Anyboard: a Platform for Hybrid Board Games

Simone Mora, Tomas Fagerbekk, Matthias Monnier, Emil Schroeder, and
Monica Divitini

Dept. of Information Technology and Computer Science,
Norwegian University of Science and Technology, Trondheim, Norway
`simone.mora@ntnu.no`, `monica.divitini@idi.ntnu.no`

Abstract. Making hybrid board games that mix the interactivity of
video games with the social impact of board games is challenging. While
the design process needs to take into account elements from the digital
and analog domains, building prototypes requires dealing with diverse
technologies in the field of Tangible Interfaces and Interactive Tabletop
and Surfaces. Anyboard provides theoretical tools to map traditional
board game interaction to the hybrid medium and lightweight technology
tools to facilitate game prototyping. Our platform provides augmented
game pieces that work with traditional cardboards, allowing designers to
easily build collaborative interactive games without requiring engineering
skills.

Keywords: Hybrid Game, Pervasive Game, Design, Prototyping, Plat-
form, Tangible Interface

1 Introduction

Despite the popularity of computer games, board games still represent a signif-
icant part of the market [3]. The reason lies in board games providing a more
human-oriented experience compared to computer games [15]. Playing board
games is indeed a social ritual where families and friends get together; having
low entry-barriers for cost and skills, board games represent a cross-generational
form of entertainment.

Over the last twenty years, research has pointed at combining the interactiv-
ity of computer games with the benefits of board games [15, 13] creating interac-
tive and augmented board games; hereafter referred as *hybrid board games*. For
a review see Haller et al. [8]. Hybrid games have been extensively implemented
using interactive surfaces such as tabletop computers, recently complemented
by novel technologies in the field of *Multi-Display Environments* (MDE) [20]
and *Around-the-Device Interaction* (ADI) [5]. In hybrid games, large interactive
surfaces recreate the social affordances typical of board games, whilst the use
of active or passive playing pieces preserve physical affordances similar to tradi-
tional game pieces. However these technologies pose several constraints in terms
of design space, mobility and cost; and high entry-barriers for developers.

The objective of our research is to lower the barrier for developers to create hybrid games. In [19] we proposed the Interactive-Token approach (ITo) to hybrid games. The approach, grounded in Tangible User Interfaces frameworks [21, 9], is based on the concept of *tokens*, *constraints* and *interaction events*. It proposes the design of hybrid games based on interactive game pieces only, without using interactive surfaces. In the approach, the board is preserved as a low-tech component. Because of not relying on interactive surfaces or external sensing infrastructures, ITo provides an extended design space and mobile alternative to interactive tabletops and surfaces.

In this paper we present *Anyboard*, a platform for creating hybrid board games based on the ITo approach. The platform supports design and prototyping of games. Anyboard design concepts help modelling hybrid game user interaction with technology-augmented game pieces, facilitating the transition between design and implementation. Anyboard software and hardware tools support prototype development with (i) a set of interactive physical game pieces that can be reused across different games, and (ii) a software library to support the coding of games. Our platform aims at reducing the gap between design and prototyping making it easier for designers to use technologies such as sensors and actuators that usually require engineering skills. It should be noted that Anyboard focuses on supporting interaction design rather than more "traditional" game development components such as game engines.

The paper is structured as follows. We first provide a definition of the ITo approach, followed by existing tools for supporting development of hybrid board games. Second we describe the architecture of Anyboard, listing the tools and APIs provided to developers. We evaluate the power of Anyboard illustrating how it could be used to implement common board game mechanics. We conclude highlighting future work.

2 Hybrid Board Games: The ITo Approach

With the dawn of the ubiquitous computing age, board games gained new interest. Advances in interactive surfaces enabled for the development of hybrid versions of board games with or without augmented playing pieces. For example in [2] players use interactive physical iconic or symbolic artifact as their avatars to perform actions in the game. In [11], players can interact on a virtual game board with smart-pens and combine physical and virtual objects to solve puzzle games.

