

# Coding4Girls Game Manual v1.5f

**Olivier Heidmann, UTH**

**Spyros Panagiotopoulos, UTH**

**Costas Katsimedes, 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.



**Erasmus+**

# CODING4GIRLS USER MANUAL

## TABLE OF CONTENTS

<b><i>Introduction</i></b>	<b>3</b>
<b><i>Word of caution</i></b>	<b>5</b>
A few words on the general structure of C4G	6
<b><i>The C4G Software</i></b>	<b>8</b>
Availability	8
The Game launcher	8
<b>1. Logging in and selecting a course</b>	<b>10</b>
The introduction scene	10
The Lobby	10
<b>2. Selecting a challenge in a course</b>	<b>16</b>
<b>3. Inside a challenge</b>	<b>18</b>
Instructions	18
Snap Activities	18
Brainstorm	20
Accessing other students' solutions	20
<b>4. The minigames</b>	<b>21</b>
Dice game	22
Multiple choice questions	24
Match3	26
Inventory	28
Stepping Game	30
Sound game	31
Snake game	33
Puzzle game	34
Pattern Matching game	36
<b>5. UI and pause menu</b>	<b>38</b>
Pause menu	38
<b><i>Table of figures</i></b>	<b>42</b>



Creative Commons - Attribution-NonCommercial 4.0  
International Public license ([CC BY-NC 4.0](https://creativecommons.org/licenses/by-nc/4.0/))

## 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: <https://snap.berkeley.edu/>

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

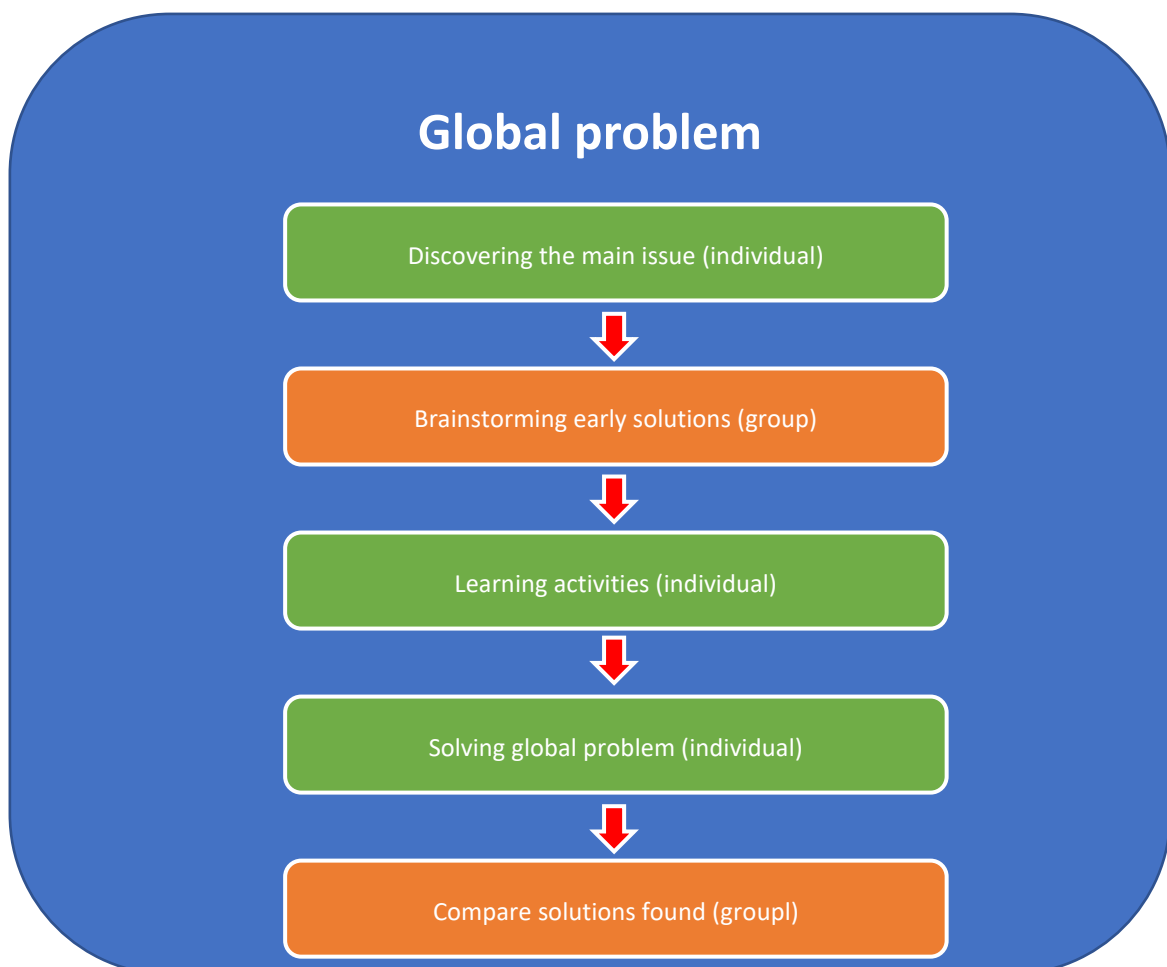


Figure 1: C4G project methodology

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 teacher's platform is found in a different document.

From now on the abbreviation C4G is going to be used to represent the project's full name, Coding4Girls. The Snap! coding space will be often called a Snap! canvas.

## 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: [olivier.heidmann@gmail.com](mailto:olivier.heidmann@gmail.com).

Because of its structure briefly presented below, the Coding4Girls software does require the user to have a constant Internet connection. The game cannot be played offline.

