlike a standards-based scheme e.g., QRCodes [9]). As a consequence, they are rarely used beyond special-purpose experiences requiring a *custom app*. Importantly, in both cases, the data payload of printed images is inherently static (i.e., one marker maps to one value). To achieve multiple values or states, lay users often use multitple markers on the same document. Figure 1 provides several real-world examples, including a restaurant menu, school test, business card, store placard, and children's book.

In contrast to printed markers, electronic tags are dynamic, able to change in value and even relay changes in their state or environment. This sophistication increases cost and complexity — either the tag itself is reasonably capable (e.g., microcontrollers with wireless communication [31, 34]), or low-cost but require a specialized reader (e.g., RFID tags, Sozu [45], Vibrosight [46]). In both cases, the price and skill required to deploy electronic tags is much higher than printed fiducial markers, and for this reason, it is rare to see such systems deployed in the public sphere (e.g., a local restaurant).

While both printed and electronic tags have their strengths and key use domains, we believe there is value in a middling approach, one with dynamic, state-changing capabilities that also retains the simple, easy, and low-cost nature of printed markers. A high-level overview of this design space is seen in Figure 2.

In response, we created DynaTags, simple paper mechanisms that extend the capabilities of commonplace fiducial marker schemes, enabling new multimodal physical-digital experiences. These tags can be printed at home or work and require no special skill to assemble, yet they are able to express a variety of useful states, making previously static experiences more interactive and flexible.

Figure 1: Real-world examples of end-users using multiple markers to express several states or functions. From left to right: menu with different sub-menus [24], school test with multiple choice answers [11], business card with two language options [6], business placard with different payment and service options [38], and children's book with different clothing choices [35].



	Printed Fiducials	DynaTags	AR SDKs	Electronic Tags
Unit Cost	Low	Low	Low	High
Tag Expressivity	None	Low	None	High
Deployability	Easy	Easy	Hard	Hard
Standard or Custom App	Standard	Standard	Custom	Standard
Make "at Home"	Yes	Yes	Yes	No
External Infrastructure	No	No	No	Yes

Figure 2: Pros and cons of DynaTags in comparison to other major tagging technologies.

By building on conventional and proven fiducial marker schemes, our system is robust and works "out of the box". Indeed, DynaTags are already compatible with reader apps installed on billions of smartphones, both web-based and native, and users need not learn anything new to engage with them. This stands in contrast to image-based computer vision matching schemes (e.g., Vuforia [39], Apple's ARImageAnchor API [3], Unity's Image Tracking [37]), which require pre-registration, custom apps, proper sizing and lighting, and complexity in content.

After reviewing related work, we describe different dynamic payload strategies and a wide variety of paper mechanisms we developed to showcase the generalizability of our approach. We conclude with a series of example applications that incorporate our mechanisms, highlighting new and creative interactive uses for fiducial markers.

2 RELATED WORK

We first briefly review fiducial marker schemes, which are now in widespread use. We then discuss "sensor tags", with a particular emphasis on low-cost systems that might compete with printed tags. We conclude with a more focused discussion on work most related to our own: passive and low-cost tags that have at least some dynamic capabilities. Figure 2 provides a high-level overview of the pros and cons of popular tagging technologies.

2.1 Passive Fiducial Markers

Many passive (i.e., unpowered) approaches have been explored for encoding unique identifiers, including magnetic strips (like that on the back of a credit card), physically-notched acoustic barcodes [20], and patterned ferromagnetic ink [26]. However, visual schemes — graphic patterns readable by laser scanners or RGB cameras — are by far the most popular. These include 1D barcodes [25, 28], 2D barcodes [9], and various fiducial markers schemes [13, 22, 23, 29, 30] (see [14] for a more comprehensive review). Most importantly, these markers can be printed on paper and are robust, both in terms of software recognition and physical durability. While passive, unstructured images (e.g., logos, magazine covers, photographs) can be recognized and tracked using computer vision (e.g., Vuforia [39], Apple's ARImageAnchor API [3]), this type of content requires pre-registration. More problematic is that accuracy is not as robust as high contrast, highly-structured visual tagging schemes. For this reason, supermarkets use barcodes and not image-based computer vision to recognize their thousands of products.

There have also been efforts to increase the payload of fiducial markers by making them customizable and more visually expressive. d-touch [10] is an open-source system that allows users to draw their own markers, so long as they abide by topology-based tracking constraints. With the ARTTag system [21], artistic images can be transformed into markers by strategically placing circles within the image. Aesthetic QR Codes [44] blend noise-based QR codes and artistic images together. With all of these examples, the markers can be designed so that their appearance reflects their purpose; they increase payload by offering information even before they are scanned. However, each marker is still only associated with one ID.

2.2 Low-Cost Sensors and Mechanisms

Self-contained IoT-style "tags" and "sensor boards" that require wall power or batteries are sold commercially by several large vendors (e.g., Texas Instrument's SensorTag [34], STMicroelectronics' SensorTile [31], ONSemi's Multi-Sensor Board [18, 27, 40]). Similarly, researchers have created tags/mechanisms that backscatter or reflect energy, such as RF [33, 41, 42] or laser light [46]. Alternatively, mechanisms can harvest their own energy, as demonstrated in Sozu [45] and OptoSense [43].

Although the aforementioned unpowered tags can be manufactured inexpensively (a few cents or dollars), they require special-purpose readers that are often hundreds of dollars. Put simply, consumers cannot read these tags using devices they already own (i.e., smartphones), making them unlikely to be adopted for consumer uses. Additionally, deploying and managing fleets of these boards require a reasonable level of technical sophistication and up front investment, which will likely hinder their adoption in small, independent stores and businesses (i.e., not chain stores with dedicated IT departments).

2.3 Dynamic Fiducial Markers and Mechanisms

Most related to our work are prior fiducial marker approaches that have the ability to appear/disappear or modify their payload in some manner. Chief among this related work is "Tangible Interfaces with Printed Paper Markers" [47], on which this work builds. Like DynaTags, the latter work also used simple paper mechanisms with printed fiducial markers to convey state in tangible experiences. For example, a "push button" mechanism is implemented using folds in a piece of paper, that when depressed, brings two sides together to form a complete AruCo Marker, which can then be detected. This is similar to our abutting-style payload technique described later. The authors also put forward the notion of tearing tags, wetting them with water, and stretching a kirigami pattern to render tags undetectable. A series of applications are presented, including a game controller and augmented popup book.

Our work extends [47] in several important ways. Foremost, in the prior work, state change is achieved through the binary presence or absence of a tag. In order to achieve multiple states, multiple tags must be used (as typified by their slider mechanism, which is made from a series of push buttons). In contrast, most of our mechanisms change the inner payload of a tag, and can achieve multiple states using a single tag. Second, a fixed camera is used in [47] because if a tag is removed from the field of view, it is ambiguous to the system if the tag has been activated/deactivated, or simply

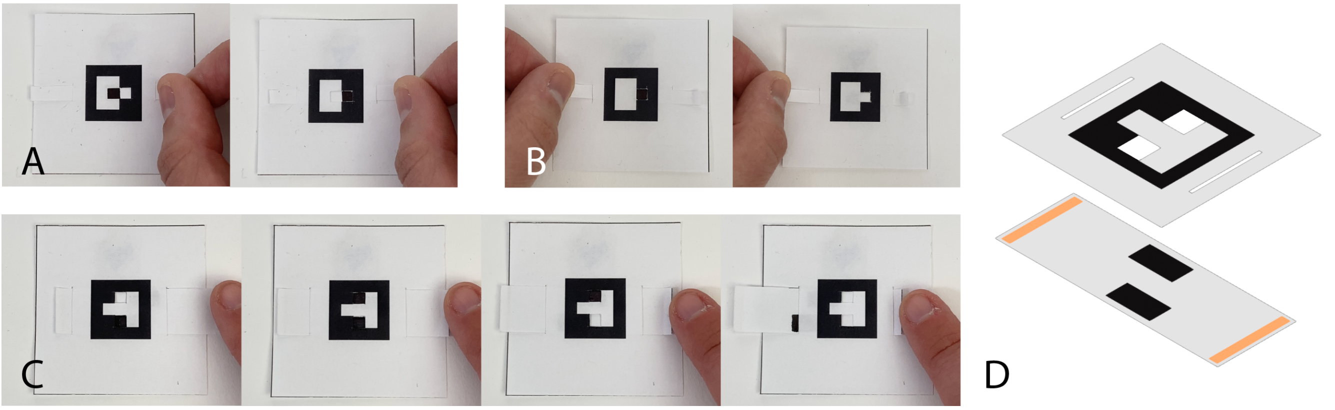


Figure 3: Example pixel window sliders using a 2x1 window (A), a 1x1 window (B), and two 1x1 windows (C). Illustration in (D) shows how the layers of (C) are constructed.