In recent years the large adoption of smartphones and tablets fostered the creation of hybrid games for mobile platforms. To address the limited screen size that impedes co-located play, technologies in the field of Multi-Display Environments (MDE) [20] and Around the Device Interaction (ADI) [5] were adopted. In hybrid games such as *Capture the flag* [20], gesturing and juxtaposition of private smartphone screens onto a shared screen reveal hidden layers and provide player-custom perspectives into the game. ADI technologies have demonstrated the development of augmented game pieces capable to control the behavior of

a smartphone game by manipulation over or around the screen. *Airsteroids* [6], a spaceship game, demonstrates how the game space can extend across several tablets brought in by players and the use of augmented cards to control spaceships' heading.

These examples show how hybrid games usually rely on a mix of private and shared interactive surfaces as a primary mean to convey interactivity, with augmented game pieces as helpers. This approach poses several limitation in terms of mobility and cost and it requires engineering skills to orchestrate the diverse technologies; limiting the adoption of hybrid games outside research labs.

In the ITo approach, game pieces (tokens) with embedded sensors and actuators are the primary means to add interactivity to board games, while the cardboard is preserved as passive. In this way the same tokens can be used to play multiple levels or games just printing a new cardboard with consumer-grade devices. Technology is used to enhance *control* and *representation* roles of game pieces. Tokens can *control* the state of a game by capturing players' interaction events. For example an accelerometer embedded in a dice can sense the result of a dice throw, and notify a computer game engine in a way that is transparent to the player. At the same time, tokens can *represent* the state of a game by producing *digital feedbacks*; e.g. via graphics, haptics or sound via embedded displays and actuators.

Distributing interactivity across multiple components opens for a wider space of possibility in designing game experiences. Tokens are active playing pieces and they can influence the state of a game not only when they sit on an interactive surface, but also when they are manipulated over and around it. Finally, because of relying on embedded technology not requiring external hardware, ITo offers a more mobile and cost effective alternative to tabletop and ADI technologies. For more information about the ITo approach see [18].

2.1 Challenges to the development of hybrid games

The development of hybrid games based on the ITo approach poses several challenges, mixing issues typical of both board and computer games [12]. These challenges are often common to many pervasive [13] games. From a design point of view there is a lack of theoretical tools to model players' interaction as well as abstractions to describe common elements of games. In particular, the complexity of dealing with different low-level hardware and software technologies, for example to detect manipulation of augmented game pieces, makes game development hard to non-expert, creating a gap between design and prototyping tasks [16]. Further, besides game pieces and mechanics can be generic, the implementation is often so specific that cannot be re-used to develop similar games [12].

Software frameworks and game engines for development of video-games can partially help development, however special tools are often needed to exploit at full the design space. Tangible User Interfaces prototyping toolkits, such as *Arduino* [17] and *Phidgets* [7] can be used to implement interactive pieces, although they don't specifically address game development. Conversely, toolkits to

support hybrid game development often focus on tabletop or tablet computing platforms; for example supporting object recognition and tracking as for game pieces [4], dealing with multiple screens [14], designing terrain composition and dealing with input management [12], or facilitates the development of games that feature playing pieces on and around a tablet screen [1].

Only a few toolkits [16, 10] explicitly support the development of interactive game pieces required by the ITo approach, and only as complement to tabletop interactive surfaces. An integrated set of design and prototyping tool to support the development of ITo hybrid games could not be found.

3 The Anyboard platform

Building on analysis of development tools available and our experience in building hybrid games [19] we created the Anyboard platform (Figure 1). AnyBoard implements the ITo approach providing:

