

Coding4Girls Course Platform Manual v1.5f

Olivier Heidmann, CERTH

Spyros Panagiotopoulos, UTH

This project has been funded with support from the European Commission. This software reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



CODING4GIRLS TEACHER'S PLATFORM USER MANUAL

TABLE OF CONTENTS	
Introduction	3
Word of caution	5
A few words on the general structure of C4G	6
1. Logging in	8
2. Courses	12
Created Courses	12
Public Courses	13
3. Creation of a new course	17
Importing Data into a Snap! canvas	19
Creating the course's challenges	26
Table of figures	32



Creative Commons - Attribution-NoDerivatives 4.0 International Public license (<u>CC BY-ND 4.0</u>)

INTRODUCTION

Coding4Girls addresses the gap between male and female participation in computer science education and careers by introducing early methodological learning interventions that make computer science attractive to all. It showcases interventions that target the factors that lead girls to not choose computer science, namely a) misperception of the roles and professional careers; and b) wrong assumption of insufficient skills. The main goal is to attract girls by raising their awareness on the rich possibilities for professional and personal growth that computer science offers and by preparing them for future engagement in computer science careers.

The Coding4Girls platform is designed to challenges the learners to see the big picture before designing a detailed solution, encourages them to consider wider community interests, and also challenges them to think entrepreneurially on how digital technologies can be used to address real-world problems.

The main idea of the project is to provide to the learners an innovative fun and interesting environment for their coding courses using the Snap! programming language, a Berkeley-based derivative of the famous visual programming Scratch, which elaborates on its ancestor by adding some high level programming possibilities. More instructions on how to use Snap! and its capabilities can be found on: <u>https://snap.berkeley.edu/</u>

This coding environment has been tailored to raise the interest of girls into coding by making them play games they particularly like, but boys also appreciate the type of games offered by the platform. All the games offered to the students are related to and exemplify the actual programming concepts that are at the heart of their courses. The courses, using elements of the design thinking methodology, present to the students an



Figure 1: C4G project methodology

overarching issue to solve and present the tools to solve it in a step-by-step approach.

The Coding4Girls project is actually divided into two distinct entities:

- A web-based platform where teachers will prepare their coding course using Snap!, follow the advancement of their students and access a public repository of courses created by other teachers
- A Unity 3D videogame that the students will use to discover and complete the courses prepared by their teachers in a fun, engaging and gamified fashion.

The current manual aims to explain how to use the teacher's platform efficiently rather than describe how it was designed and goes into details about implemented mechanisms rather than specifications and design process. The exact specification can be found in the Coding4Girls game design document.

The manual of the game is found in a different document.

From now one the abbreviation C4G is going to be used to represent the project's full name, Coding4Girls.

WORD OF CAUTION

The platform is a work in constant progress. This means that the online version of the tool is updated frequently in order to add new features and correct bugs, up to several times a day. So until the software reaches it's very final version, this manual is also a work in progress.

You might find for example that there are some discrepancies between some screenshots and the actual images included in the manual, or maybe some features in the software are not detailed in the manual. Don't worry, it just means that a new version of the manual is about to be published.

If you have any question, please feel free to send an email to the lead developer at this email address, he will answer you promptly: <u>olivier.heidmann@gmail.com</u>.

A FEW WORDS ON THE GENERAL STRUCTURE OF C4G

Please note that from now the following terms will be used interchangeably, meaning the same thing throughout this manual:

- teacher's platform and platform
- users and teachers
- players and students
- copy and clone

The Coding4Girls platform has been designed with efficiency and a minimalist credo in mind. In order for students to access to their coding activities (grouped in what the platform calls **courses**), they have to receive from their teacher a certain code corresponding to the relevant course and register themselves to the course by using this very code.

Courses are created by teachers and are functionning as a grouping space for thematically related activities, all connected to an overarching issue. This problem is presented at the very beginning of the course to the students who can brainstorm all together to collaboratively elaborate tentative solutions. They can place postits (akin to what is done in the real-life process) on a board to pin ideas down. These post-its can be made of text (enriched or not), images or video, offering an inovative flexibility. After the brainstorming phase, the students will be given in a step-by-step fashion, specific activites (presented in a consecutive order) designed to present the tools necessary to solve the overarching problem.

Those activities are called **challenges**, with each challenge tackled by each student in the specific order set by the teacher. For example if a teacher wants to create a course about basic programming knowledge, the first activity could concern the concept of booleans, the second conditional structures and the last one loops. In order to access to all the challenges of the course, a student has to unlock them by playing the preceding challenge. In our last example, a student newly enrolled wouldn't be able to access the loop challenge before having followed the one concerning condictional structures.