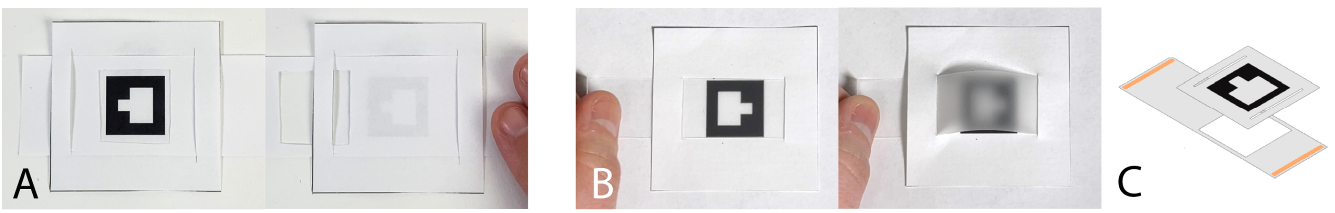


Figure 4: Example abutting sliders that abut two pieces of one marker (A) and three markers in sequence (B). Illustration in (C) shows how the layers of (B) are constructed.

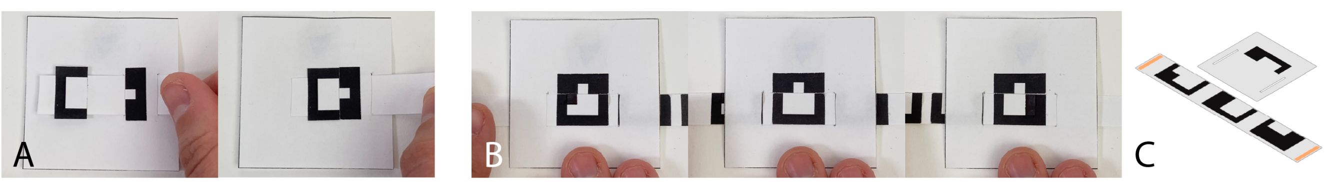


Figure 5: An example slider that conceals/reveals a marker (A), or blurs/unblurs a marker (B). Illustration in (C) shows how the layers of (A) are constructed.

removed. In contrast, our technique works with a standard and unconstrained smartphone, requiring no special apparatus (such as the webcam underneath glass in [47]). Third, our work usefully and significantly extends the mechanism vocabulary presented in [47], offering practitioners a wider array of mechanisms from which to build experiences. Indeed, we present a series of example applications we created using our pipeline to illustrate the generalizability of DynaTags. Finally, we also experimentally evaluate our approach, finding our mechanisms are robust for many tens of thousands of actuation cycles, and at ranges up to 2.0m.

Another relevant work is "Stacks on the Surface" [5], which presented a fiducial marker design that permits stacking of markers. A camera operating beneath the markers (in this case, a diffuse illumination multi-touch table) can resolve the presence and ordering of markers on the surface. This is achieved using structured holes in the markers that allows the camera to interrogate up into the stack of markers. Drogemuller et al. [12] is also related, which uses fiducial tags to instrument real-world objects for use as tangible interfaces in VR/AR. The authors describe a folding and accordion interaction, which is like an input mechanism. Finally, we note there are color changing fiducial markers, sometimes called printed sensors [15, 19], which use functional inks to change their data payload in response to e.g., water or heat exposure. These act more as sensors than user input mechanisms, but one could imagine co-opting the general idea for HCI purposes, such as using thermochromic ink to detect the heat from a user's finger.

3 DYNAMIC PAYLOAD TECHNIQUES

Fundamentally, for a DynaTag to express different states, one or more payload bits must change (from black to white, or white to black). We identified three pixel-changing techniques from which almost all of our mechanisms (described in Section 4) are derived: pixel windows, abutting, and revealing. We use the example of a slider mechanism to help explain each technique. To simplify our figures and facilitate reader comprehension, we built the example DynaTags used in this section using the 3x3 square binary AruCo standard [13]. However, our technique can work with both larger payload sizes and different marker standards.

Windows. This technique requires two layers, an upper layer having a small cut-out "window" that reveals details on a lower layer. To create a slider, one of the layers is able to be translated by the user, while the other remains fixed. The example slider in (Figure 3A) uses a 2x1-sized pixel window, inside of which a payload pixel can move back and forth. Alternatively, a single-pixel-sized window (i.e., 1x1) can be used, and the sliding action causes a payload pixel to be revealed or occluded (Figure 3B). This idea can be extended to multiple pixel windows that encode 2^n states, such as the four-state slider in (Figure 3C). It is also possible for a DynaTag payload to be manipulated by more than one input slider. The illustration in Figure 3D shows how the layers of a four-state slider (C) can be constructed.

Abutting. An entirely different way to create a slider is through printed features that abut, creating one complete marker that expresses a unique state. As a simple example, the slider in Figure 4A only has a complete (and thus recognizable) marker when it is slid entirely to the right. Alternatively, partial markers can be arrayed along one side of the slider (Figure 4B). Unlike pixel windows, abutting elements can be achieved using one layer. The illustration in Figure 4C shows how the layers of a multi-state abutting slider (B) can be constructed.

Revealing. Finally, and perhaps most straightforward, is to design mechanisms that simply change the detectability of a tag. For instance, a slider could cover or reveal tags to convey its state (Figure 5A). Blurring is another way to obscure or reveal tags — we found that vellum, tracing paper, or other transparent materials are ideal for this effect. When such paper is in direct contact with another layer, it is possible to see the contents underneath. However, if the translucent paper is lifted even slightly, the tag becomes diffuse and unrecognizable (Figure 5B). The illustration in (Figure 5C) shows how the layers of a mechanism like (A) can be constructed. It is also possible to obscure tags permanently either by marking them (e.g., with a sharpie) or by physically destroying them, such as through ripping or puncturing. We outline these mechanisms later in the paper. While these are not mechanisms in the traditional sense, they are manipulations that capture a unique physical state change.

4 EXAMPLE DYNATAG MECHANISMS

Using the dynamic payload techniques described in the previous section, we now briefly describe 23 DynaTag varieties we created. We recommend referring to the Video Figure to gain a visual appreciation for how these mechanisms operate. We note that several of these mechanisms are inspired by origami and kirigami techniques (see e.g., [7, 8, 17, 32] for a survey).

4.1 Window-Based Mechanisms

4.1.1 Basic Slider. A slider is a simple, yet versatile input mechanism. It consists of an upper layer with one or more windows, and a sliver of paper that can be translated underneath. The sliver has a black region and a white region, arranged so that when it is pulled from either end, the window in the upper layer fills with either black or white, presenting two different marker payloads (Figure 6, A & B). Note that more windows n can be added to enable 2^n states, as shown in our later textbook example with two windows and four payload states (see Figure 18 and Video Figure).