## 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 functioning 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 post-its (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 innovative flexibility. After the brainstorming phase, the students will be given in a step-by-step fashion, specific activities (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 conditional structures.

Each challenge is structured in the following fashion:

1. 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 optional 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 their 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 have 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 divided by the other students. Figure 2 below illustrates the structure of a C4G course .

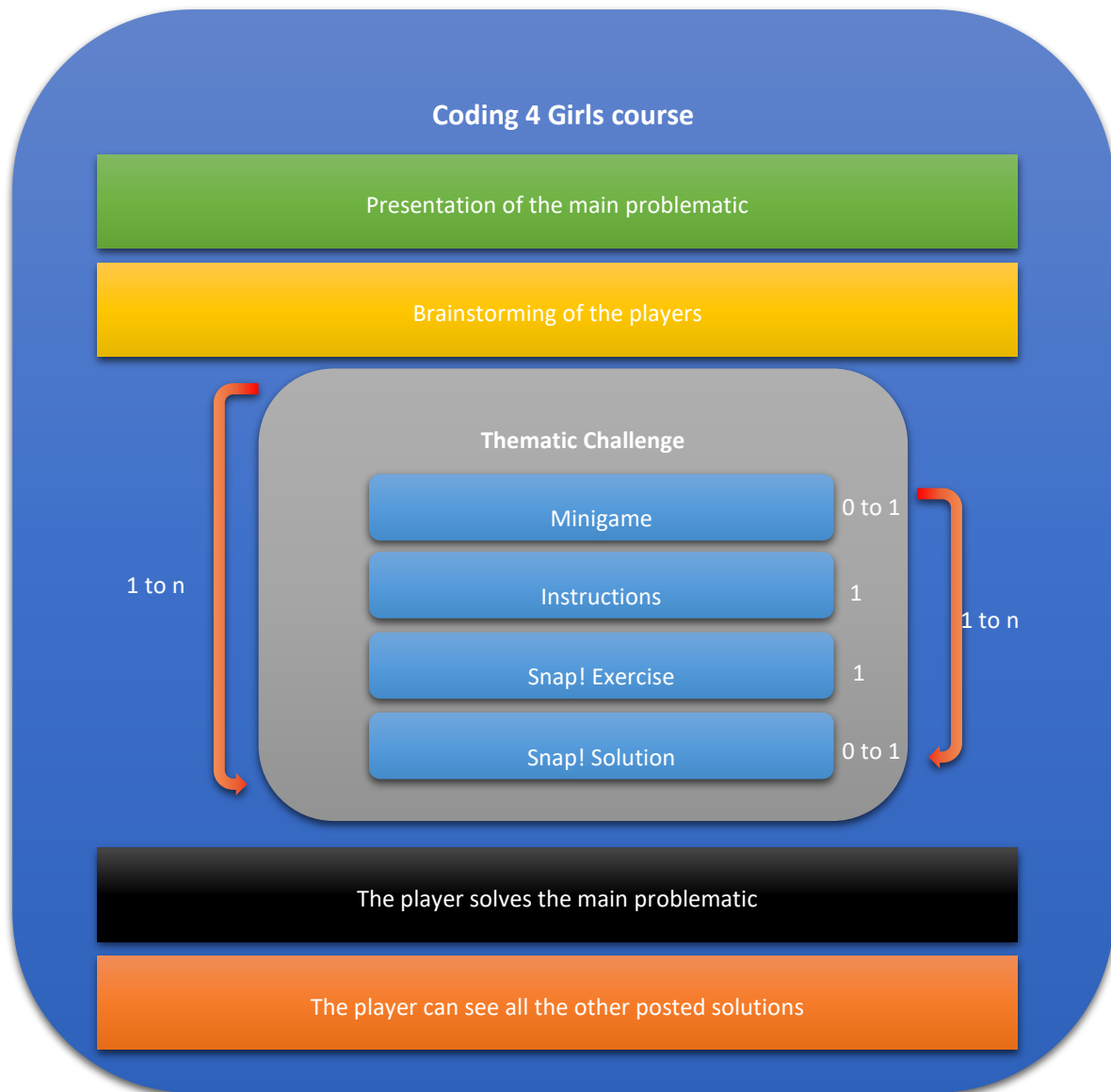


Figure 2: Structure of a C4G course

## THE C4G SOFTWARE

### AVAILABILITY

The student game is available through a downloadable executable. The game has been designed to work mainly on Windows but an OSX and a Linux version are also available. All the versions are downloadable at the following URLs:

- Windows: [https://ctl.e-ce.uth.gr/downloads/c4g/launcher/w64/c4g\\_win.zip](https://ctl.e-ce.uth.gr/downloads/c4g/launcher/w64/c4g_win.zip)
- Mac: [https://ctl.e-ce.uth.gr/downloads/c4g/launcher/m64/c4g\\_m64.zip](https://ctl.e-ce.uth.gr/downloads/c4g/launcher/m64/c4g_m64.zip)
- Linux: [https://ctl.e-ce.uth.gr/downloads/c4g/launcher/linux/c4g\\_linux.tar.gz](https://ctl.e-ce.uth.gr/downloads/c4g/launcher/linux/c4g_linux.tar.gz)

### THE GAME LAUNCHER

When you launch the game (the program is called C4G Launcher) you will be greeted with the patcher screen as presented in the figure below.