Each challenge is structured in the following fashion:

- one introductory minigame illustrating the programming concept at hand. The teacher will decide which minigame (if any) to play by selecting them from a list of existing minigames. For the moment 11 different minigames exist, ranging from a Match3 game to a multiple choice question quiz. The presence of a minigame is optional and the teacher creating the course can decide that a specific challenge doesn't require a minigame at all.
- 2. one HTML page (possibly enriched by images or videos) with instructions presenting the context and specific nature of the task to be fulfilled in Snap!
- 3. a Snap! canvas, based on a template provided by the teacher, containing the programming activity
- 4. a Snap! canvas displaying one solution to the activity. This solution canvas is fully optionnal and its presence depends on how the course was written by the teacher.
- 5. As many repetition of points 2) to 4) that are needed by the course. This allows the teachers to divide thir challenges in simple elementary steps where the answer to the preceding activity becomes the template in which the next activity is to be executed, allowing for an incremental scaffolded way of teaching coding.

Once the players has need completed all the challenges of the course, they are lead back to the initial coding problem and will be asked to solve it thanks to the new knowledge they just acquired. At the very end of the course, the players will be able to see all the solutions to the problem that have been divised by the other students. Figure 2 below illustrates the structure of a C4G course.

At any given time, teachers can have access to the Snap! solutions their students submitted.

Teachers also have the ability to either create **public courses** or transform their current courses into public ones. Public courses are presented in a separate menu and allow any teacher that uses the platform to discover existing courses. Any public course can be cloned and then used as a course by a user with their students. The main difference in functionnality between a public and a private course is that the answers from the players will not be registered with a public course. Users have to be careful about this point as it has two direct consequences:

- Transforming a private course into a public one mens losing the entire base of answers from the students. If teachers want to offer one of their course as a public one, they should first clone it and make the clone public
- Using a public course in the context of a classroom will mean that no user solution will ever be recorded. This means using a public course is a good solution for a quick test or demonstration but a bad idea in the framework of the integration of the C4G platform in coding courses.

Private courses subscribed to by a teacher can also be cloned by the said teacher.



Figure 2: Structure of a C4G course

1. LOGGING IN

The teacher's platform is available at this URL: <u>https://coding4girls.e-ce.uth.gr</u>

The platform has been designed to work on desktop but can also be accessed from mobile devices. Google Chrome is the recommended browser to get the best experience out of the platform.

Once you type in the URL, you are greeted with a minimalistic introduction page. The entire platform has been design to be as functional and easy to use as possible with very little fluff of extra things.

The black ribbon at the top is the main menu, from which you can access all the available functionalities. As we just arrived on the platform, we are offered the choice (right hand side of the screen) to login or to register if you don't have an account already. The English flag on the top right of the screen shows the game interface is displayed in English. Clicking on the flag cycles between the available languages for this platform: English, Greek, Turkish, Bulgarian, Croatian, Italian, Slovenian and Portuguese.



Figure 3: The home page

The login page only contains two fields, one where the user can enter their username (or the email address they used when registering) and the other where they enter their password.

One both fields are filled, press the green Login button to proceed. If the remember me tick is selected the next time the user will access the platform they will be automatically logged in.

Coding4Girls			Login	Register 311
	Login			
	Username or E-mail	Password		
	Remember me			
	Login			

Figure 4: The login page

Because this is a platform designed only for the teachers, learners' accounts are not allowed to login.

The Register page asks the user for a username, a valid email address, a password, their first name, their last name and a code.

Coding4Girls			Login	Register 311
	Register			
	Password	Verify password		
	E-mail			
	First Name	Last Name		
	Code			
	This site requires to collect your email adress and name to function propert	y. To continue using Coding4Girls, you must consent our Privacy policy	nit	

Figure 5: The register page

The Code field is here to prevent learners registering as teachers. If the right code is entered in this field, the account created is automatically given teachers rights. Otherwise the account won't be created at all.

In order to know the special teacher code, please contact us.

In both Login and Register page, when an error occurs (for example, the user has entered a wrong password) an error message will appear at the bottom and the corresponding field will have its border turned to red to indicate visually what went wrong.

Coding4Girls			Login	Register MM
	Login			
	Username or E-mail	Password		
	student1			
	C Remember me			
	Login			

Figure 6: Example where user entered the wrong password

Brror Wrong password

Abiding by the GDPR directive, the user also has to accept the privacy policy of the platform in order to be able to register.