4.1.2 Multi-Input Slider. A multi-input slider (Figure 6, C–F) follows the same basic rules as the basic slider, but there are multiple windows and multiple paper slivers underneath. As before, the number of independent windows enables 2^n payload states. This mechanism is particularly useful for applications needing 2D directional movement.

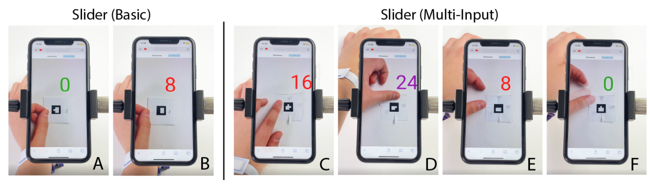


Figure 6: An example of basic slider (A & B) and multi-input slider mechanisms (C–F).

4.1.3 Elastic Slider. Elastic sliders follow the same basic design as our earlier slider in that there is an upper marker with a window and a lower sliver with black and white regions. However, in this mechanism, the sliver is an elastic material that is fixed at one end. Now, instead of the lower sliver translating, it is stretched (Figure 7, A & B). This provides an elastic resistance, while also having the effect of returning the slider to a "home position" when released.

4.1.4 Linear and Rotational Switches. By adding a protruding paper tab to the translatable sliver, we can create a switch-like mechanism. These can be linear (Figure 7, C & D) or rotational in design (Figure 7, E & F).

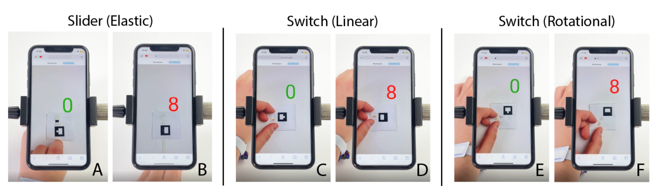


Figure 7: An example of elastic slider (A & B), linear switch (C & D), and rotational switch mechanisms (E & F).

4.1.5 Dial. Our dial mechanism works similarly to a rotational switch, but it is controlled by a central knob that can be continuously rotated, rather than a tab off to the side with a fixed track (Figure 8, A–C). Underneath the upper marker layer is a dial with varying black and white regions, which change the payload via one or more windows. This mechanism is best applied to applications that require triggers or transitions, especially those involving cyclic or twisting motions (e.g., the day-night cycle in our children's book example; see Figure 18 and Video Figure).

4.1.6 *Button.* For a button, the top layer is simply a marker with a window. This is attached to plain white paper via accordion-like paper folds, acting like springs. The resting state of the window is black, due to the shadow that exists when the top layer is sprung above the lower. However, when the marker is pressed, the dark shadow disappears and the white of lower paper is seen (Figure 8, D & E). Regardless of the number of windows employed, this mechanism can only have two states (pressed and not pressed), and thus is most useful for applications needing a discrete trigger.

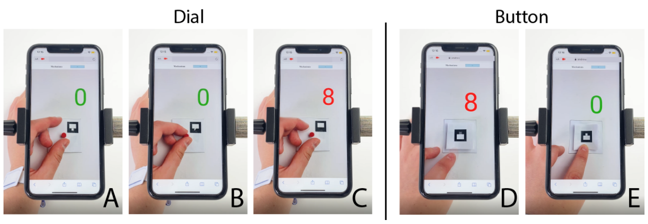


Figure 8: An example of window dial (A–C) and button mechanisms (D & E).

4.1.7 *Stacking.* Windows cut into markers fundamentally let information through from a lower layer. If these are correctly structured, it is possible to express many states, as well as the ordering of the stacked elements [5]. In its simplest form, one can imagine overlaying a tag onto another to change just a single bit, as depicted in Figure 9, A & B. This mechanism is best suited for applications that involve addition/subtraction or consumption/collection.

4.1.8 *Flap.* Flaps are a mechanical variant of stacks. Rather than overlaying an independent element onto another, flaps operate as an integrated mechanism, in which a fixed hinged element overlays onto another. (Figure 9, C & D). This physical affordance makes it well suited for applications that involve concealing/revealing, opening/closing, or locking/unlocking actions (e.g., opening a treasure chest in a board game; see Video Figure).

4.1.9 *Presence.* All of the data payload changes discussed thus far have resulted from manipulations of special-purpose paper mechanisms. However, it would also be useful to have payloads that are dynamic based on the state of an object itself. As one example, we created DynaTags that express the presence or absence of an element, for instance, a wooden dowel inserted during furniture assembly (Figure 9, E–G). To achieve this, we use removable stickers on the outside of the object with a small window that changes color upon insertion (i.e., presence) of the dowel. This can be done with a small white piston that is pushed to the surface, or the dowel itself could be painted white on its ends.

4.2 Abutting-Type Mechanisms

4.2.1 *Basic Plug.* In this mechanism, two pieces of a marker are aligned and connected together, much like puzzle pieces, allowing a marker to be detected (Figure 10, A & B). While two abutted pieces can only express one state, it is possible to create libraries of abutted elements that express many states, as seen in our later tangible music synthesizer demo (Figure 19).

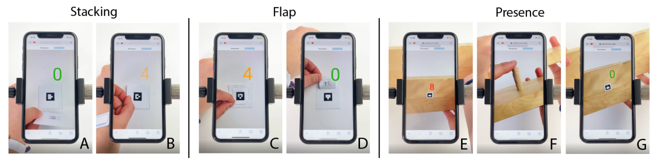


Figure 9: An example of window stacking (A & B), window flap (C & D), and pixel presence mechanisms (E–G).

4.2.2 *Switch (Abutting).* In addition to our window-based switch design, we also created a switch using our abutting primitive. For this, we use a marker sitting on a paper hinge, which can be pushed in two or more ways, such that it abuts a printed element that completes the tag (Figure 10, C & D). This mechanism is most useful for applications that require triggers as well as directional motion.

4.2.3 *Dial (Abutting).* In a similar manner to our abut-type switch, we created a dial mechanism that abuts parts of markers such that when markers are completed, they are able to be read. While the example shown in Figure 10 (E & F) illustrates a single payload, the dial can feature multiple different payloads that are rotated into place, with only one able to be active at a time.

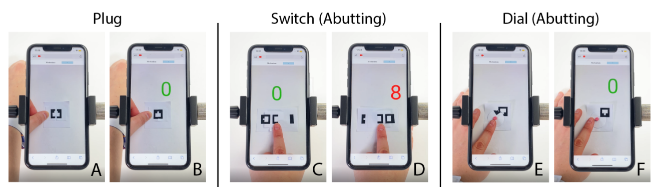


Figure 10: An example of basic abutting plug (A & B), switch (C & D), and dial mechanisms (E & F).

4.2.4 *Flap (Abutting).* A second type of flap can be made with an abutting technique, wherein one section of a marker is placed at the edge of the flap, and the reciprocal section is placed on a layer below the flap. When the flap is closed, the sections meet, and the marker is detectable (Figure 11, A–C).

4.2.5 *Accordion.* In our previous examples, abutted elements were roughly two halves of a fiducial marker. However, more complex arrangements are also possible, such as four corners, and even dozens of small slices. As one complex example, we created an accordion mechanism in which a marker is printed as many small slices on the ridges of a multi-folded piece of paper. When the folds are pushed together, one marker is visible and scannable (Figure 11, D & E). When the accordion is extended, a different marker printed inside is detected.

4.3 Reveal/Conceal Mechanisms

4.3.1 *Flap.* Another way to use markers is to reveal or conceal them with flaps. There could be a marker on a flap that is only detected in the flap's closed state, or there could be a marker below the flap, which is only detected in the flap's open state (Figure 12, A & B).

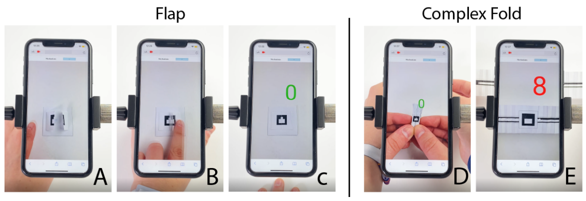


Figure 11: An example of abutting flap (A–C) and complex fold mechanisms (D & F).

