CONSTRUCT: AN EDUCATOR-ORIENTED DESIGN STUDIO FOR AMBIENT EDUCATIONAL GAMES

Evangelos Poutouris1, Maria Korozi1, Asterios Leonidis1, Nikolaos Louloudakis1, Constantine Stephanidis1,2

1Foundation for Research and Technology – Hellas (FORTH) – Institute of Computer Science (ICS) (GREECE)
2 University of Crete, Department of Computer Science (GREECE)

Abstract

This paper presents ConstrAct, an authoring tool that enables educators to create new content and game scenarios for various types of ubiquitous educational mini-games through an intuitive UI. The main goal of this system is to provide an educator-friendly mechanism through which the user would be able to “program” some of the games’ aspects. Five game types are currently supported, giving the educators the opportunity to create a rich portfolio of fun exercises for their students. The editor has been successfully used by engineers to design more than 20 mini-games with various themes for different systems. This work reports the requirements elicitation process, the functionality of the tool and the usability findings of the heuristic evaluation conducted with UX experts.

Keywords: authoring tool, ubiquitous educational games.

1 INTRODUCTION

In the recent years, many educators adopt the playful learning approach, where students are engaged as active participants [1], as an alternative to the traditional way of lecturing where they passively receive information [2]. Modern technology promoted the creation of a plethora of educational applications; nevertheless, educators are mainly confined to use the content that comes prebuilt with each application. Such limitation becomes even more prominent when ubiquitous learning applications [3] are concerned, as in such cases the need for customization spans along multiple dimensions both game-related (e.g., content, gameplay, interaction techniques, etc.) and environment-related (e.g., physical-to-digital world mapping, sensor networks, configuration of computer vision algorithms, etc.).

This paper presents an authoring tool, named ConstrAct, which enables educators to create new content and game scenarios for various types of ubiquitous educational mini games through an intuitive User Interface. It provides an educator-friendly mechanism that allows the customization of several aspects of a game with minimal effort (e.g., content, theme, storyline, gameplay, etc.) or the design of entirely new games from scratch.

Currently, the editor supports five different game types, namely “Multiple choice quizzes”, “Find the correct sequence” games, “Classification” games, “Wrong item detection” games and “Execute a process” games. The generated content is interpreted at real-time by the respective custom game runtime to appropriately render the game, whereas any configuration parameters determine its runtime behavior and activates / deactivates the available interaction modalities. In more details, the resulting mini-games are capable of supporting multimodal interaction as the players can interchangeably use either the digital facilities offered by a typical GUI or various physical input sources that facilitate interaction over horizontal non-digital surfaces (e.g., printed cards on a tabletop setup [4], custom-built devices that enable “point-and-click” on a wooden table [5]. Furthermore, multisensory output alleviates difficulties in comprehending written text, thus addressing a mixture of situations that could likely be present in the target audience of gamers (e.g., lack of reading skills, limited cognitive functionality in case of disabled users, etc.). In addition to content creation, ConstrAct encapsulates the complexity stemming from the incorporation of the involved technologies and assists the authors, using highly sophisticated algorithms, to properly configure any relevant parameters (e.g., designate the boundaries of the touch-enabled surface, set the brightness level to match the lighting setting of the educational space, generate and print a QR-code to identify an individual user, select the appropriate card face to ensure that it will be correctly recognized by the computer vision algorithm, etc.).
2 RELATED WORK

Gamers have always been fascinated and eager to build their own games and share them with other passionate peers around the world [6]. Game design studios respond to that demand by offering the necessary content creation tools [7] and enabling players to generate their own content. Many examples of mainstream online games confirm the positive acceptance of that approach, as the user-generated content became extremely popular amongst the relevant gaming communities [8], [9], [10], [11], [12], exceeding initial expectations. Moreover, recent statistics highlight that the design of digital game assets (i.e., 3D models, levels, game artifacts, etc.) is a highly active and profitable domain [13] endorsed by many domain professionals. Certain studios went one step further than content creation; they enabled players to author game code and gameplay events via scripting, thus permitting them to modify the original game (a practice commonly known as game mod) [14] in order to either customize it to their needs and preferences [15] or to create brand new games based on it [16].