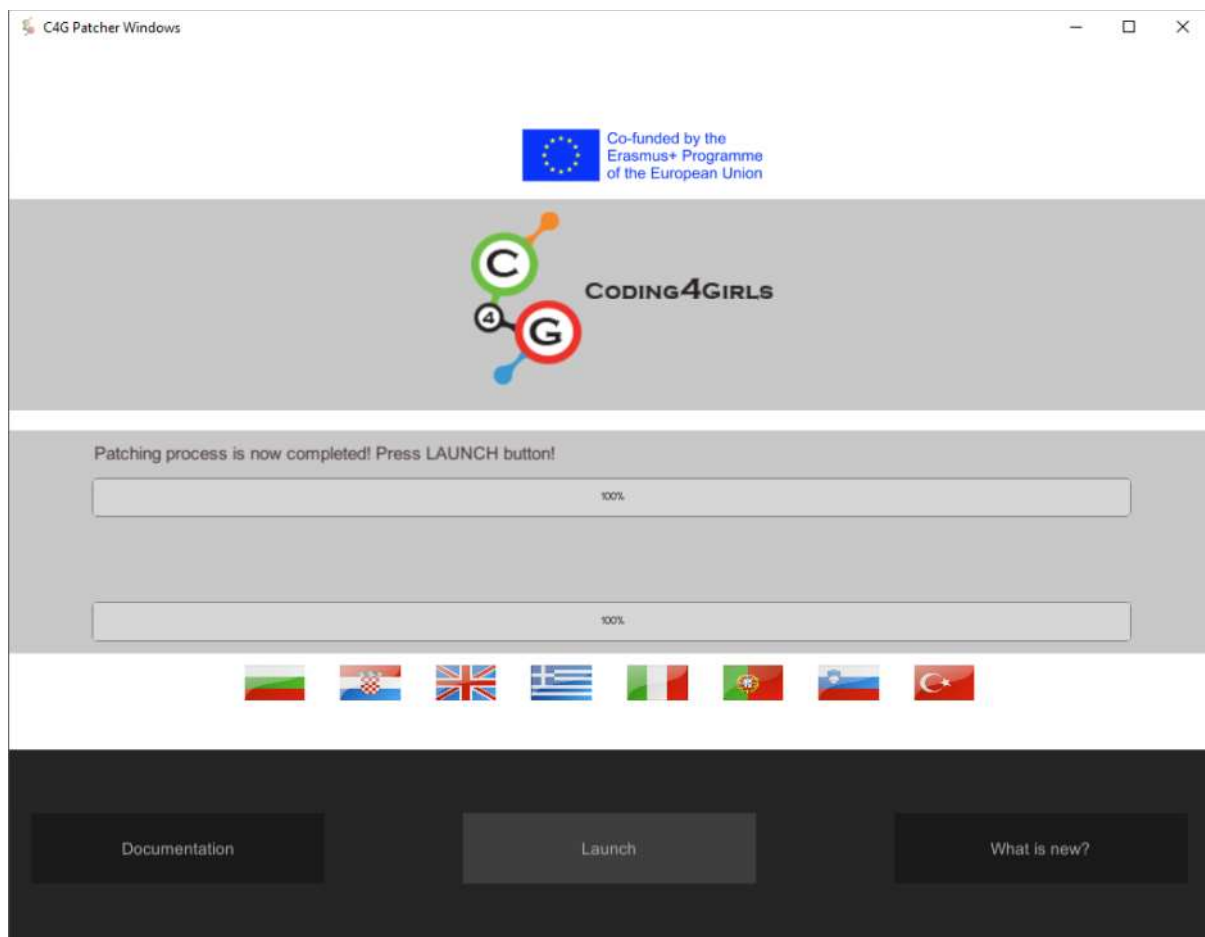


Figure 3: The C4G patcher

This patcher will automatically retrieve any update or patch to your current version of the software. Once the download and patching activity are finished you will be greeted with the message "Patching process is now completed, Press Launch button".

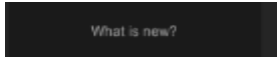


Note that the Linux version has no onboard patcher installed and will directly start with the game itself.

The patcher offers different options:

- You can change the language of the game to any of the C4G consortium's languages by clicking on the relevant flag 

- You can access the patchnotes to see what is new in the current version of the game by clicking on



- You can access the present manual by clicking on



- You can launch the game by clicking on



Once the Launch button clicked, the game configuration window appears.

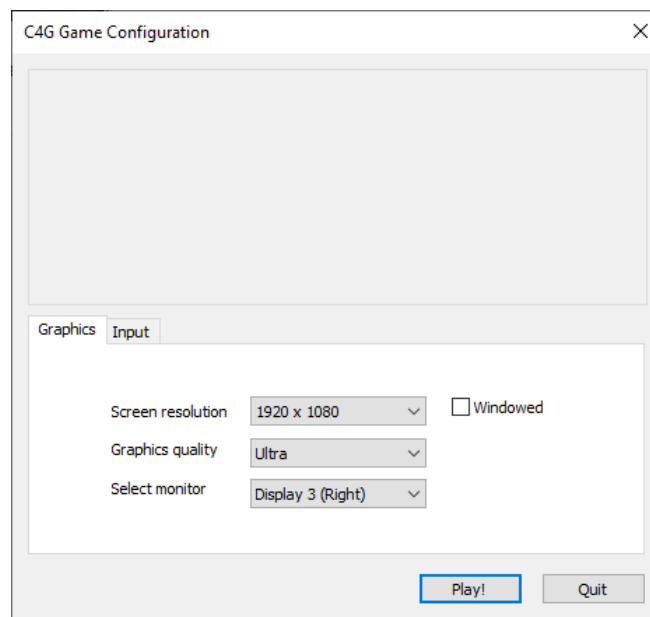


Figure 4: Game configuration window

This screen allows you to configure the global graphical quality of the game, the screen resolution and the monitor for display. **If you are running this game on an old or not very powerful computer, be sure to select the lowest graphical quality and a lower screen resolution for a better game experience. Selecting a low resolution, lower than your native screen resolution will also help immensely. You can always have a rough idea of the performances inside the game by activating the FPS counter in the Menu->Video option. Anything above 30FPS is very playable, anything under 15-20 is more difficult to play with.**

Clicking on the Play! Button launches the C4G game.