Coding4Girls		Login	Register 2010
	Privacy Policy		
	One of our main priorities is the privacy of our visitors. This Privacy Policy document contains types of information that is collected and recorded i Coding4Girls platform and how we use it.	by	
	If you have additional questions or require more information about our Privacy Policy, do not hesitate to contact us through email olivier.heidmann@gmail.com	at	
	General Data Protection Regulation (GDPR)		
	We are a Data Controller of your information.		
	Coding4Girls legal basis for collecting and using the personal information described in this Privacy Policy depends on the Personal Information v collect and the specific context in which we collect the information:	ve	
	Coding4Girls needs to perform a contract with you You have given Coding4Girls permission to do so Processing your personal information is in Coding4Girls legitimate interests Coding4Girls needs to comply with the law		
	Coding4Girls will retain your personal information only for as long as is necessary for the purposes set out in this Privacy Policy. We will retain an use your information to the extent necessary to comply with our legal obligations, resolve disputes, and enforce our policies.	nd	
	If you are a resident of the European Economic Area (EEA), you have certain data protection rights. If you wish to be informed what Person Information we hold about you and if you want it to be removed from our systems, please contact us.	al	
	In certain circumstances, you have the following data protection rights:		
	 The right to access, update or to delete the information we have on you. The right of rectification. The right to object. The right of restriction. The right to data portability 		

Figure 7: Coding4Girls teacher's platform privacy policy

Once all the information fields have been filled, clicking on the green SUBMIT button proceeds to the main screen. For the following examples, we are going to be logged in as "teacher1" a fictional teacher used as an example throughout this manual.

Once logged in, the user can see on the bottom a success message that temporarily displayed (it disappears when the green bar at the bottom reaches the left hand side).



Figure 8: Login success message

This type of success message (and the error message shown in figure 6) is routinely displayed throughout the software as a feedback for the users when they take some actions such as editing parameters.

2. COURSES

CREATED COURSES

The Courses menu displays both the courses the teacher created and the courses the teachers has copied or subscribed to. If you have created a public course, it will not appear here but in the Public courses menu. Before making a course public, a good practice is to clone it first and then make its clone public rather than the original course.

Courses from other teachers are subscribed to by using the code of that specific course. The courses screen

CodrastGala Garme Victoria	See by Sade bits Times yours		Ø Sinkhalmerer, regen BB
	Created courses		
	My cloned course Chameleon	2.8	
	Test1 UL Description of the Assemblic	2.0	
	Second Tryout beautif Spear of the database		
	Demo course a man for demonstration pagean.		
	Test nor min		
	Test 2.0		

Figure 9: The courses page

The principle of pairing a course with an access code gives us an elegant way to deal with a host of usual issues in that kind of platform (such as scheduling, rights, etc.) as you can only access a course if you have the code for it and you need nothing more than your code and an account to access the data you need. It's akin to a subscription system. In order to unsubscribe from a course that you haven't created, you simply have to click on this little icon

As we can see on figure 8, some courses do display this icon and some others do not, marking if the course has been created by the user (and hence is part of their core list of courses) or by another teacher (and hence can be subscribed or unsubscribed at will).

To help teachers categorize courses visually, the platform provides a tag system that gives the ability to the teachers to make some keywords appear as badges on the courses list. Yellow badges, such as demo, represent the tags defined at the level of the entire course of the lobby and the blue ones, like Random, are the tags that appear on each of the course's challenges.

The ability to copy or clone a course is provided, accessed by clicking this icon 🛄 . When this button is used, the entire contents of the copied course (with the exception of all the enrolled students' answers) are now at

your disposal as if it was one of your own courses. A pop-up will appear asking you to fill in a new name, description and code for the course. Remember that course codes are supposed to be unique.

Coding4Girls Courses Public courses	Clone course		
Join By Code	e Name		
Demo	Description	2-0	
A course fo	r demonstration Code		
ETTIOE Repuyadh	Tag Movement) (Banda		
		Save Discard	

Figure 10: Clone course window

PUBLIC COURSES

The teacher platform also presents a repository of courses created by other fellow teachers and platform users that intended their creation to be publicly available to all the users.

In order to reuse a public course in your own classroom, simply copy/clone by clicking on the ^{lim} icon and it will appear in your list of own courses.

Public courses	
Q South by menutaneighter Sameh by tag	
Far ballare lo sprite 2 %	
Scoprire Snap! 2-9 Ander of Automa exception Diverfacile of longer a programmer if prove spatia in mode the 4-manual experi-	
Vacanze estive di un Camaleonte 2: # Imparare a proprietazione con para amplica dinar l'oppara camina il proprio colleve estande il active ambli dinari. 2: # Il proprio camina il proprio colleve 2: #	
Καλοκαιρινές διακοπές του Χαμαιλέοντα	
Ανακαλύψτε το Snap! : μετακινήσε μία εικόνα διάδησε του μάδησε να ανακαλομούν του υδητήσεμα δρηγοημοδημότε διαφ και κα ανάδαστάσμαι τη τρώτη ποια τούτοι από του παίτε και τοι μάδο. ())	

Figure 11: Public courses

Courses can be filtered by language, associated tags or name as the complete list of public courses can be quite long.