– *Design entities* to model interaction styles and provide a common terminology to describe games both in design and implementation phases
– *Interactive game pieces and cardboard templates* to support the construction of the game hardware (Figure 2)
– *AnyboardJS library* to facilitate coding of games by abstracting the complexity of dealing with hardware and low-level programming
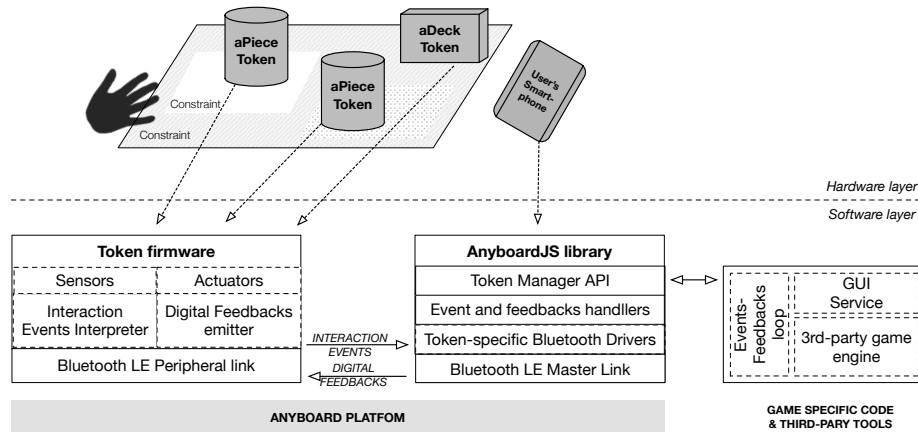

Fig. 1. Anyboard platform overview

3.1 Design entities

Design entities derived from the ITo approach [19], listed below, constitute the foundation of a language to model interaction styles with Anyboard games. They

have been mapped to traditional board games elements in Table 1, to help bridging the gap between game design and development.

Table 1. Mapping between Anyboard and board game concepts

Anyboard design entities	Traditional Board Game elements
Tokens	Game pieces
Constraints	Board tiles or sockets
Interaction Events	Player's actions

Tokens are technology-augmented artifacts which resemble traditional game pieces; ranging from elements of chance (e.g. an augmented dice in *Backgammon* or RFID-enabled cards in *Monopoly*) to game pieces, e.g. pawns augmented with an LCD displaying the player's rank in the game.

Constraints are either visual or physical confining regions in the board space. The association or dissociation of a token within a constraint can be mapped to digital operations to activate game dynamics. Examples of constraints are checks for *Chess* pieces and territories in *Risk*.

Interaction events are player-triggered manipulations of tokens, that modify the (digital and physical) state of a game. We define three types of events:

- *token event (TE)* - the manipulation of a single token on, over around the board; e.g. the action of rolling a dice or drawing a card
- *token-constraint event (TCE)* - the operation of building transient token-constraint associations by adding or removing tokens to a constrained region of the board; e.g. moving army pieces beyond a territory line as an attack action
- *token-token event (TTE)* - the operation of building transient token-token adjacency relationships, achieved by moving tokens on the board; e.g. moving a token next to another token to exchange a resource between two players.

An Anyboard game can be therefore defined as a sequence of player-initiated interaction events that modify spatial configurations of tokens with respect to board constraints and other tokens. As a consequence tokens' *intangible* (digital feedbacks) representations are updated.

Sequences of interaction events describe players' actions during the game and are mapped to specific game dynamics implemented in a game engine.

The game engine connects and exchange data with tokens using the AnyboardJS library (Section 3.3) running on the user's smartphone, for example activating specific game mechanics according with interaction events sensed by tokens or triggering digital feedbacks.

Design entities have driven the development of hardware and software tools to assist the making of Anyboard games.
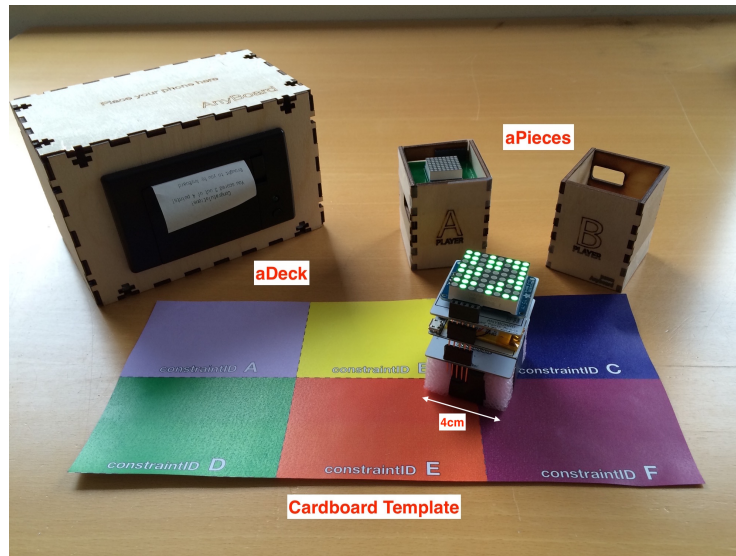
Fig. 2. Interactive game pieces and carboard template prototype