## 1. LOGGING IN AND SELECTING A COURSE

### THE INTRODUCTION SCENE

A quick introduction scene presenting the required disclaimer and the project and Erasmus+ logo is displayed for 4 seconds.



Figure 5: The introduction scene


### THE LOBBY


After the intro scene has finished, the player will be transferred to a minimalistic room called the lobby. It contains one terminal and a closed door. The terminal can be used to login or register in the platform.



Figure 6: The first room

On the top right of the screen an icon indicates the quality of the game connection to the C4G server:

This symbol  indicates the connection works well and the game can be played without an issue.

This symbol indicates  the connection with the server has been lost and you should either verify your internet connection or send an email to us in case the game's server is down.

You can move around your character (which is never displayed on screen) by using the W,A,S,D keys. Go close to the terminal and click on it. A short animation will play and the terminal top side will be focused.

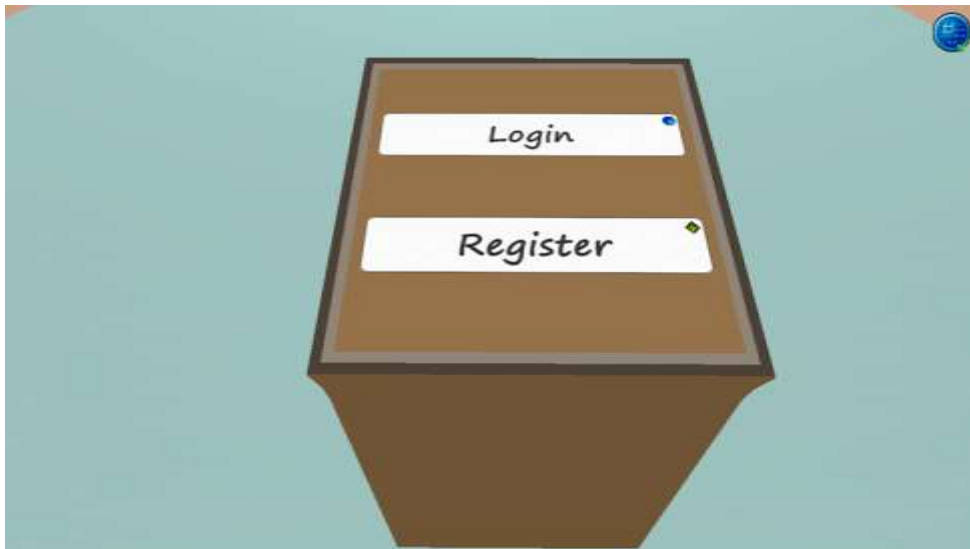


Figure 7: The terminal

Clicking on login, will show a panel asking for username and a password in order to authenticate the user.

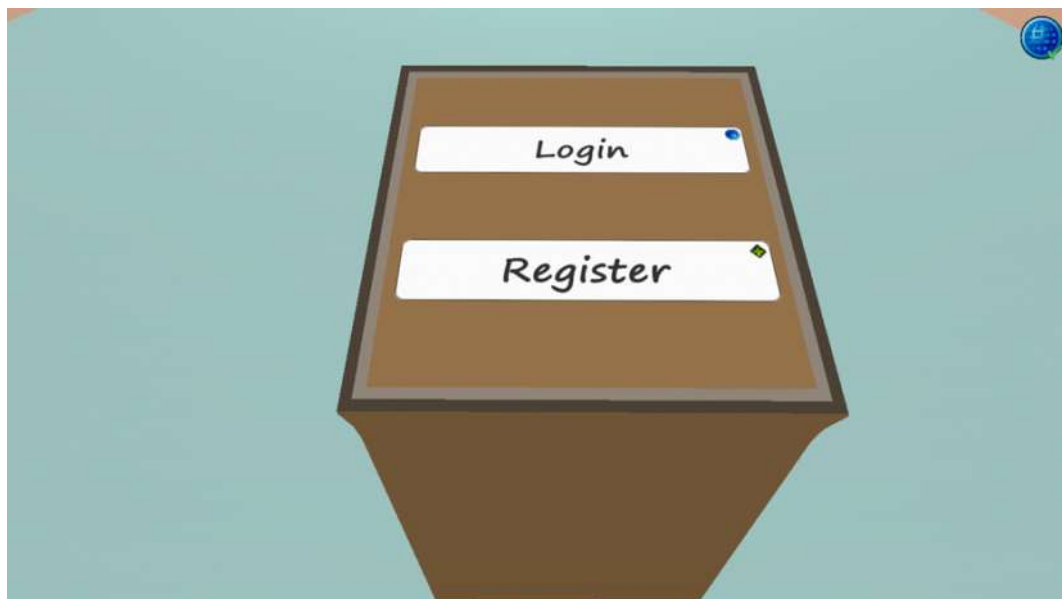


Figure 8: The login panel

New students (and only students!) can be created by clicking on the register button. The new user will need to specify his details, including a first and a last name, a username, and a password.