4.3.2 Push Blur. A blur button uses thin, translucent material (e.g., tracing paper, vellum, plastic diffuser) that controls the visibility of a marker underneath. When the material is laid flat directly on the marker, the marker is seen through the material. However, when the diffuse material is lifted slightly, scattering the light, it is no longer detectable. Using this effect, we can create a different type of push button mechanism, seen in (Figure 12, C & D).

4.3.3 Pinch Blur. Also using translucent material, we can create a mechanism that requires a user's pinch. This action causes the translucent layer to bulge or pop from the surface, obscuring the marker underneath (Figure 12, E & F). When the pinch is released, the translucent material generally wants to return to its flat state. However, but introducing a preloaded kink or curve in the material, the opposite behavior can be created, where users much pinch the material apart (i.e., stretch) to cause it to lay flat, revealing the marker.

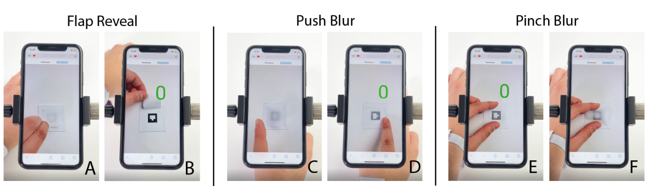


Figure 12: An example of revealing flap (A & B), push blur (C & D), and pinch blur mechanisms (E & F).

4.3.4 Cross-Out. Drawing a large slash or X over a marker has the effect of rendering it undetectable (Figure 13, A & B). This action is destructive and irreversible, and thus gives experience designers an interesting physical-digital primitive to play with. This is much like instructions to tear cards in half (destroying them) in many legacy-style board games.

4.3.5 Fill-In. Another way to use ink or paint is to alter the payload of a tag by coloring pixels in manually (Figure 13, C & D). If both white and black ink is available, changes are reversible, but if an experience only includes one color, the change is permanent. Such a tag can be used for something as simple as a checkbox, all the way to expressing a complex inventory.

4.3.6 Peel-Off. Another reveal/conceal method is peeling (or pasting) a layer of paper to remove or reveal a marker (Figure 13, E–G). In this way, the removal of a marker from an area or element could trigger an action, as could the appearance of one. If the adhesive layer is repositionable (like post-it notes), the action is bi-directional. However, it may be advantageous in some experiences to use a strong glue such that it is not possible to remove the tag without destruction.

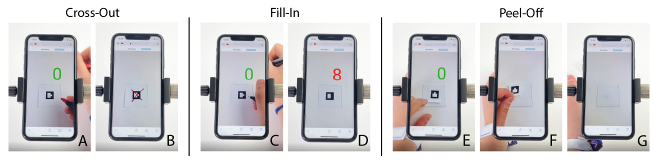


Figure 13: An example of cross-out (A & B), fill-in (C & D), and peel-off mechanisms (E–G).

4.3.7 Rip. When a marker is physically ripped into two or more pieces, it is no longer detectable (Figure 14, A–C). This manipulation is useful for applications that involve opening and completion.

4.3.8 Punch. Finally, by poking a hole in a marker that lies on a soft surface or above a hole, it can be rendered undetectable (Figure 14, D–F). This mechanism is also useful for experiences that involve completion of tasks.

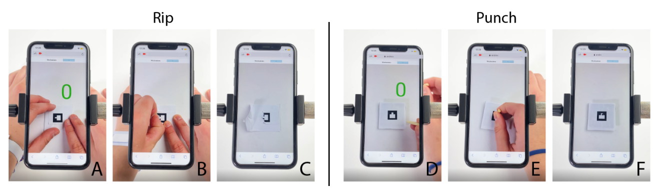


Figure 14: An example of ripping (A–C) and punching mechanisms (D–F).

4.4 Multi-State DynaTags

For illustrative purposes, we chose to present simple bi-state examples of DynaTags to easily convey their operation. The only exception to this was our multi-axis slider (section 4.1.2), which is unique in that operates in two axes. That said, many of our mechanisms can be trivially extended to multiple states. For example, single sliders (section 4.1.1) can have multiple windows to express many states, as we show in our example interactive textbook about the water cycle (Figure 18A, see also Video Figure). For window-based methods, the number of windows can express 2^n payload states. Likewise, abutting-type mechanisms, such as our dial (section 4.2.3) can have many different partial payloads printed on the dial itself.

5 READER SOFTWARE

Our proof-of-concept implementation is written in JavaScript and utilizes the A-Frame [1] and AR.js libraries [4], which include functionality to read AruCo markers. For demos incorporating sound, we use the Tone.js library [36]. These software packages allow our core reader code and demo applications to be cross-platform (iOS and Android, as well as desktop operating systems) and easily incorporate rich multimedia, such as animated graphics and sound.

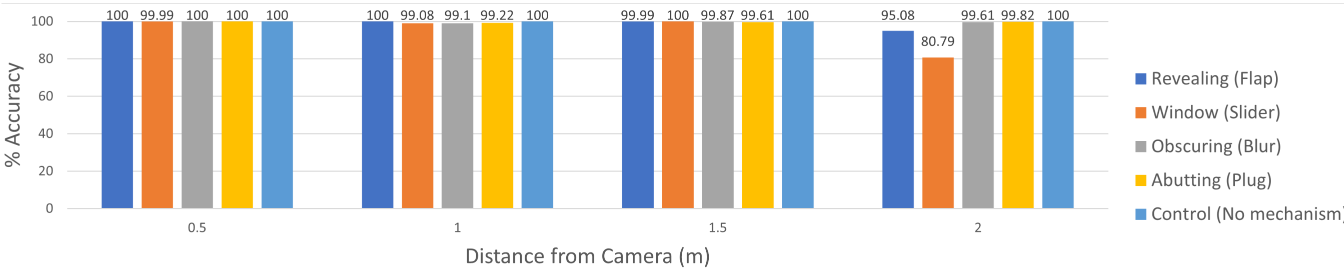


Figure 15: Recognition accuracy of DynaTag mechanisms across four distances in comparison to a standard (control) marker.

We also created an example pass-through augmented reality (AR) application using the same packages. To avoid read errors while a tag is being physically actuated by a user (i.e., intermediate states), we utilize a time-to-life threshold (i.e., the same data payload must be detected n times in a row before it is triggered as being present). In cases where a user leaves a tag partially actuated, the tag can oscillate between states across frames. However, our requirement for multiple sequential frames to detect the same payload effectively suppresses this in the end user experience. We also note that we use 3x3 or 4x4 square binary ArUCo markers [13, 30] in all of the examples presented in this paper. These can represent a maximum of 64 and 8192 unique IDs respectively. Of course, marker schemes with larger payloads are possible too. We use the ArUCo square binary format as a proof of concept.

6 OPEN SOURCE

To allow others to easily utilize, replicate, experiment and extend our technique, we have open-sourced the DynaTags design files and reader software at http://github.com/FIGLAB/DynaTags. On this page, you will also find a Getting Started guide, providing several no-code / low-code examples.

7 EVALUATION

Fiducial markers are a well established technology. However, in this work, there were three key questions we wished to answer. 1) Do our DynaTag mechanisms impair recognition when using a standard marker reader API? 2) Can DynaTags work across typical use ranges? 3) Given these mechanisms are made from paper, can they survive thousands of actuation cycles (e.g., tearing, ink rubbing off, crinkling)?