3.2 Interactive game tokens and cardboard template

Anyboard currently prototypes two different types of interactive tokens: *aPiece*, an augmented game piece, and *aDeck*, an augmented card deck (Figure 2). Being two of the most common elements of board game they can be used to implement a wide range of games.

aPieces are a technology-augmented version of game pieces commonly found in most games. They are capable of capturing the set of interaction events and produce digital feedbacks. Table 2 and 3 cluster interaction events and digital feedback supported by typology and provide a player-perspective description of their role. Possible mapping with dynamics commonly found in board games are also exemplified. For example the action of shaking an aPiece can be mapped to the draw of a random number or event.

aPiece has been designed and implemented using custom components. This choice allowed the use of miniaturised components, leading to aPieces with volume and affordances similar to traditional game pieces. aPiece embedds an ARM processor, which runs a custom-designed firmware, and a BTLE (Bluetooth Low Energy) radio transmitter. It implements algorithms to handle sensors data and drive feedback actuators as well as establishing wireless links with the game engine. The core unit is complemented by circuitry for battery charge and firmware upgrade.

TEs (Token Events) are recognised by a 3-axis accelerometer implementing algorithms for extracting features to identify physical manipulation events (e.g. SHAKE, TAP).

Table 2. Interaction events implemented by aPiece

Type	Interaction Event	Description	Sample mapping with game mechanics
TE	SHAKE	aPiece is shaken	Throw a random number
	TILT	aPiece is tilted upside down	Undo a previous action
	TAP	aPiece is tapped on the top side	Increase a resource by one unit
	DOUBLE-TAP	aPiece is double-tapped on the top side	Decrease a resource by one unit
TCE	ENTERS_[cID]	aPiece is moved inside a cID sector of the board	Signal player's placement and movements among
	LEAVES_[cID]	aPiece is moved away from cID sector	different board sectors
TTE	APCHES_[aID2]	aPiece is moved close to another one	Trade a resource between two players
	LEAVES_[aID2]	aPiece is moved away to another one	Break a relationship between two players

TCEs recognition is implemented by assigning and imprinting unique colors to different sectors of a game board (representing visual constraints to token's locations). A color-sensor located on the bottom of aPiece samples the color temperature of the surface the device is lying on, returning an unique color-code which is used as a fingerprint for board constraints, enabling to detect when aPiece is moved between two sectors. TCE detection also makes use of accelerometer data to ensure that color sampling is performed only when the device is steady on a sector, deactivating the sensing routine when aPiece being moved.

TTEs are recognised by computing the distance between two pawns using RSSI (Received Signal Strength) data from the radio transmitter.

Token digital feedbacks are implemented using three different devices: an RGB LED, a 8x8 LED Matrix and a vibration motor.

The aPiece firmware (Figure 1) can be extended using Arduino code. In this way new functions can be added allowing implementing new interaction events and extending the range of feedbacks. Further, the firmware can be used as a template to develop new tokens. The appearance of aPiece can also be customized with handcrafted or 3D-printed models blending the digital feedback with a static iconic or symbolic shape.

aDeck is an interactive version of the card deck commonly found in board games. Instead of holding stacks of cards, aDeck prints out its own cards using a small thermal printer. aDeck provides visual feedbacks as listed in Table 4. Besides having textual and graphical information on card, aDeck can also be configured to print cards with 2D-barcodes that can be recognised by smartphone cameras or dedicated devices, and used in certain games to trigger game

Table 3. Digital feedbacks implemented by aPiece

Type	Feedback (FEED-BACKID)	Description	Sample mapping with game mechanics
Visual	LED_[color]	aPiece lights up in the color defined by [color]	Show the status of a resource
	MATRIX_[text]	aPiece top side display shows the string [text]	Shows player's action point allowance
	MATRIX_[icon]	aPiece top side display shows the icon [icon]	Show the result of a dice roll
Haptic	SHRT_HAPTIC	aPiece produces a short haptic feedback	Signal a player to move to the next turn
	LNG_HATIC	aPiece produces a long haptic feedback	Signal a player an action not allowed

mechanics. This is not yet specifically addressed by our platform but can be easily implemented with third-party tools.