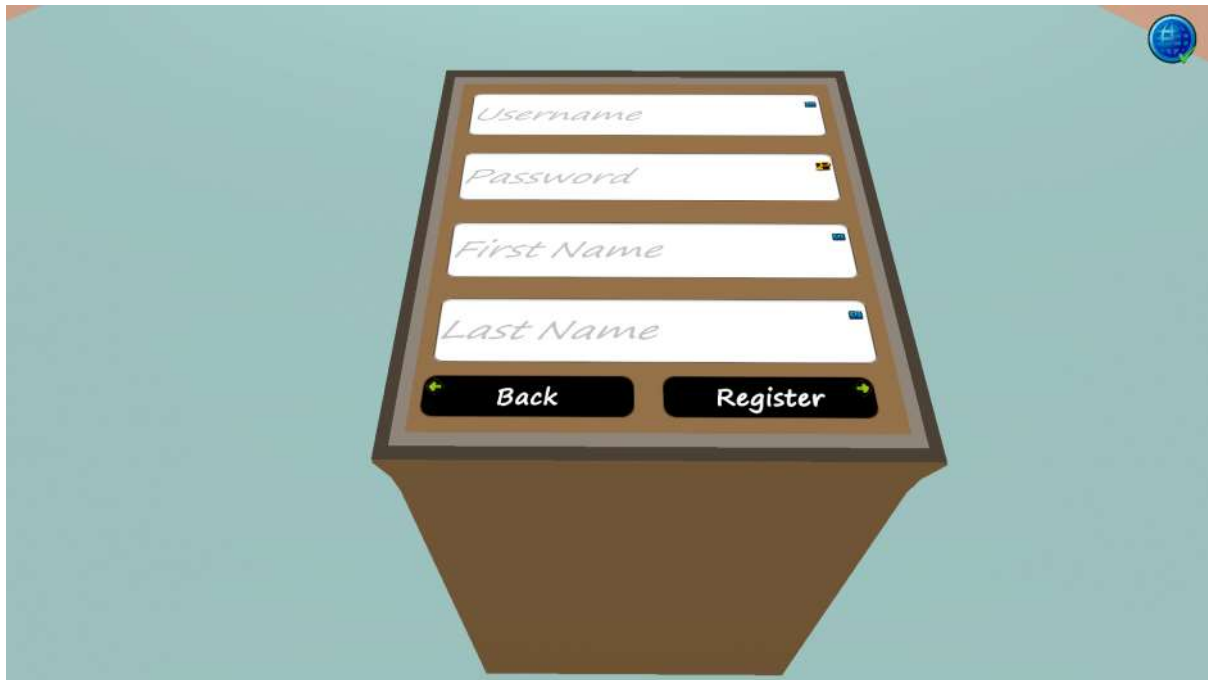
A 3D-rendered digital panel with a light blue background. The panel is brown with a white input area. It contains four text input fields labeled 'Username', 'Password', 'First Name', and 'Last Name' in a cursive font. Below the fields are two black buttons with white text: 'Back' on the left and 'Register' on the right. A small blue circular icon with a white 'H' is in the top right corner.

Figure 9: The register panel

If the authentication failed (usually when user typed the wrong password) the panel will change indicating that an error happened.

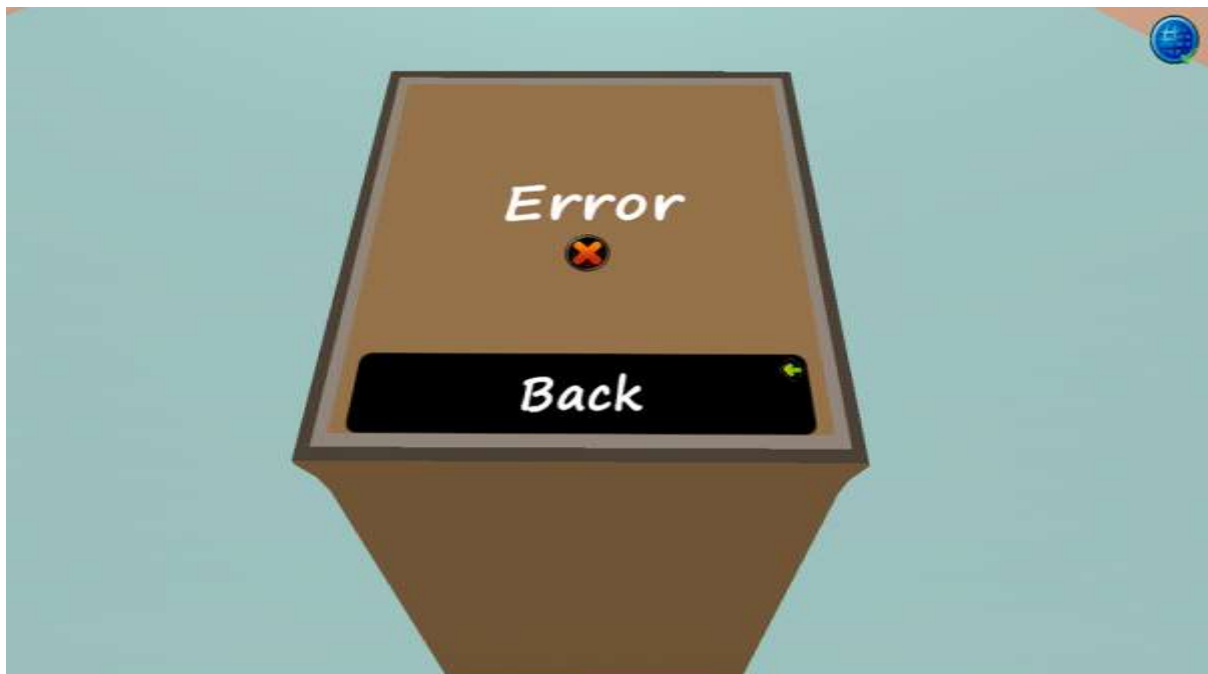


Figure 10: Unsuccessful authentication message


After a successful authentication, a confirmation message will appear on the panel.