7.1 Procedure

We selected representative mechanisms for four dynamic payload categories. For revealing, window, obscuring and abutting, we selected flap, slider, blur and plug mechanisms respectively. We also included a fifth, non-mechanical, standard marker as a control. Tags were 17x17mm in size, which we placed in a typical lit office at varying distances (0.5, 1.0, 1.5, and 2.0 meters) from a HD webcam connected over USB to a laptop running our reader software. To mechanically actuate our DynaTags, an Arduino-controlled SG-90 servo motor with a 2cm arm was used, which was controlled by our laptop software. Each trial consisted of actuating a DynaTag to one of its two states, then recognizing any ArUCo Markers in the scene, and comparing it to what ID should be present. If the ID did not match, or no fiducial markers were found, the trial was counted as a failure. If the expected ID was detected, the trial was recorded as a success. In the case of our non-mechanical control tag, no actuation occurred and one ID was present throughout. Our software ran 10,000 trials per tag type, per distance, alternating state every one second, resulting in a total of 200,000 trials.

7.2 Accuracy

The main result can be seen in Figure 15. In short, DynaTags perform slightly worse than a standard, non-mechanical tag, which achieved 100% recognition rates at all tested distances. Meanwhile, the mean recognition rate of the four DynaTag categories up to a distance of 1.5m was 99.79%. The few errors that did occur were due to poor or incomplete alignment of tag elements during actuation, which is inevitable in a mechanical system, especially one made of paper. At 2.0m, mean accuracy of the four DynaTag types drops to 93.83%, though most of this error comes from a significant drop in accuracy of the window mechanism specifically, which we suspect was due to a small gap that formed between the layers, introducing some parallax causing misreads. This experiment covered typical range of use, but we note that detection range could be further extended with larger tags or a higher-resolution webcam.

7.3 Durability

With respect to DynaTag durability, none of our DynaTags showed visible degradation by the end of the experiment and there was no downward trend in accuracy over time. As a result, we decided to continue mechanically actuating our DynaTags to see if there was a failure point. In total, we ran an additional 500K actuation cycles across our four test mechanisms, after which we stopped the experiment (chiefly due to our servo motors failing, not our DynaTags). Again, there was no significant degradation of the DynaTags, underscoring the surprising robustness of paper, and they continued to be recognized correctly by our software.

8 EXAMPLE USES

To underscore the generalizability and utility of DynaTags, we created a series of functional example applications. These demos incorporate 12 of our 23 DynaTag mechanisms. Please also refer to the Video Figure.

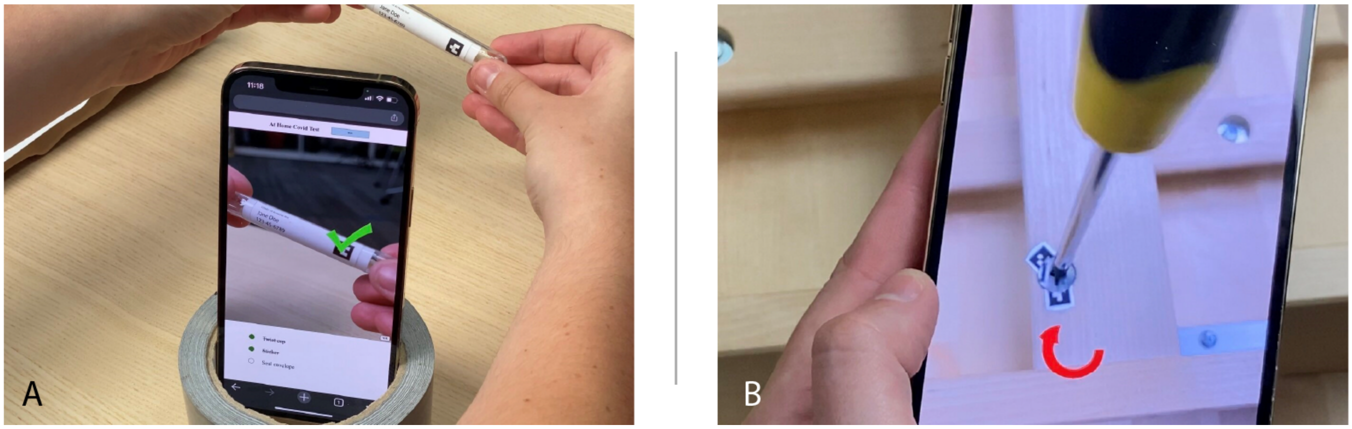


Figure 16: A) A COVID-19 sample collection kit with a companion augmented reality guidance app. The kit materials utilize abutting dial, stacking, and window flap mechanisms. B) A augmented reality, guided furniture assembly experience, using punching, abutting dial, and presence mechanisms.

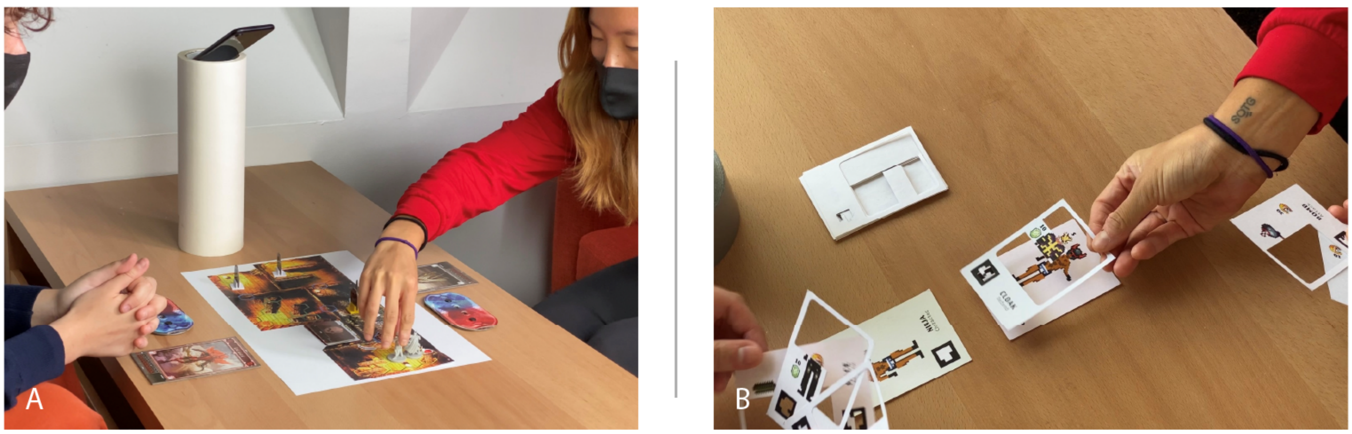


Figure 17: A) An augmented board game, featuring abutting flap, flap reveal, and peel-off mechanisms. A phone held in an accessory stand augments the experience with sounds and storytelling. B) An augmented card game, utilizing stacking DynaTags mechanisms. In this example, the phone keeps track of score so that players can focus on the game.

8.1 Guidance Apps

Augmented reality has long been championed as a powerful modality for in-situ guidance, with canonical tasks such as fixing an engine. With DynaTags, we can bring structured and sequential guidance to more mundane tasks with a high degree of robustness.

As one example, we augmented a COVID-19 sample collection kit (Figure 16A; see Video Figure for full sequence). These kits usually have a number of written steps that must be carefully followed to ensure successful collection and processing. To aid in this task, we instrumented the paperwork and materials with DynaTags and created a companion Augmented Reality (AR) app. For instance, after collecting a specimen with the swab, the test tube must be properly capped with the swab inside. To detect this, we use half of a marker that rotationally abuts to complete a full marker. When the cap is fully secured, a success sound will play from the phone and an AR checkmark will hover over the cap of the test tube. Next, the patient must peel an identifying sticker from the instruction sheet and affix it to the test tube. The marker on the sticker has a pixel window that shows through to a black background on the instruction sheet. However, this pixel becomes white when affixed correctly to the test tube, triggering another success chime and checkmark. Finally, the patient places their test tube in the envelope and seals the envelope; a marker with a window on the flap of the envelope allows for detection of the envelope being sealed, triggering a final success sound.