Table 4. Digital feedbacks implemented by aDeck

Type	Feedback	Description	Sample mapping with game mechanics
Visual	CARD[text]	A new card with textual information is printed	Inform the player about an event
	CARD[icon]	A new card with an graphical icon is printed	Represent a resource that the player owns
	CARD[barcode]	A new card with a barcode is printed	Trigger an action when barcode is scanned

aDeck is implemented using a thermal printer capable to print on conventional thermal paper up to 4.8cm length, and an Arduino-compatible board.

Cardboard templates provide support to creating custom cardboards that interact with iPieces. As mentioned above, Anyboard use color temperature sensing to implement constraint regions.

We therefore selected nine colors that provided satisfactory sensor recognition rate across different materials and printing techniques and associate them with unique IDs to be used in game development. A color palette to draw cardboard with custom constraint regions is provided as a template file that can be modified with vector graphic editors (example in Figure 2).

3.3 AnyboardJS

AnyboardJS is a javascript library providing developer-friendly functions to code games (Figure 3). It acts as an interface between the interactive tokens provided by the Anyboard platform and game-specific code developed as standalone or with third-party game engines. AnyboardJS provides two main functions. It establishes wireless links with aPieces and aDeck enabling data exchange using specific drivers for each type of token. It provides a *TokenManager* interface to enable developers to easily write javascript code to handle interaction events from tokens and generate digital feedbacks, to implement game-specific routines. The library is designed to be extensible, support to new interactive tokens can be added by writing driver modules. Although tokens are the primary user interface with a game, it is possible to code a secondary GUI to display extended information and controls on the smartphone screen; e.g. to be used to edit game settings, to display game rules and leader-boards. This functionality can be implemented with external libraries (e.g. jQuery mobile[1]).

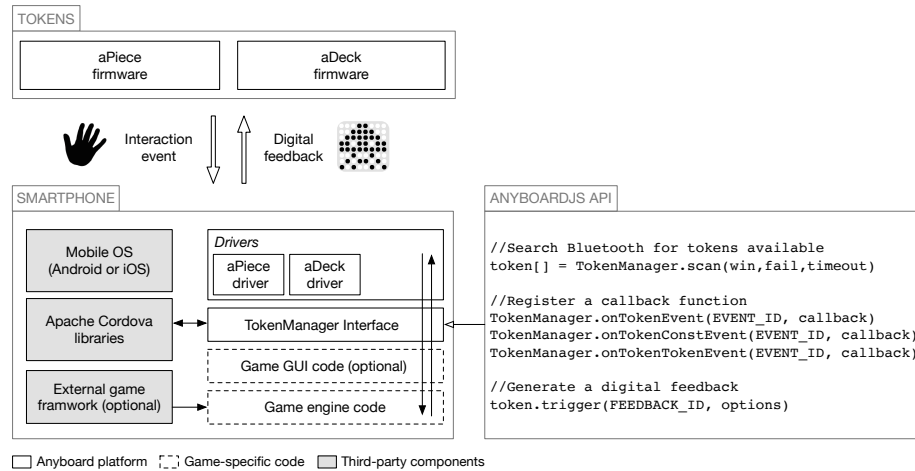A subset of the *TokenManager* interface functions is provided in Figure 3.

Fig. 3. AnyboardJS library internals and API

AnyboardJS is built on top of the *Apache Cordova* framework[2], and it can deployed on a number of mobile platforms. Further the library has been designed to be used with standard javascript code, to implement simple games, but it can also be integrated with third-party game development frameworks such as *UnityMobile*[3] or *Phaser*[4] for building complex games.