In order to leave the terminal and move around the room press the right mouse click once.



Figure 11: The panel after successful login

After a couple of seconds (usually almost instantly), the door will slide open and the student can move to the next room. A new icon has appeared near the one indicating the quality of the connection with the server. This

coin icon  indicates how much game currency the user currently has. The currency can be gained by accessing challenges and can be spent in customizing the game environment or by playing a minigame at will. This gamification aspect is presented in more details a bit later in this manual.

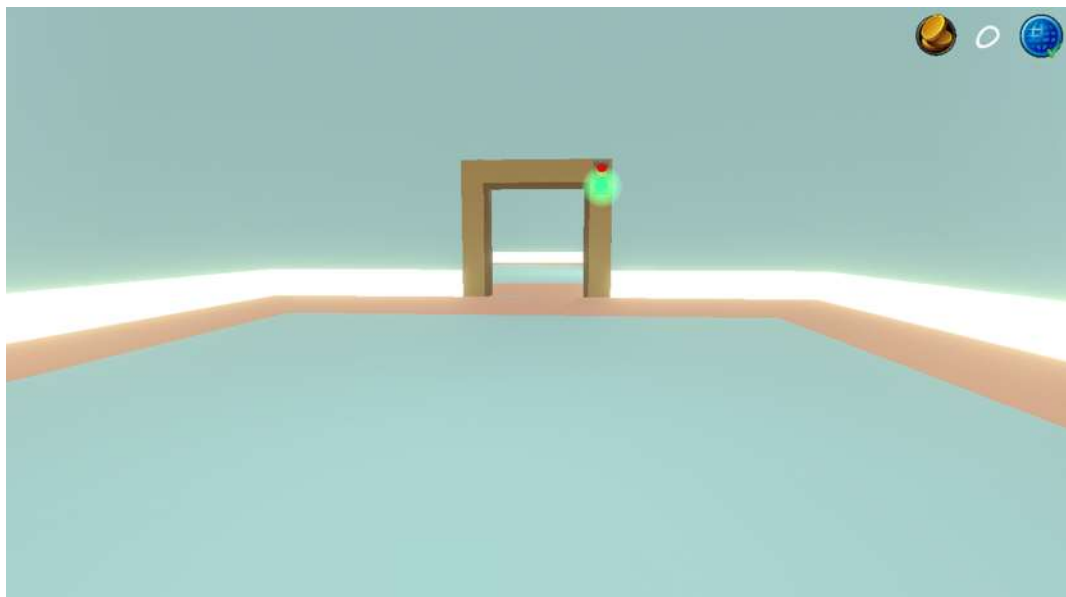


Figure 12: The unlocked door in the first room

Step through the door threshold to enter into the second room, where you can either select the course you want to play or join new courses. When stepping into the room you can see one purple portal in the middle and two terminals on either side of the portal.

If the portal is transparent, the connection with the game server is still ongoing and will require a few more additional seconds. The name on the top of the portal indicates the currently selected course, which is usually the last one selected.



Figure 13: The second room

The user can join a course using the terminal on the left of the portal and by entering the correct code.



Figure 14: The terminal to join a lobby

A right mouse click will help you leave the terminal.

The students can enter an already subscribed course by using the portal found on the wall. Next to the portal, there is also a terminal that is used for selecting the preferred course.



Figure 15: The lobby portal and the selection terminal

The user can change the destination (course) of the portal by clicking on the terminal. The terminal will cycle through all the courses available to the user.



Figure 16: The terminal that selects a course

When the selection has been made, the user can move closer to the portal and enter it by left clicking on it.

## 2. SELECTING A CHALLENGE IN A COURSE

The course structure appears in a new screen, with the course title followed by all the different course chapters, each associated with a number. You can always navigate back in the course selection room by clicking on the red X icon, located at the top corner.



Figure 17: The course panel

The student can select a challenge by clicking on it. The challenge details will appear on the bottom of the screen, with the challenge's name on the left, the challenge's associated minigame in the middle and a button to start the challenge.