As another demo assembly task, we augmented a piece of Ikea furniture with DynaTags (Figure 16B; see Video Figure for full sequence). Classic screw holes are covered with sticker tags, which are visualized in a companion AR app as screw insertion points. Inserting the screw destroys the tag, and the software tracks the user's progress. Upon each destruction, the app gives a verbal instruction of how many screws are left in this stage of assembly. For cam locks, a rotational abutting mechanism tracks the progress. If it is not screwed in sufficiently, a curved arrow is displayed in AR

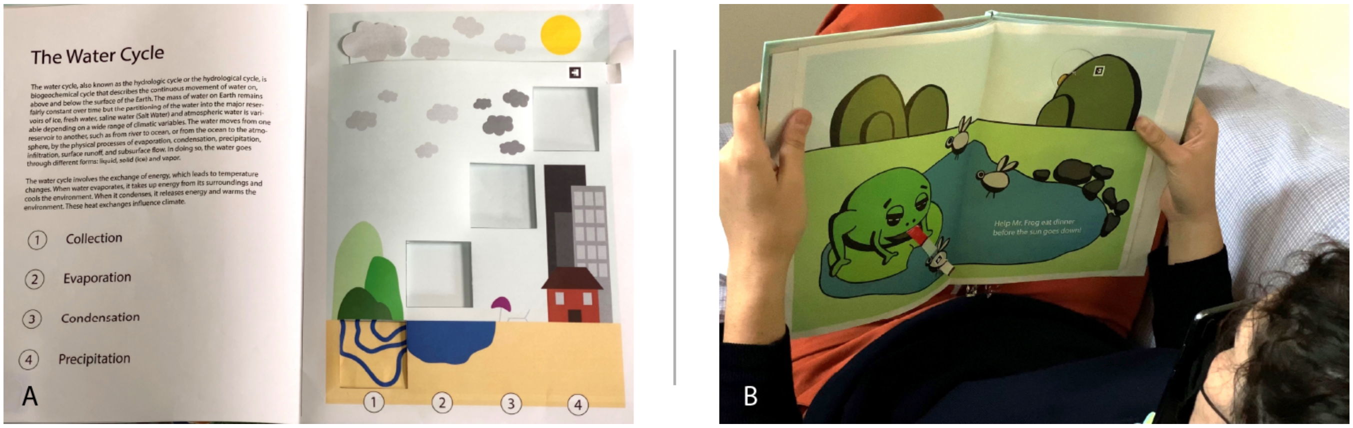


Figure 18: A) An augmented textbook, featuring a multi-input slider mechanism. A phone (not seen in the photo) provides ambient sound and lighting effects. B) An augmented kids' book, featuring elastic slider and window dial mechanisms. A phone resting on the user's chest enhances the book with sounds and lighting.

and the app will instruct the user to keep twisting. Once secured, the app will show a checkmark and give a word of praise to the user. In a later step, to ensure the user does not forget to add dowels between two pieces, we use presence DynaTags.

8.2 Playful Apps

Card and board games continue to be popular in the digital era. The simple fact is that people enjoy coming together to play games where the focus is not on a computer screen. However, DynaTags could augment these analog games, making gameplay easier or more immersive (e.g., keeping score, reading narration), but without getting in the way. We envision a phone running DynaTag-aware software being placed into an accessory stand, looking down at the play surface.

As one example, we created a DynaTag-augmented game loosely based on Gloomhaven (Figure 17A; see Video Figure for full sequence). In the conventional game, reaching a door tile sometimes triggers narration and rule changes, which must be read from a companion book. In our version, a door is a paper flap, which upon being opened reveals a tag that triggers the phone to read aloud narration. An equivalent mechanism is used for treasure chests — in the conventional game, users simply look up a corresponding treasure number (e.g., #14) once they reach the tile. In our version, the chest is physically opened, and the now-visible tag triggers narration, and could even randomly generate loot. Departing from the original game, we also created items that could be picked up or consumed. For example, picking up a sword or eating a plate of food requires peeling a sticker from the game board, removing the item whilst simultaneously revealing a tag, again triggering narration or instructions. Across all of these example interactions, the phone is also used to render coordinated sound effects (e.g., a door creaking open).

We also created a card game loosely based on War (Figure 17B; see Video Figure for full sequence). Players select starting characters with different base attributes (health, attack and shield). They then draw a hand of cards from a common deck, which contains items for the torso (e.g., armor, cloaks) and hands (e.g., potions, swords, shields), which modify the base attributes. Cards contain window-based DynaTags, such that a smartphone can automatically track what items are currently played on top of each player. Each turn consists of players selecting an item to play from their hand, both players simultaneously revealing their choices, placing the revealed card onto their character, and the two characters fighting one round of battle. The smartphone plays appropriate sound effects given the played items (e.g., sword against metal armor) and then announces the winner and current health levels (the first player to reach zero loses). The turn ends with both players drawing a replacement card to keep their hand size at four.

8.3 Books

DynaTags could augment reading experiences to improve both the quality of learning and the independence of the learner. Similar to our example playful apps, we envision a phone held in a stand or in a sleeve of a special accessory pillow. From this vantage point, it can augment a reading experience by adding sound and other effects, such as vibration and ambient lighting. Importantly, the phone is not the focus (or ideally not seen at all), and instead supplements the reading experience.

As one example, we designed a page of an interactive textbook about the water cycle (Figure 18A; see Video Figure for full sequence). The student can read about the subject matter on the left page, and an interactive diagram on the right page helps to better contextualize what they have read. By sliding a tab, the reader can move through the stages of the water cycle, and the phone tracks the state. In the collection phase, the phone's display glows, casting various shades of yellow on the paper, while also playing daytime sounds. In the evaporation phase, the sounds shift to waves at a beach. In the condensation phase, the screen darkens to a shifting blue-grey animation, and the sound changes to a strong wind. Finally, in the precipitation phase, the screen flashes between white and black, briefly illuminating the page to emulate lightening, and thunderstorm sounds are played.

As another example, we designed a page of an interactive children's book (Figure 18B; see Video Figure for full sequence). The aim

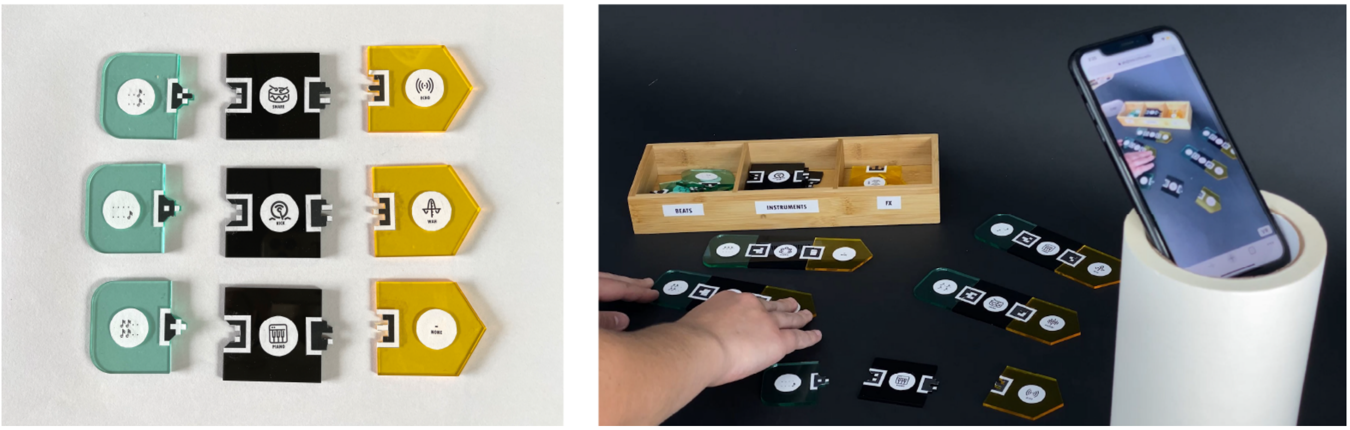


Figure 19: A music synthesizer application, with tangibles representing beat patterns, instruments, and acoustic effects. Different elements can be connected together to create music, with DynaTag abutting plug mechanisms read by an accompanying phone.

of the activity is to feed the frog dinner (flies) before he goes to bed. In one corner of the book, the reader controls the time of day with a window dial mechanism. The page begins at daytime, symbolized by the sun on the dial, which triggers peaceful daytime background music and a bright yellow glow from the phone's screen, the light of which is cast onto the page. The reader continues to engage — by tugging on the frog's tongue (an elastic slider), it "eats" a fly, and a gulping sound effect is played. The reader then turns the dial through the rest of the day. When the reader turns the dial to sunset, the moon and sun become partially visible, a sound effect of a child saying "Bedtime!" is triggered, and the phone's screen turns to pink to simulate sunset. When the dial finally reaches night, an ambient forest soundtrack is played (crickets, wind, rustling leaves, fire crackling, etc.) and the phone's screen casts a violet color.