EXPLORING A COURSE

By clicking on a course (either in the Courses or the Public Courses menu), the teacher can access the collection of challenges inside the course. If there are no challenges created, a grayed out "None" will be there instead.

On the left of the page there is a list which contains the current enrolled members of the course (this information is displayed only if the course is yours).

In the figures below, the account we are using is not the one that created this lobby. In this case, we still have access to the contents of the course and its challenges but they can't be modified and the access in the users' solutions is restricted.

Pressing the button will take you to the settings of the course. The layout of the page is exactly the same with the course creation page but it is prefilled with the details of the current course. Those can only be modified by the creator.





Figure 14: Course page viewed by the course creator

On the side menu, useful details about the course can be seen. Here OlivierTeacherC4G the creator's name is mentioned. Below that, the password of the course appears as educourse and finally, the members of the course can be seen in a list



Figure 15 : The member list of course

Here, challenges are organized in boxes, containing the following information from top to bottom:

- The name of the challenge. The number on their name shows their current index.
- A description underneath

• The mini game of this challenge at the bottom.



Figure 16: The challenges information boxes

The arrow is used as a visual clue to indicate the order of the challenges. The teachers can change this order by simply dragging the challenge boxes and dropping them in the position they want. The changes are saved automatically.

Pressing the cog on the top corner opens the challenge settings page. This page, like course settings, is similar to the challenge creation page (of course with the values of the selected challenge prefilled) so more details can be found in the following pages.

3. CREATION OF A NEW COURSE

When inside the Courses menu, pressing the button Create new course will lead you to a new screen that allows you to create a new course. The following information is required:

- Name
- Code
- Description
- Learning objectives and expected learning outcomes
- Tags
- Public course

You can create one or multiple tags that will appear on the courses list as yellow badges. Multiple tags are separated using the semi-colon character ";" that acts as the delimiter between different tags. If you want to tag your course as being for age 12 and being about loops, then you would type "age12;loops" in the Tag field.

By checking the Public course checkbox, the created course will be public. Public courses will be visible by all teachers and will not register the answers from the users. A good practice is to use public courses in classes only for demonstration purposes and clone/copy them to make them your own if you want to use them for an actual course where you will want to use the full capacity of the platform.

Select the language of the course by clicking on the flag box and selecting the one corresponding to your course's language. Eight language flags are available in the framework of the C4G project: English Greek, Turkish, Bulgarian, Croatian, Italian, Slovenian, and Portuguese.

Below are 3 tabs each containing a different window: "Instructions", "Snap Template" and "Brainstorm canvas".

At the bottom of the screen (don't forget to scroll down to see them), you can find the "Save" button that creates the new course and the "Discard" button that cancels the creation and takes you back to the courses page.

Coding4Girls Dourses Public courses		O Oliviar Inschort, Ally Lauport 🛤
	Paana	
	Code	
	Description	
	Laarning Objectives and Expected Learning Outcomee	
	Tay	
	Public course	
	Language Std *	
	Instructions Flouid Instructions Single Implaite Brainstorm same	
	D Z U D 19 V9 Hi Ha IE IE K, K ⁴ IE IE (T Harma F Norma F A K Sensbert F Sc Z, S IE IE Z	
	inset Had Pare	

Figure 17: The course creation screen

INSTRUCTIONS TAB

In the instructions tab the teacher can fill some information for the students about the content of the course. It will be displayed to the user in-game at the very beginning of their course, before the Snap! canvas containing the issue in code is presented. It should typically contain a global presentation of the situation and the issue at hand and should introduce the following screen, the canvas of Snap! code that the students will need to use to solve the problem.

The instructions tab contains a rich text editor allowing teachers to make the instructions more visual pleasing. It also allows the use of embedded videos and images. The icons and the functionality of the editor are similar to the ones found in the most common text editors and will be not explained here.

Two sets of instructions exist for the teachers to fill:

Global instruction presenting the problem and displayed as the very first screen the students will see in the course

Final instructions that will be displayed when the students have followed all the courses challenges, just before they are asked to write the code to solve the main problem. This set of instruction should present a recap of what the overarching problem is and what the students have now learnt to solve it.



Figure 18: The instructions editor

SNAP! TEMPLATE

This template present the problem to be solved, both at the very beginning at the course and in the end of it when the students have now learnt all the knowledge they need. It should present the problem in a certain context. Figure 13 presents the example of a course where the students will need to write code to move a sprite representing a chameleon around the screen and dynamically change its physical appearance based on the background. The Snap! template only shows the background and all the chameleon related asset, with no line of code already written. The very first time the students see it (at the very beginning of the course) the students can begin trying to solve the issue, but should lack the knowledge to do so fully. The second time the students are presented this template, after they have completed all the courses challenges, they should now be able to fully complete the task.