Nevertheless, most of the aforementioned approaches target existing commercial video games and are employed by either domain professionals (e.g., IT experts, game designers, etc.) or highly skilled players (e.g., professional eSports players, etc.). When it comes to educational games, the domain is radically different; games are much simpler and content authors are either the educators [17] or experts from various disciplines [18] with limited experience in digital games and computer software. Therefore, work in this context mostly refers to services and applications that offer graphical authoring wizards through which educators could adjust the content of a game and a limited set of its rules, without affecting the major gameplay aspects and principles (i.e., interaction, game mechanics, etc.).

Various online services enable educators to adjust the content of several well-known arcade games in order to compile their own personalized learning applications [19], [20], [21], [22], [23]. Despite the plurality and the ease-of-use of such services, the resulting games are far less engaging than most of interactive game types [24], as they range from simple quizzes to matching games and crosswords. To address those shortcomings, more sophisticated desktop-oriented applications [25], [26], [27] have been implemented that permit the creation of full-scale games that incorporate physics, feature a complete storyline, use rich animations, etc. However, the main target audience of these applications are experienced game designers who aim to develop feature-rich games, rather than educators.

So far, the aforementioned systems allowed the modification of video games that run on common gaming platforms (e.g., PCs, gaming consoles, smartphones, etc.); however, the emergence of the Internet-of-Things (IoT) [28] resulted in the introduction of context-sensitive games that exploit advanced technological affordances (e.g., augmented reality, sensors, etc.). The inclusion of Ambient Intelligence introduces another dimension of complexity in the editor’s functionality, as it has to facilitate the modification of aspects (e.g., computer vision parameters, etc.) with which the users have limited experience. [29], [30], [31] enable educators to create from scratch educational games that make use of such facilities (mainly AR or QR codes) through GUIs; yet, their openness and the lack of any predefined games complicates the overall process and constitutes it less friendly for the target audience. Acknowledging the aforementioned benefits, the “ConstrACT” design studio constitutes an innovative interactive design studio that facilitates both content creation (for multiple difficulty levels) and ambient facilities customization for various types of ubiquitous educational mini-games.

3 REQUIREMENTS ELICITATION AND DESIGN PROCESS

Initially, ConstrACT was built to support the creation and modification of educational games for the students of the Rehabilitation Centre (RC) for Children with Disabilities in Heraklion, Crete. In particular, three systems, consisting of several mini games, have been created through this tool.

The HomeGame. It aims to familiarize students with household objects, the overall home environment and the daily activities that take place in it [4]. It provides two alternative input modalities: (i) touch-based interaction (via a touch-enabled PC) and (ii) interaction using physical cards placed on a board.

The Money Game. It teaches students the coins and bills of Euro and their value, supports familiarization with monetary transactions through virtual shopping and promotes good behavior regarding shopping and monetary transactions in general [32]. It provides two complementary input modalities: (i) mouse-based interaction and (ii) real Euro coins and banknotes.

The Cooking Game. It aims to teach students how to cook simple meals (e.g., bread with butter and honey, lettuce salad, pasta with tomato sauce, etc.) and fundamental rules of safety and hygiene that should be applied during food preparation [32]. In terms of interaction, the player can make a selection
using a custom point-and-click device and introduce new items (e.g., ingredients, cutlery, utensils, etc.) by showing the appropriate printed card to a camera.