---

[1] jQuery mobile - `https://jquerymobile.com/`
[2] Adobe Cordova - `https://cordova.apache.org/`
[3] UnityMobile - `https://unity3d.com/`
[4] Phaser - `http://phaser.io/`

4 Discussion: Making an Anyboard game

Making an hybrid game using Anyboard is a two-steps process. First it requires to model game pieces, rules and mechanics using the provided design entities (Section 3.1). Second it involves prototyping the game using the provided interactive game pieces and the AnyboardJS library. Our platform saves developers from taking care of low-level implementation details, yet leaving freedom for customization.

As a working example, we consider the popular board game mechanic of resource trading, for example transferring the ownership of a property in a game like Monopoly. This mechanic could be implemented in hybrid board games by having two tokens, avatars of players, moved close to each other to symbolize the resource transaction between the two. Digital feedbacks can acknowledge the player with approval or failure (in case of no funds available) with visual or haptic feedbacks.

Implementing such mechanic with current tools would take multiple efforts. First, game designers and developers should agree on common terminology to describe the mechanic and playing pieces involved, because there's usually no such "resource trading" API in game development toolkits. Second, game developers should build the tokens and implement recognition of tokens manipulation and production of digital feedbacks, dealing with diverse hardware and software low-level technologies. Third, the developer should design data transfer protocols and implement interfaces between interactive tokens and game-specific code.

Developing the described mechanic with Anyboard requires only two steps. First, Anyboard design entities offer a choice of opportunities to model the trading mechanic, simplifying transition between game design and implementation. Two *aPieces* tokens can be used as avatars for the two players, the seller and the buyer. When the two tokens are moved close to each other in a *Token-Token Interaction Event*, the resource is exchanged among the two player. As a consequence, tokens' feedbacks can acknowledge the player with a graphic representation. Second, using AnyboardJS, the mechanic can be easily translated into code which handles interaction events and triggers feedbacks (Listing 1.1); acting as an interface with digital model of the game implemented in a game engine.

Listing 1.1. Example of game mechanic implemented with AnyboardJS

```
var trade = function(initToken, respToken, options){
    if (initToken.fundsAvailable())
    {
        transfer_property();
        initToken.trigger(MATRIX, "ICON_OK");
        respToken.trigger(MATRIX, "ICON_OK");
    }
}
TokenManager.onTokenTokenEvent("MOVE_NEXT_TO", trade);
```

This simple example can be further extended to leverage other Anyboard functions. For example, the trading action to be only allowed when both tokens are in a specific constrain sector, detected by a *Token-Constraint event*. Otherwise *aDeck* could be used to issue an "ownership certificate" reporting the name of the new owner and current property value.

Anyboard makes it easy for developers to explore design choices just editing a few lines of code. For example, the digital feedback in the example above, provided in a form of iconic visualization can be instead provided with haptic feedbacks or with color codes, using aPiece's LED. Information provided to the player can mix tangible and intangible representations. For example, static information like players identities or roles can be provided by customizing aPieces with 3D-printing and handcrafting work, instead action points or other dynamic information can be provided using digital feedbacks.

Anyboard has been designed to be extensible, experts can create interactive tokens of their own, with different sensors or feedback capabilities, tinkering with the Arduino-compatible firmware developed for aPieces and by writing *driver modules* for AnyboardJS.

5 Conclusion and Future Work

The Anyboard platform contributes to hybrid board game research with design and prototyping tools to create games based on the ITo approach. Design entities facilitate modelling game mechanics, while interactive tokens supported by a Javascript library shield developers from dealing with hardware complexities and low-level software development. We expect Anyboard to lower the thresholds of technical competences needed for game development.

Anyboard source code and hardware schematics are open source[5]. Future work will have multiple efforts. On one side we aim at better understanding the expressive power of Anyboard design entities by systematically modelling popular game mechanics, adding to the mapping already provided in Tables 2, 3 and 4. On the other side we plan to run game design workshop to evaluate developer friendliness and performances of the tools. Finally there are also plans to build new type of interactive tokens to further explore the space of opportunities provided by the ITo approach.

Acknowledgements We thank Dr. Ines Di Loreto for her contribution to the ITo approach formalisation and her feedbacks on Anyboard.