Figure 19: The Snap! template of the chameleon course

IMPORTING DATA INTO A SNAP! CANVAS

Rather than creating their own Snap! canvas for the students to work on every time, Snap! offers the possibility to import files. Those files can be entire solutions or simply assets such as graphics or sounds.

A good practice for a teacher would be to start their course from the end with a full solution loaded as the very last step of the course, and then remove parts of the solution one by one up until arriving at the very beginning of the course with an empty solution.

In order to import something in Snap!, you need to click on this icon in the Snap! menu bar. Then select Open (see figure 20), and click on the Computer icon (see figure 21). This will open the usual window (depends on your operating system, the Windows one is showed figure 22) to select a file. Click on the file you want and then Open (or double click on the file) to import it inside the Snap! canvas.



Figure 20: Importing menu



Figure 21: Import window



Figure 22: Selecting a local Snap! xml file

Files containing a full (code + assets) Snap! solution are usually in the *.xml format. Sounds are usually in *.mp3, *.wav format and images in *.png or *.jpg format. This might vary a lot, especially between operating systems.

Be careful, there is currently a 3Mb size limit for any uploaded file. If you receive an error message while or after uploading a file, it might very well be linked to this limit. Try to upload as small as possible files.

Once a file has been uploaded, you can use it as if it was a native part of Snap!. Figure 23 shows the example of a full Snap! activity uploaded, with both the code and assets ready to be used.

Note that you can import file for any Snap! canvas in your course, whether it's the canvas concerning the main issue, one showing one step of an activity or a solution canvas.



Figure 23: A full Snap! activity has been imported

BRAINSTORM CANVAS

The brainstorm canvas is the place where all the students in the course can communicate and share ideas about how to solve the problem that has been presented to them. The teacher can promote or guide this procedure by placing some post-it which will be visible to all students the moment they enter the canvas.

Add new note
Figure 24: The brainstorm canvas
The canvas will resize automatically to fit all the notes.

Pressing Add new note allows you to create a new post-it.

POST-IT NOTES

By default the default post-it note is only text, with the message "Edit me Double Click me" displayed. "Edit me" is in italic and "Double click me" is in bold, showcasing from the get-go the formatting capacities of the post-it text engine.

Edit me	
Double click me	
ā	6

Figure 25: The default post-it

In order to avoid issues with users concurrently working on the same canvas, when a user adds a new note, the note won't appear directly on the canvas but it will open a pop-up instead. There, the post-it can be modified as it was directly on canvas and by clicking save it will show up to all the other users too.

click me		
	CICK IN	

Figure 26: Add new note pop-up

To change this text the user has to double click on the post-it (double tap on mobile) on the pen icon on the

bottom left and to destroy the post-it the user has to click on the trashcan icon \blacksquare on the bottom right. In order to confirm deletion, the button has to stay pressed for a certain amount of time before the note is deleted. Once the blue bar on top of the screen is full, the post-it is deleted. This mechanism has been introduced to avoid any possible accidental deletion.

Edit me	
Double click me	
ā	6

Figure 27: Post-it getting deleted

Figure 28: Editing the text

All the changes done to a post-it are immediately shown to all the other members of the team if they are consulting the brainstorm canvas at the same time.

When the user wants to add a photo, they are prompted with a button asking them to select a file from their hard drive. The said file will be automatically and transparently be displayed on the screen, being uploaded to our servers for them to handle it.

anane Double cl	ck me	
	37 (MA)	
Choose File No f	le chosen	

Figure 29: Inserting a photo

The icon at the right of the camera icon is a text icon \equiv , in case the user wants to go back to a text only post.



Figure 30: a photo embedded in a post-it

If the user wants to insert a video, they are prompted with a message asking them to copy here the URL of the video. Youtube, Vimeo and Dailymotion are supported at the moment.

Figure 31: Inserting a video



Figure 32: A video embedded in a post-it

At the bottom right of the post-it we can see a padlock icon. It is open ⁶ when the post-it is editable or locked

when the post-it cannot be edited. Only teachers have the possibility of locking and unlocking post-its, students can only know if a post is locked or not without being able to act on it.

Grabbing the little triangle symbol at the bottom left of a post-it allows to resize the post it to the desired size. This is especially useful if the post-it contains a huge photo or a long text, as otherwise the content will just overflow the post-it.