Requirements elicitation involved questionnaires and semi-structured interviews [33] with twelve RC educators with professional expertise in various domains, such as psychology, occupational therapy and pediatrics; apart from the functionality-related features yielded, major focus was also put on extracting requirements regarding the game’s authoring tool. Following the aforementioned process, the identified requirements for ConstrAct were augmented through cross-disciplinary focus-groups involving experts from various domains (i.e., UX experts, developers, educators, etc.) to incorporate additional functionality that aimed to evolve it from a limited content manager system to a full-featured design studio, able to create ubiquitous educational games that share similar gameplay principles and can be executed via the supported game runtimes. Based on those findings, high-fidelity prototypes were designed and subsequently reviewed by UX experts in order to obtain their feedback. Eventually, the fully implemented system – including the necessary refinements – was re-assessed by five UX experts via heuristic evaluation [34] in order to test the overall usability and address any problems before conducting a full-scale user-based evaluation with the target audience (i.e., educators).

4 BUILDING MINI-GAMES WITH CONSTRUCT

Five game types are currently supported, giving the educators the opportunity to create a rich portfolio of fun exercises for their students. Next, the requirements of each type will be presented along with examples of games built using the ConstrAct studio and deployed in educational environments [35].

4.1 Multiple choice quizzes

The ConstrAct studio enables the creation of multiple choice quizzes by inserting questions with at least two possible answers. When creating a new question, the educator initially has to: (i) type the question’s textual description and (ii) define the number of possible answers. Next, for every answer, the educator types its textual description and optionally inserts a couple of images to better communicate its meaning, and specifies any positive and/or negative feedback messages by providing their textual descriptions and illustrative images. Figure 1 displays the data required for defining the question “Which is the correct behavior” in the context of the Home Game. In that case, the two possible answers are accompanied by their illustrative pictures, while a negative feedback message will be rendered by the game runtime if the player makes the wrong choice.

![Figure 1. A quiz question with two possible answers and a feedback message for the wrong one.](image)

4.2 “Find the correct sequence” games

The player’s objective in these games is to place a number of items (e.g., task steps, letters, objects, etc.) in the correct order. ConstrAct facilitates the creation process by decomposing it into two distinct phases. In the “declaration” phase the educator collects the desired items in a temporal cart either by importing existing ones or by creating them from scratch; the creation process for each item requires the definition of its description and of a representative image to convey its meaning. Then, in the “usage” phase, the educator arranges them in the correct sequence(s).

The daily activity “Taking a bath”, which appears in the Home Game, is an illustrative example. It consists of the following five steps at the simple level: (1) “get into the bathtub”, (2) “wash hair”, (3)
“wash body”, (4) “get out of the bathtub” and (5) “wipe body”. The correct sequences are both sequence \{1, 2, 3, 4, 5\} and sequence \{1, 3, 2, 4, 5\} since washing the body can as well precede washing the hair. To accomplish that via the ConstrAct studio, the educator has initially created the first sequence by importing the appropriate steps from the pool of available items (during the declaration phase) and then arranging them via drag ‘n’ drop (Fig. 2a). To simplify the definition of alternative sequences, the studio permits the duplication of an entire sequence; therefore, the educator had to rearrange only a subset of the steps. This straightforward action alleviates much of the workload making the creation of alternative sequences a quick and simple task.

![Figure 2. (a) A snapshot of the design studio while manually defining the correct sequence of the daily activity “Taking a bath”. (b) The same activity featuring the “pin” facility.](image)

However, manual definition of every sequence is not appropriate for cases with many correct alternatives, since it would require a lot of time and effort to specify each one of them; to this end, the “pin” feature (Fig. 2b) was introduced through which educators can designate the correct position for an item in a given sequence to always be the “pinned” one (e.g., “get into the bathtub” should be always the first step). As a result, the studio will automatically generate, behind the scenes, all the appropriate sequences by populating the empty positions with the remaining items in every possible combination. Finally, given that ConstrAct inherently supports multiple levels of difficulty, the definition of the correct sequence(s) for the same activity for a different level is straightforward: the educator simply duplicates the existing sequences from another level and injects the new step(s) to each one.