8.4 Creative Apps

One of the main purposes of the smartphone acting as an environmental accessory and not a screen is to augment existing tangible interactions with audiovisual effects, as we showed in our previous example applications. However, we also explored how the phone could be used to create new audiovisual experiences.

As one example, we designed a tangible music synthesizer (Figure 19; see Video Figure for full sequence), inspired by seminal table-based systems like reacTIVision [22]. Modular blocks can be connected in sequences to create looping sounds. These sequences are created from three-block groupings: a beat pattern (e.g., beats on fours), instrument (e.g., drums, saxophone), and acoustic effect (e.g., reverb, synth). Blocks have marker subsections on their plug-areas that specify their functionality. A complete sequence of three blocks will inherently display two complete markers, detected by an overhead phone. A song can be created by assembling many sequences. Our software uses a simple distance heuristic to determine what blocks are connected together when many sequences are present in the scene. We note that tangibles such as ours are ideal for collaborative multi-user scenarios, as opposed to clustering around a small tablet.

9 LIMITATIONS

In this project, we emphasized the do-it-yourself nature of these paper mechanisms. Perhaps the biggest reason for the success and ubiquity of contemporary fiducial tags has been their print-at-home and easy-to-deploy nature. While DynaTags still only require basic craft tools (e.g., scissors, tape, printer), we acknowledge that our technique could still be intimidating for those who are less experienced with physical prototyping. However, DynaTags are still orders of magnitude faster and easier than e.g., coding an app using the ARKit or Vuforia SDKs. For this reason, we believe we were successful in creating a middle ground approach: modest extra deployment complexity for a significantly richer design space. This is typified by our example applications, most of which are simply not possible with contemporary tagging approaches.

Another limitation of our approach is compatibility with fiducual marker standards. For instance, QR Codes include error correction bits that make simple, single-pixel state changes impossible. Instead, groups of pixels would have to be updated, which has already been shown to be possible in [15]. However, this would increase fabrication complexity of DynaTags. We also note that our proof-of-concept designs used the 3x3 and 4x4 square binary format [13], which offers a maximum of 64 and 8192 unique IDs, respectively. The largest ArUco format is 7x7, offering 2^{45} possible IDs, though tags are most often formed into a dictionaries with a minimum hamming distance to increase robustness to errors (e.g., a hamming distance of two still leaves trillions of unique IDs).

Finally, as a computer-vision-driven approach, our tags suffer from standard occlusion and lighting issues (i.e., not unique to our approach). More unique is that our mechanisms can degrade over time from being handled and mishandled (in the same way as e.g., popup books). For this reason, they are ill-suited for high-traffic contexts, unless the owner is willing to refresh them often. However, some of our mechanisms can be made robust, such as the plugs used in our music synthesizer example, which were cut from acrylic. Several of our other demos were integrated into existing paper items, such as books and board games, where other elements must be treated with equivalent care.

10 CONCLUSION

In this paper, we have presented DynaTags, an extension to conventional fiducial markers that affords an extra and interesting degree of interactivity. Although simple, our set of mechanisms can infuse dynamic, digital capabilities into otherwise static experiences. For example, board games and paper books can be augmented with indirect lighting and sound effects. Importantly, because we use proven, high-contrast fiducial marker schemes (and not preregistered image-based computer vision matching and tracking), our approach is very robust and can use standard marker reader apps and APIs. We describe an extensive list of DynaTag mechanisms we created as part of our explorations. We conclude the paper with a series of example use cases incorporating our mechanisms, which have the ability to augment reality without necessarily consuming the user's entire attention. In the future, we believe online DynaTag generators and downloadable kits that are "print-cut-and-play" could further lower the barrier to entry. Such a kit could contain instructions on how to construct the DynaTag mechanisms, as well as a library of functions that allows users to create interactive experiences with DynaTags without necessarily having a deep understanding of AR.js.

ACKNOWLEDGMENTS

We thank Cathy Fang for her early contributions to this project.

REFERENCES

[1] A-frame 2022. A-frame AR Development. https://aframe.io/blog/aframe-v1.1.0.

[2] Apple 2022. ARKit Documentation. https://developer.apple.com/documentation/arkit/.

[3] Apple ARImageAnchor 2022. ARImageAnchor Class, ARKit Documentation. https://developer.apple.com/documentation/arkit/arimageanchor.

[4] AR.js 2022. AR.js Documentation - Augmented Reality on the Web. https://ar-js-org.github.io/AR.js-Docs.

[5] Tom Bartindale and Chris Harrison. 2009. Stacks on the Surface: Resolving Physical Order Using Fiducial Markers with Structured Transparency *(ITS '09)*. Association for Computing Machinery, New York, NY, USA, 57–60. https://doi.org/10.1145/1731903.1731916

[6] Brandly. 2019. QR Code Business Cards: Everything You Need to Know. https://www.brandly.com/blog/everything-you-need-to-know-about-business-cards-and-qr-codes.

[7] Zekun Chang, Tung D Ta, Koya Narumi, Heeju Kim, Fuminori Okuya, Dongchi Li, Kunihiro Kato, Jie Qi, Yoshinobu Miyamoto, Kazuya Saito, et al. 2020. Kirigami Haptic Swatches: Design Methods for Cut-and-Fold Haptic Feedback Mechanisms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[8] Gary PT Choi, Levi H Dudte, and L Mahadevan. 2019. Programming shape using kirigami tessellations. *Nature materials* 18, 9 (2019), 999–1004.

[9] QR Code. 2020. Automatic identification and data capture techniques. Bar code symbology. https://www.iso.org/standard/30789.html. Accessed: 2020-09-30.

[10] Enrico Costanza and Jeffrey Huang. 2009. Designable Visual Markers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1879–1888. https://doi.org/10.1145/1518701.1518990

[11] Kurtin Diggs. 2022. A new spin on multiple choice questions. https://www.pinterest.com/pin/141441244518620355.

[12] Adam Drogemuller, James Walsh, Ross T. Smith, Matt Adcock, and Bruce H Thomas. 2021. Turning Everyday Objects into Passive Tangible Controllers. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Salzburg, Austria) *(TEI '21)*. Association for Computing Machinery, New York, NY, USA, Article 54, 4 pages. https://doi.org/10.1145/3430524.3442460

[13] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Manuel Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47 (06 2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[14] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005

[15] Nenad Gligoric, Srdjan Krco, Liisa Hakola, Kaisa Vehmas, Suparna De, Klaus Moessner, Kristoffer Jansson, Ingmar Polenz, and Rob van Kranenburg. 2019. SmartTags: IoT Product Passport for Circular Economy Based on Printed Sensors and Unique Item-Level Identifiers. *Sensors* 19, 3 (2019). https://doi.org/10.3390/s19030586

[16] Google 2022. ARCore Documentation. https://developers.google.com/ar/develop.