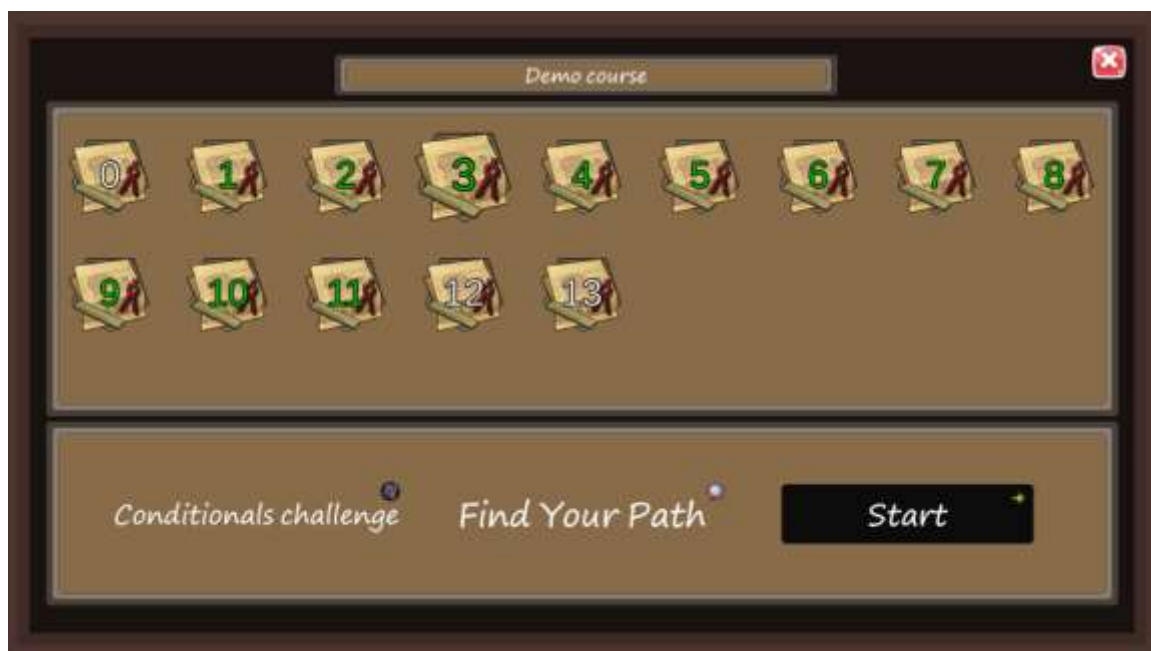


Figure 18: A selected challenge in course panel

By pressing **Start**, the challenge will launch.



In this panel, the challenge number 0 represents the general instructions given at the beginning of the game and opening it will take you to the global problem instructions, the related Snap! template followed by the brainstorm canvas.

The users have to access one challenge to unlock the next one. The challenges that are not unlocked yet are hidden from the panel. For a user starting a new course it's hence normal to begin with a panel that displays only 2 challenges at first, all the others being hidden.

A challenge number written in green represents the fact student has already played it and submitted a solution.

### 3. INSIDE A CHALLENGE

A C4G course is based on solving a global coding problem by giving to the students all the tools needed in a step-by-step fashion. A course always starts with some instructions, an overview of the problem in Snap!, a brainstorming phase for all the students, a certain number of coding activities illustrated by minigames and a zone where students can compare their own solutions to that their comrades came up with.

#### INSTRUCTIONS


Instructions will appear in the game as a multimedia HTML page. After the students have read or watched everything presented they can continue by clicking on the right arrow in the bottom of the screen. 



Figure 19: Teacher's instructions

#### SNAP ACTIVITIES

All the coding activities appear in game as Snap! canvases very similar to the one found by starting Snap! at this URL : <https://snap.berkeley.edu/snap/snap.html>.

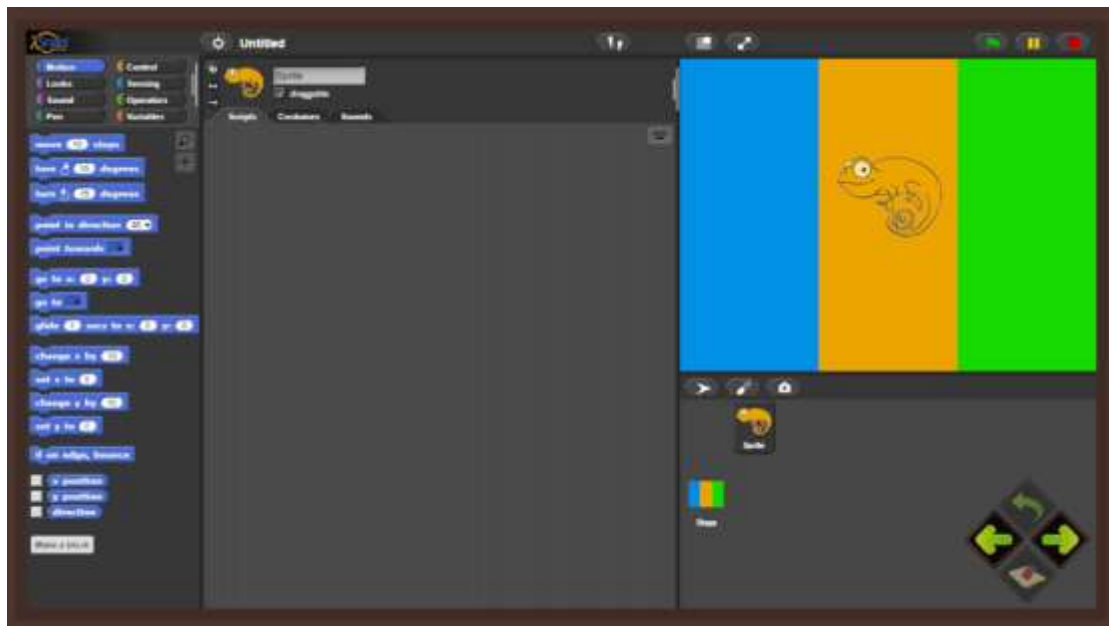





Figure 20: Snap! canvas for the global problem

The players can change the language of the interface, load assets or other materials and use coding bricks exactly as they would do it in a classical course setting. The standard Snap! interface has been only slightly modified from the original ones and the C4G coding experience is exactly the same as the standard Snap! coding experience.

If the context allows it, they can also (see figure 21) go back to read the instructions by pressing the back button , or cancel their modifications of the provided Snap! canvas with the cancel button  or remove every modification they have done on this canvas and go back to the basic version provided by the course by clicking the erase button .