Currently, the “Find the correct sequence” game type has been used to create mini-games for different systems. In particular, for the “Home game”, it was used to build a mini-game that aims to improve the player’s skills related to daily living and self-care, for which eleven activities (e.g., “bathing”) and four routines (e.g., “What do we do at night before going to sleep”) were decomposed into steps that the player can learn about the value of the coins and bills of Euro by sorting them in an ascending order. Finally, it was also used to create a single-player “Spelling game”, which allows the players to train their language skills by forming the words that correspond to the items on the screen.

### 4.3 “Classification” games

This type of game allows the players to train their classification skills by placing items on their correct location on top of specially designed backgrounds (i.e., boards, maps, diagrams, etc.) either digitally via touch or physically using printed cards on a tabletop setup. The background illustrations may vary from simple (e.g., two distinct areas to categorize coins and bills, six colored areas to classify similarly
colored items, etc.) to rich representations of rooms (e.g., a kitchen where the player has to find the correct location of items such as utensils or food), places (e.g., a backyard for placing the animals in their homes), landscapes, maps (e.g., the map of Europe for placing famous sights to the countries that they belong) and much more. ConstrAct facilitates the creation of this type of games by guiding the educator to gradually provide the necessary details (Fig. 3a). In particular, the educator has to:

**Define the map/board** of the game by inserting a picture (e.g., a kitchen, a backyard, map of Europe, etc.) which will be rendered as the background and will “host” the areas of interest.

**Designate the areas of interest** (Fig. 3c) on that picture (e.g., stove, cupboards, refrigerator, etc.). To do so, the educator simply uses her pointing device (i.e., computer mouse) to outline any visual element that is expected to be interactive.

**Collect the items to be placed on the areas of interest** (e.g., pot, dishes, glasses, etc.) in a temporal cart (Fig. 3b); similarly to other games, the items can be either imported from the pool or inserted from scratch.

**Associate the items with the areas of interest** (Fig. 3d). To do so, the educator has to: (i) select the desired area of interest, (ii) choose the desired item(s) and (iii) set their position. The latter is accomplished either by dragging the item to a specific location or by setting its exact coordinates (i.e., X, Y) with respect to the board. Considering that an item might have more than one correct positions, ConstrAct permits its association with multiple areas of interest either on the same or across different boards. Finally, a “show/hide” switch reveals any previously associated items to provide an overview of the populated board, or hides them to remove any visual obstacles and deliver a “clean” board.

![Figure 3. (a) The design studio while creating a classification game. (b) A cart is used to collect the desired items. (c) Definition of a new area of interest. (d) Association of items with areas of interest.](image)

![Figure 4. Two classification games created through ConstrAct.](image)
4.4 “Wrong item detection” games

Similarly to the “Classification” games, this game type also revolves around specially designed boards; each board is digitally enhanced with several items, some of which do not belong in it conceptually, and the player has to designate the wrongly placed items. When creating such a game, the educator firstly has to define the board, select the items (correct and wrong) that are going to appear on it and set the number of correct and wrong items to be displayed per round. As soon as the game is launched, the game runtime randomly selects the designated amount of correct and wrong items and displays them in arbitrary positions on the board. However, depending on the purpose of the game being designed (e.g., a game to identify which items do not belong in a specific house room), placing the items into random positions is not ideal; to this end, ConstrAct enables the educators to determine the exact position of each item, ensuring that they will be placed on “valid” locations where they fit appropriately in terms of physical dimensions. That way, the player will put effort to identify the wrong item(s) rather than merely detect them by scanning the screen for visual abnormalities (e.g., a milk carton floating in the middle of the bathroom instead of being placed on the top of a surface, etc.).

4.5 “Execute a process” games

This is the most complex—in terms of creation—type of game, as the player has to follow a process (e.g., cook a recipe) by completing different sorts of activities (e.g., bring or mix ingredients, use utensils, interact with appliances, etc.). Therefore, to accommodate this variety, ConstrAct offers three distinct types of basic building blocks:

**Actions:** determine what the player has to do next; e.g., bring {an ingredient on the kitchen counter}, slice {an existing ingredient}, wait, mix {a number of ingredients}, etc. Every action derives from a set of predefined types that communicate their behavior to the game runtime (e.g., introduce a new item denotes that user input is required, combination between existing items limits the type(s) and the number of items that can be used, etc.); it also has a descriptive name and an accompanying image.