LPML .
Chi al Johns Houring.
The at advance to be feel with titles
The subsect subscreen subscre
····
Number of the second statistics
Or leave & entry's and use the B-4 instituted
An net out o when have not recovery by rune
INTER CONTRACTOR OF A CONTRACTOR OFTA CONTRACT
sufficient which can be particulated and the first
stration.
sources and the set of set of set of set of the second set.
the set of the set of the labor the
must be a set of a set of the set of the set of the set.

Figure 33: Example of content overflowing

To avoid such overflow, please re-dimension your post-it accordingly.

POST-ITS ENRICHED TEXT RULES

The complete reference for the enrich text rule is available at this URL: <u>https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet</u>.

Note: not all functionalities described on this web page have been implemented in DesignIT.

Headers can be defined by using # symbols (the more # there are, the smaller the header's text) or alternatively by underlining the text with the = symbol.



Figure 34: Post-it containing various headers

Emphasis (italic, bold, bold and italics or strike through) can be obtained by placing the text between * and _. If we write *italics* or _italics_ the post-it will display *italics*, **strong** or __strong__ are displayed as **strong**, and **_italics strong_** are displayed as *italics strong*. Finally ~~strike~~ is displayed as strike.

Deptans, and fails, with annexis or orderstream	
2010 group haits and white an antistate in an adventure	
Control emploie off-amelias and and environme	
Tabledy suger sees that these Second date	
	8

Figure 35: Post-it containing emphasis

The system also supports bullet point lists, numbered lists and hypertext links (see the linked URL above for more details).



Figure 36: A post-it containing lists

CREATING THE COURSE'S CHALLENGES

Once a course has been created, click on it to access its detailed presentation. Being the course owner grants

you access to more buttons than the only presented above. The buttons accessible to the owner are the following:

•	Create New Challenge The challenge creation page
•	Answers The answers for each challenge by each user
•	Course answers The answers of the course challenge by each user
•	Brainstorm canvas

CREATE NEW CHALLENGE

The challenge creation page is similar to the course creation page, especially in the upper part of the page. Each challenge needs to have a name specified and a short description.

53	G			Edit challeng	e	
104	-			Luitonining		
	Ersty	challenge				
Ch	hallen	ge Description:				
4	Dulle	një vittout a niniganë	· ·			
-	i i Ga	ne Category				
	Noce					
Ta	9					
11	10000					
	1000					
	Hone .	Instructions	Snap template	Brup solution	Show solution in game	
	1	instructions.	Stragt template	Erup solution	Shaw solution in game	•
	1	Instructions	Shap tenglate	Imp solution	Show solution in game	0
	2		Srup tengliste	Enep solution	Drew solution in game	0 0 0
	2	Freihuschers	Srogr Georgiate	Errop solution	Shew solution in game	0 0 0
	9 2 2 4		Srup template	Ine solution	Shee solution in game	•

Figure 37: The challenge creation page

The teacher can also choose to attach a minigame to the challenge using the dropdown menu. The menu items with the colored background indicate the category of the items below. Items without a background are selectable. The categories are related with programming topics.

AGirls Courses Public courses	Banning and a second	
enus - autoria - autoria	None	0001-10
	Longe	
	Loops Felit challenge	
	Conditionals Conditionals	
	Candilionale	
	Variation	
	Data types	
	Data structures	
	Statements	
	Sequence of statements	
	Sounda	
	Misserreit	
	Looks	
	Drawing	
	Parallelism	
	Simultaneous sounds/movements/characters/interactions	
	Operators	
	Radi: operations	
	Advanced operations	
	trigonometry	
	Random	
	Reading	
	Events	
	Multiple Owies Quantines	
	Multiple Guestions Game	
	(CAB	
	AR .	
	# Instructions Snap templete Snap solution Show solution in gen	

Figure 38: The minigames dropdown menu

Attaching a minigame to a challenge is optional; the user can decide to attach no minigame at all if needed. Selecting the All category allows the user to select any minigame available.

8		
1	A0	
м	lini Game	
I	Match3 Game *	
ī	Matuto3 Geme	
	Find your path Same	
	Inventory Clama	
	Stepping Geme	
1	Sound Game	
	Sheke Came	
	Puzzle Game	
	Pattern Matching Game	
	Stepping Gene	
	Diss Game	
	Multiple Questions dame	

Figure 39: The All category

There are a total of 11 minigames available for the moment, corresponding to the basic principle of coding in Snap! (plus the multiple choice question one). When a concept and associated minigames have been chosen to be in a challenge, the teacher will have to define the corresponding parameters.

Mini Game Category	1		
Mini Game			
Sound Game			-
Timer			
	٢		

Figure 40: A minigame with one variable

If the mini-game requires questions to be asked and answers to be found, the teacher will need to supply them.