[17] Jens Emil Grønbæk, Majken Kirkegaard Rasmussen, Kim Halskov, and Marianne Graves Petersen. 2020. *KirigamiTable: Designing for Proxemic Transitions with a Shape-Changing Tabletop*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3313831.3376834

[18] Jeremy Gummeson, James Mccann, Chouchang (JACK) Yang, Damith Ranasinghe, Scott Hudson, and Alanson Sample. 2017. RFID Light Bulb: Enabling Ubiquitous Deployment of Interactive RFID Systems. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 12 (jun 2017), 16 pages. https://doi.org/10.1145/3090077

[19] Liisa Hakola, Kaisa Vehmas, and Maria Smolander. 2021. Functional inks and indicators for Smart Tag based intelligent packaging applications. *Journal of Applied Packaging Research* 13, 2 (2021), 3.

[20] Chris Harrison, Robert Xiao, and Scott Hudson. 2012. *Acoustic Barcodes: Passive, Durable and Inexpensive Notched Identification Tags*. Association for Computing Machinery, New York, NY, USA, 563–568. https://doi.org/10.1145/2380116.2380187

[21] Shinichi Higashino, Sakiko Nishi, and Ryuuki Sakamoto. 2016. ARTTag: Aesthetic Fiducial Markers Based on Circle Pairs. In *ACM SIGGRAPH 2016 Posters* (Anaheim, California) *(SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 38, 2 pages. https://doi.org/10.1145/2945078.2945116

[22] Martin Kaltenbrunner and Ross Bencina. 2007. ReacTIVision: A Computer-Vision Framework for Table-Based Tangible Interaction. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana) *(TEI '07)*. Association for Computing Machinery, New York, NY, USA, 69–74. https://doi.org/10.1145/1226969.1226983

[23] Hirokazu Kato and Mark Billinghurst. 1999. Marker tracking and HMD calibration for a video-based augmentedreality conferencing system. *The 2nd International Workshop on Augmented Reality (IWAR 99)*, 85–94. https://doi.org/10.1109/IWAR.1999.803809

[24] Brax Landing. 2022. Brax Restaurant Menu. https://www.braxrestaurant.com.

[25] Beth M. Lange, Mark A. Jones, and James L. Meyers. 1998. Insight Lab: An Immersive Team Environment Linking Paper, Displays, and Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Los Angeles, California, USA) *(CHI '98)*. ACM Press/Addison-Wesley Publishing Co., USA, 550–557. https://doi.org/10.1145/274644.274718

[26] Zhengxiong Li, Baicheng Chen, Zhuolin Yang, Huining Li, Chenhan Xu, Xingyu Chen, Kun Wang, and Wenyao Xu. 2019. FerroTag: A Paper-Based MmWave-Scannable Tagging Infrastructure. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* (New York, New York) *(SenSys '19)*. Association for Computing Machinery, New York, NY, USA, 324–337. https://doi.org/10.1145/3356250.3360019

[27] Rong-Hao Liang, Meng-Ju Hsieh, Jheng-You Ke, Jr-Ling Guo, and Bing-Yu Chen. 2018. RFIMatch: Distributed Batteryless Near-Field Identification Using RFID-Tagged Magnet-Biased Reed Switches *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 473–483. https://doi.org/10.1145/3242587.3242620

[28] Henrique Teles Maia, Dingzeyu Li, Yuan Yang, and Changxi Zheng. 2019. Layer-Code: Optical Barcodes for 3D Printed Shapes. *ACM Trans. Graph.* 38, 4, Article 112 (jul 2019), 14 pages. https://doi.org/10.1145/3306346.3322960

[29] Jun Rekimoto and Yuji Ayatsuka. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments* (Elsinore, Denmark) *(DARE '00)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/354666.354667

[30] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. 2018. Speeded Up Detection of Squared Fiducial Markers. *Image and Vision Computing* 76 (06 2018). https://doi.org/10.1016/j.imavis.2018.05.004

[31] STMicroelectronics 2022. SensorTile Development Kit. https://www.st.com/en/evaluation-tools/steval-stlkt01v1.html.

[32] Yichao Tang, Gaojian Lin, Shu Yang, Yun Kyu Yi, Randall D Kamien, and Jie Yin. 2017. Programmable Kiri-Kirigami Metamaterials. *Advanced Materials* 29, 10 (2017), 1604262.

[33] Md. Farhan Tasnim Oshim, Julian Killingback, Dave Follette, Huaishu Peng, and Tauhidur Rahman. 2020. *MechanoBeat: Monitoring Interactions with Everyday Objects Using 3D Printed Harmonic Oscillators and Ultra-Wideband Radar*. Association for Computing Machinery, New York, NY, USA, 430–444. https://doi.org/10.1145/3379337.3415902

[34] Texas Instruments 2022. SensorTag Development Kit. https://www.ti.com/tool/CC2541DK-SENSOR.

[35] Thefunnyvideosable. 2013. ASMR Animal Crossing New Leaf Magnets QR Design Book. https://www.youtube.com/watch?v=iWLD8R7VVN0.

[36] Tone.js 2022. A Web Audio Framework. https://tonejs.github.io.

[37] Unity 2022. AR tracked image manager, Unity AR Scripting Manual. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/tracked-image-manager.html.

[38] VictoriaMaxwell. 2022. Multi QR Code Business Social Media Sign | Salon Sign | Beauty Sign | Hairdressers Beautician Sign. https://www.etsy.com/listing/980786229/multi-qr-code-business-social-media-sign.

[39] Vuforia 2022. Vuforia Developer Library. https://library.vuforia.com/.

[40] Kristin Williams. 2020. IoT Stickers: Enabling Lightweight Modification of Everyday Objects *(UIST '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 183–188. https://doi.org/10.1145/3379350.3415807

[41] Xiaoying Yang and Yang Zhang. 2021. *CubeSense: Wireless, Battery-Free Interactivity through Low-Cost Corner Reflector Mechanisms.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411763.3451599

[42] Hui-Shyong Yeo, Ryosuke Minami, Kirill Rodriguez, George Shaker, and Aaron Quigley. 2018. Exploring Tangible Interactions with Radar Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 200 (dec 2018), 25 pages. https://doi.org/10.1145/3287078

[43] Dingtian Zhang, Jung Wook Park, Yang Zhang, Yuhui Zhao, Yiyang Wang, Yunzhi Li, Tanvi Bhagwat, Wen-Fang Chou, Xiaojia Jia, Bernard Kippelen, Canek Fuentes-Hernandez, Thad Starner, and Gregory D. Abowd. 2020. OptoSense: Towards Ubiquitous Self-Powered Ambient Light Sensing Surfaces. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3, Article 103 (sep 2020), 27 pages. https://doi.org/10.1145/3411826

[44] Yongtai Zhang, Shihong Deng, Zhihong Liu, and Yongtao Wang. 2015. Aesthetic QR Codes Based on Two-Stage Image Blending. In *MultiMedia Modeling*, Xiangjian He, Suhuai Luo, Dacheng Tao, Changsheng Xu, Jie Yang, and Muhammad Abul Hasan (Eds.). Springer International Publishing, Cham, 183–194.

[45] Yang Zhang, Yasha Iravantchi, Haojian Jin, Swarun Kumar, and Chris Harrison. 2019. Sozu: Self-Powered Radio Tags for Building-Scale Activity Sensing. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 973–985. https://doi.org/10.1145/3332165.3347952

[46] Yang Zhang, Gierad Laput, and Chris Harrison. 2018. Vibrosight: Long-range vibrometry for smart environment sensing. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 225–236.

[47] Clement Zheng, Peter Gyory, and Ellen Yi-Luen Do. 2020. *Tangible Interfaces with Printed Paper Markers.* Association for Computing Machinery, New York, NY, USA, 909–923. https://doi.org/10.1145/3357236.3395578