**Items:** every interactive artifact that can be shown in the game (e.g., lettuce, knife, etc.). An item has a specific type (e.g., interactive, passive, movable, stationary, etc.) that defines its in-game behavior, a name and a representative image.

**Areas of interest:** visual elements that are expected to be interactive (e.g., the sink) and might “host” items or expect specific actions. They are specified similarly to the “Classification” games, however, in this particular type, they might allow or reject certain actions (e.g., a power outlet is an area where a device such as a toaster or a coffee machine can be “plugged”).

Before starting the compilation of the game flow, the educator must collect the “interactive” items (e.g., ingredient, portable appliances, etc.) that will be used during the process; next, she can insert the desired steps of the process. Each step is marked as mandatory or optional and is comprised by one or more of the following components:

1. The accompanying **multimedia:** representative pictures and videos that will be reproduced as a reward after the player completes the step.
2. The **action** expected to be performed by the user. Depending on the action type, the game runtime will wait for: (i) certain items to be selected, (ii) new items to be introduced or (iii) an educator-defined amount of time before proceeding to the next step.
3. The **items** involved in the current step and the **area** where the step takes place.
4. The **outcome** of the step (e.g., a bowl of chopped potatoes). If the item resulting from that process does not exist, the educator creates it on-the-fly by providing its description and an illustration.

As soon as all the necessary parts of a step are defined, the studio produces on-the-fly a visual and textual representation of the step, providing the appropriate feedback to the educator. Figure 5 depicts a snapshot from ConstrAct that was captured during the creation of the recipe “French Fries” for the Cooking game. In more details, it presents the creation of the step “place the pan on the electric stove” where: the educator has already (i) selected the type of the action required (i.e., place), (ii) inserted the appropriate multimedia, (iii) pointed at the pan, which is located on top of the kitchen table, and (iv) was about to click on the stove to signify that the item (step 3) must be moved to that area of interest. To validate the process, at the bottom of the screen, the educator can see the auto-generated visual and textual description of the step that has been just created (i.e., Place the pan on the electric stove).
Currently, this game type has found application in the Cooking game, where the students have to follow predefined steps of recipes in order to learn how to cook simple meals. However, it can be used to create a great variety of games ranging from simple exercises (e.g., instructions on how to clean clothes, etc.) to complex simulation games of real activities (e.g., physics experiments, building/crafting games, etc.). As an example, consider the puzzle game “Crazy Machines” created by FAKT Software GmbH [36], which uses a physics engine to simulate various in-game variables such as air pressure, electricity, gravity, and particle effects. The players build imaginative machines and experiment with gears, robots, explosives and more in a virtual lab. The proposed design studio could be used to create a similar interactive game (Fig. 6) where the player would have to: (i) Fetch and mount two gears on the board, (ii) Fetch rubber bands and use them to connect the gears with the valves, (iii) Fetch and mount a wooden tile on the board and (iv) Push the ball to start rolling.

4.6 Content reuse and configuration of ambient facilities

The ConstrAct design studio was built with scalability and reusability in mind. Therefore, apart from defining the game flow(s), it also acts as a general content management system through which the educators can manipulate the entire data pool and potentially reuse content across games (e.g., an item created for game type A could be re-used in game type B). In addition, ConstrAct encapsulates
the complexity stemming from the incorporation of aforementioned ubiquitous technologies and assists the authors, using highly sophisticated algorithms and appropriate visualizations, to properly configure relevant parameters. As an example, the educator, through the studio, can: (i) designate the physical boundaries of the touch-enabled surface, (ii) set the brightness level to match the lighting setting of the educational space, (iii) generate and print QR-codes to identify players, (iv) insert appropriate illustrations for items that could be used physically as printed cards, which maximize the recognition rate and minimize false-positives by the computer vision algorithms, etc.