Aini Game					
Multiple Questions (Game	•			
imer					
25					
Question 1	QUestidsfsd	https://.imour.com/6rFfl	Choose file	No file chosen	Correct answer
Hint	Spyros hints you	1			
Answer 1	stdsfs	Image uni	Choose file	No file chosen	
Answer 2	nwf	Image uni	Choose file	No file chosen	8
	sdfs	Image unt	Choose file	No file chosen	8
Answer 3	and and a	image inf	Choose file	No file chosen	
Answer 3 Answer 4	2/20120	manife in a			
Answer 3 Answer 4	55050				
Answer 3 Answer 4	ssusu	and the second s			

Figure 41: A minigame with dynamic amount of questions

In the minigame of "Multiple Questions", clicking the $oldsymbol{\Theta}$ adds one question and it can be removed with $oldsymbol{\Theta}$.

Also, where image upload is allowed, you can either provide your own URL or upload a picture using the "Browse..." button.

Mini Game Category		
Mini Game		
Find your path Gam		- 8
Time		
120		
Direction 2		
Careford (
Control answer		
word scow.		
Question 2		
Correct answer		
Mrong answer		
Question 3		
Correct answer		
Wrong answer		

Figure 42: a minigame with a fixed amount of questions

For all minigames, when there are both the timer and score inputs, they need to be defined to a value because the minigame will end if the player reaches the score or the timer end (So a low score will make the minigame end probably too fast).

List of Tags	Minigame associated
Loops	Match3
Conditionals	Find your path
Variables	
Variables - Data types	Inventory
Variables - Data structures	Inventory
Statements	
Statements - Sequence of statements	Stepping Game
Statements - Sounds	Sound Game
Statements - Movement	Stepping Game
Statements - Looks (appearance of a sprite in snap)	Snake game
Statements - Drawing	Puzzle
Parallelism	
Parallelism - Simultaneous sounds/movements/characters/interactions	Pattern matching
Operators	
Operators - Basic operations	Pattern matching
Operators - Advanced operations	Pattern matching
Operators - Trigonometry	Pattern matching
Operators - Random	Dice Game
Events	Variation of the Stepping Game
Extra Quizz Game	Multiple choice questions

Figure 43: Table of minigames

The tags system works in the same way as the lobby tags. There is also a dropdown menu with some predefined tags.

At the bottom of the page, we can see the list of activities, divided into the instructions/Snap! activity/Snap ! solution steps. You need to have at least one activity per challenge, and you can create one clicking on the plus

icon • You can also delete one (but not if you have exactly one) by clicking on the minus icon • . There are five named columns on that section. The first one represents the number of the level. The second (Instructions) allows you to edit the instructions for the activity (using the same editor as in lobby). The third one (Snap template) allows you to edit the template you want to prefill for your users. The fourth one (Snap solution) allows you to present to the students a solution for that activity. Both last two columns use Snap!. Finally, the fifth column (Show solution in game) is a toggle, that will make the snap solution to appear after students submit their work. Keep in mind that if the toggle is enabled and the snap solution is empty, an empty snap will appear.

ANSWERS

All the Snap! answers the students have submitted in each challenge can be found here, displayed in a table. The table contains each user and its details for every challenge that exists in the course and the rows can be sorted either by username, first name, last name, challenge name or solved challenges, both ascending and descending order.

When there isn't a submitted solution the "Solution link" column will be empty. Otherwise, the row will be highlighted and the word "Solution" will appear on the last column and clicking on it, it will show you the submission of the user.

Coding4Girls Courses						OlivierTeacherC4G Logout 20
	0					
	Unername	First same	1 and married	Challanna Aurea -	Roboting Tale	
		11111200				
	teacher1	teacher1	teacher1	Sounds challenge		
	uteal	Sipyroli	Steel	Sounds challenge		
	OlivierTeacherC4G	Olivier	Heidmann	Sounds challenge		
	student1	First	Student	Sounds challenge		
	teacher1	teacher1	teacher1	Simultaneous sounds/move	ments/	
	uteri	Spyros	Steel	Simultaneous sounds/move	ments/	
	OlivierTeacherC4G	Olivier	Heidmann	Simultaneous sounds/move	empfvta/-	
	student1	First	Student	Simultaneous sounds/move	ments/-	
	teacher1	teachert	teacher1	Sequence of statements/m	ovaimen	
	steel	Spyros	Steel	Sequence of statements/me	overnen	
	OlivierTeacherC4G	Olivier	Heidmann	Sequence of statements/m	overnen	
	student1	First	Student	Sequence of statements/im	overnen	
	teacher1	teacher1	teacher1	MQ challenge		
	steel	Spyrox	Steel	MQ challenge		
	OlivierTeacherC4G	Oilvier	Heidmann	MQ challenge		
	student1	First	Student	MQ challenge		
	teacher1	teacher1	teacher1	Loops challenge		
	steel	Spyros	Steel	Loops challenge		
	OlivierTeacherC4G	Olivier	Heidmann	Loops challenge		
	student1	First	Student	Loopo challenge		
	taucher1	teacher1	teacher1	Looks challenge		
	steel	Spyros	Steel	Looks challenge		
	OlivierTeacherC4G	Olivier	Heidmann	Looks challenge		
	student1	First	Student	Looks challenge		
	leacher1	teacher1	teacher1	Empty challenge		
	uteol	Spyron	Stoel	Empty challenge		
	OlivierTeacherC4G	Olivier	Heidmann	Empty challenge		