References

1. Avrahami, D., Wobbrock, J.O., Izadi, S.: Portico: tangible interaction on and around a tablet. In: Proceedings of UIST, pp. 347–356. ACM (2011)

---

[5] Anyboard GitHub repository is available at `http://anyboardgames.co`

2. Bakker, S., Vorstenbosch, D., Van Den Hoven, E., Hollemans, G., Bergman, T.: Weathergods: tangible interaction in a digital tabletop game. In: Proceedings of TEI, pp. 151–152. ACM (2007)
3. Costikyan, G.: Don′t be a vidiot: What computer game designers can learn from non-electronic games. Game Developers Conference 26 (1998)
4. Dietz, P., Leigh, D.: DiamondTouch: a multi-user touch technology. In: Proceedings of UIST, pp. 219–226. ACM (2001)
5. Garcia-Sanjuan, F., Catala, A., Fitzpatrick, G., Jaen, J.: Around-Device Interactions: A Usability Study of Frame Markers in Acquisition Tasks. In: Proceedings of INTERACT, pp. 195–202. Springer (2015)
6. Garcia-Sanjuan, F., Jaen, J., Catala, A., Fitzpatrick, G.: Airsteroids: Re-designing the Arcade Game Using MarkAirs. In: Proceedings of ITS, pp. 413–416. ACM (2015)
7. Greenberg, S.: Collaborative physical user interfaces. Communication and Collaboration Support Systems (2004)
8. Haller, M., Forlines, C., Koeffel, C., Leitner, J., Shen, C.: Tabletop games: Platforms, experimental games and design recommendations. Art and Technology of Entertainment Computing and Communication (2010)
9. Ishii, H.: Tangible Bits: Beyond Pixels. In: Proceedings of TEI, ACM (2008)
10. Krzywinski, A., Mi, H., Chen, W., Sugimoto, M.: RoboTable: a tabletop framework for tangible interaction with robots in a mixed reality. In: Proceedings of ACE, ACM (2009)
11. Leitner, J., Haller, M., Yun, K., Woo, W., Sugimoto, M., Inami, M., Cheok, A.D., Been-Lirn, H.D.: Physical interfaces for tabletop games. ACM Computers in Entertainment (CIE) 7(4) (2009)
12. Lilis, Y., Savidis, A.: An Integrated Development Framework for Tabletop Computer Games. ACM Computers in Entertainment (CIE) 12(3), 1–34 (2015)
13. Magerkurth, C., Cheok, A.D., Mandryk, R.L., Nilsen, T.: Pervasive games: bringing computer entertainment back to the real world. ACM Computers in Entertainment (CIE) 3(3) (2005)
14. Magerkurth, C., Stenzel, R., Prante, T.: STARS - a ubiquitous computing platform for computer augmented tabletop games. In: Proceedings of UBICOMP,   (2003)
15. Mandryk, R.L., Maranan, D.S.: False prophets: exploring hybrid board/video games. In: Proceedings of CHI Extended Abstracts, pp. 640–641. ACM (2002)
16. Marco, J., Baldassarri, S., Cerezo, E.: ToyVision: a toolkit to support the creation of innovative board-games with tangible interaction. In: Proceedings of TEI, ACM (2013)
17. Mellis, D., Banzi, M., Cuartielles, D., Igoe, T.: Arduino: An open electronic prototyping platform. In: Proceedings of CHI Extended Abstracts, ACM (2007)
18. Mora, S., Di Loreto, I., Divitini, M.: From interactive surfaces to interactive game pieces in hybrid board games. To appear in the Journal of Ambient Intelligence and Smart Environments (2016)
19. Mora, S., Di Loreto, I., Divitini, M.: The Interactive-Token Approach to Board Games. In: Proceedings of AMI, pp. 138–154. Springer (2015)
20. Mueller, S., Dippon, A., Klinker, G.: Capture The Flag: Engaging In A Multi-Device Augmented Reality Game. In: Proceedings of ITS, pp. 277–282. ACM (2015)
21. Ullmer, B., Ishii, H., Jacob, R.J.K.: Token+constraint systems for tangible interaction with digital information. ACM Transactions on Computer-Human Interaction (TOCHI) 12(1) (2005)