5 HEURISTIC EVALUATION

A heuristic evaluation experiment of ConstrAct was conducted in order to eliminate any major usability errors before proceeding with user testing. Heuristic evaluation is the most popular of the usability inspection methods and is carried out as a systematic inspection of a user interface design for usability [34]. The process involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics"). In general, heuristic evaluation should involve more than a single evaluator, since experience has shown that different people find different usability problems. According to Nielsen [37], the best approach is to involve three to five evaluators, since larger numbers do not provide much additional information.

During heuristic evaluation, each individual evaluator inspects the interface alone and compares it with the "heuristics". As soon as all the evaluators have completed the aforementioned process, the discovered usability errors are aggregated in a list with references to those usability principles that were violated by the design in each case in the opinion of the evaluators. Next, each evaluator is asked to provide severity ratings for each problem independently of the other evaluators. The severity ratings range from zero ("not a usability problem") to four ("Usability catastrophe") [38] and are used to indicate how serious each problem is and how important is to fix it. Finally, the development team ranks each problem with an ease-of-fix ranking ranging from zero ("would be extremely easy to fix") to three ("would be difficult to fix") to designate the amount of effort needed to address it.

ConstrAct was evaluated by five User eXperience (UX) experts who inspected the interface and judged its compliance with the "heuristics". Their findings revealed 22 usability issues out of which 8 were ranked as cosmetic problems only. The remaining 14 have been prioritized in Table 1, with the most severe and hardest to fix problems listed first. In general, the major usability problems could be easily resolved (based on the ease-of-fix rating), with system-wide undo/redo facility being the most difficult to be incorporated. Following fixing, the system will be evaluated in a full-scale user-based experiment with educators from various professional domains to get additional feedback.

Table 1. The most significant problems found through the heuristic evaluation.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Severity</th>
<th>Ease of Fixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>The same button should not be used for more than one function. Each function should have its own.</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>The user should be able to undo / redo his actions.</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>The user might get confused that in some game types she has to insert items into a cart before being able to use them.</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>It is not clear how to shift from simple mode to shuffle mode in the “Find the correct sequence” games.</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>The cart should have an add button to enable the user to add items on the fly.</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Use warning messages to inform the user about certain action consequences.</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Some literals might be confusing for the user.</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Some buttons should be in more intuitive positions.</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>The colour coding of the buttons is not intuitive, e.g., in some cases the green colour is used for the “save” action, while in others its used as the main colour of a specific game type.</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>When selecting interactive items they should get highlighted.</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

8482
Instead of a cart, selecting the desired items should be a different step in many of the games.  
Whenever a cart is used to select items, it should include search functionality.  
The literals or the icons displayed on some buttons are counter intuitive.  
The cursor should change shape to inform the user regarding the current action (e.g., cross when moving items on a board).  

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Instead of a cart, selecting the desired items should be a different step in many of the games.</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Whenever a cart is used to select items, it should include search functionality.</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>The literals or the icons displayed on some buttons are counter intuitive.</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>The cursor should change shape to inform the user regarding the current action (e.g., cross when moving items on a board).</td>
<td>2</td>
</tr>
</tbody>
</table>

6 CONCLUSIONS & FUTURE WORK

This paper has presented the ConstrAct design studio that constitutes an innovative interactive authoring tool facilitating both content (for multiple difficulty levels) and game scenarios creation and customization of ambient facilities for various types of ubiquitous educational mini-games through an intuitive User Interface. Until now the editor has been successfully used by engineers to design more than 20 mini-games with various themes for different systems. The heuristic evaluation of ConstrAct, conducted with UX experts, revealed various usability issues which will be incorporated in the next version to be used to conduct a full-scale user-based evaluation of the tool with the targeted end-users (i.e. educators) to fine-tune it before its final release.

REFERENCES