Figure 44: Answers for the challenges by user

Coding4Girls Courses

1 to deep bolies	Eastman	1 and many	Challenge and	Robalan Sak	
Usemanie	FIGURATION	Last name	Criatienge name	Solution sink +	
steel	Spyros	Steel	Sounds challenge	Solution	
OlivierTeacherC4G	Olivier	Heidmann	Loops challenge		
Inebute	First	titudent	Loops challenge		
steel	Spyros	Steel	Loops challenge		
teacher1	teacher1	teacher1	Loops challenge		
OlivierTeacherC4G	Olivier	Heidmann	Drawing challenge		
student1	First	Student	Drawing challenge		
steel	Spyros	Steel	Drawing challenge		
teacher1	teachert	teacher1	Drawing challenge		
OlivierTeacherC4G	Olivier	Heidmann	Sounds challenge		
student1	First	Student	Sounda challenge		
leacher1	teacher1	teacher1	Sounds challenge		
OlivierTeacherC40	Olivier	Heidmann	Conditionals challenge		
student1	First	Student	Conditionals challenge		
uteel	Spyros	Steel	Conditionals challenge		
eacher1	teacher1	teacher1	Conditionals challenge		
OlivierTeacherC4G	Olivier	Heidmann	Sequence of statementalmovemen		
etudent1	First	Student	Sequence of statements/movemen		
uteel	Spyros	Steel	Sequence of statements/movemen		
teacher1	teacher1	teacher1	Sequence of statements/movemen		
OlivierTeacherC4G	Olivier	Heidmann	Looks challenge		
student1	First	Student	Looks challenge		
steel	Spyroi	Steel	Looks challenge		
teacher1	teacher1	teacher1	Looks challenge		
OtivierTeacherC4G	Olivier	Heidmann	Randomness challenge		
student1	First	Student	Randomness challenge		

Figure 45: The answers of the challenge by users with a solution

COURSE ANSWERS

The course answers page contains a similar table without the challenge name column. Again, rows containing users who have submitted an answer will be highlighted.



Figure 46: The answers of the course challenge

TABLE OF FIGURES

Figure 1: C4G project methodology	3
Figure 2: Structure of a C4G course	7
Figure 3: The home page	8
Figure 4: The login page	9
Figure 5: The register page	9
Figure 6: Example where user entered the wrong password	10
Figure 7: Coding4Girls teacher's platform privacy policy	10
Figure 8: Login success message	11
Figure 9: The courses page	12
Figure 10: Clone course window	13
Figure 11: Public courses	13
Figure 12: A course page with challenges	14
Figure 13: An empty course page	15
Figure 14: Course page viewed by the course creator	15
Figure 15 : The member list of course	15
Figure 16: The challenges information boxes	16
Figure 17: The course creation screen	17
Figure 18: The instructions editor	
Figure 19: The Snap! template of the chameleon course	19
Figure 20: Importing menu	19
Figure 21: Import window	20
Figure 22: Selecting a local Snap! xml file	20
Figure 23: A full Snap! activity has been imported	21
Figure 24: The brainstorm canvas	21
Figure 25: The default post-it	22
Figure 26: Add new note pop-up	22
Figure 27: Post-it getting deleted	22
Figure 28: Editing the text	23

Figure 29: Inserting a photo	23
Figure 30: a photo embedded in a post-it	23
Figure 31: Inserting a video	24
Figure 32: A video embedded in a post-it	24
Figure 33: Example of content overflowing	24
Figure 34: Post-it containing various headers	25
Figure 35: Post-it containing emphasis	25
Figure 36: A post-it containing lists	25
Figure 37: The challenge creation page	26
Figure 38: The minigames dropdown menu	27
Figure 39: The All category	27
Figure 40: A minigame with one variable	27
Figure 41: A minigame with dynamic amount of questions	28
Figure 42: a minigame with a fixed amount of questions	28
Figure 43: Table of minigames	29
Figure 44: Answers for the challenges by user	30
Figure 45: The answers of the challenge by users with a solution	
Figure 46: The answers of the course challenge	