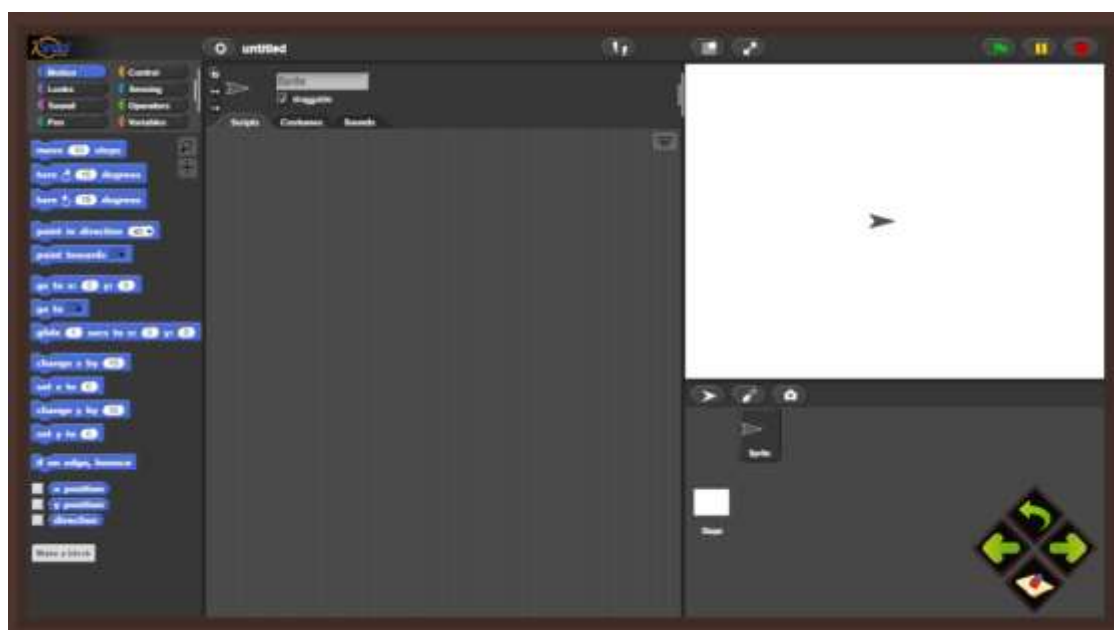


Figure 21: Snap! canvas for a coding activity

## BRAINSTORM

The brainstorm zone appears a space where the users can place and manage multimedia post-its. All the students involved in the course can write something there and everyone will see the contributions of all the stakeholders.

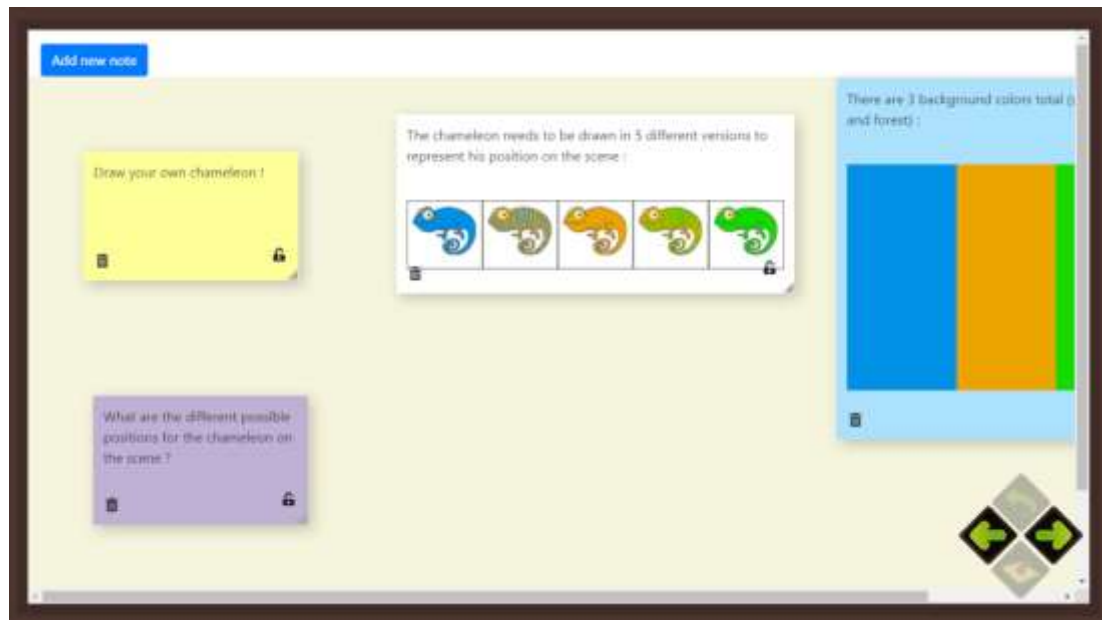


Figure 22: Brainstorm zone

## ACCESSING OTHER STUDENTS' SOLUTIONS

Once a player has completed a course by writing their coding solution to the overarching issue, they are able to compare what they did to other students' solution. They will not see the name of the other users, only the solution they have provided.



Figure 23: anonymized solution database

#### 4. THE MINIGAMES

When the users enter a challenge attached to a minigame, they will be teleported to the location where the minigame takes place. There are many locations for the different minigames varying from sandy beaches to snowy mountains.

An animated loading screen will appear each time the player changes location.



Figure 24: The loading screen

This section will describe the different available minigames, their mechanics and their locations.

## DICE GAME

The dice game takes place in dark woods. In front of the players is an abandoned cabin that they need to enter.



Figure 25: The abandoned cabin

The interactive part of the minigame is inside the cabin, on the chair across the door. It is a wooden board with 2 dice on it. By clicking on it, the minigame begins.

Both the player and the AI roll the dices in turn. The one with the biggest sum of the two dices wins the round.



Figure 26: The inside of the cabin





Figure 27: The wooden board in the Dice Game



Figure 28: Dice game in action

At the top, on the left side with the red flag you can see the score of student for this turn and on the right the score of the AI. The medal icon shows the total score. The player gets one point for winning and loses one when defeated.

## MULTIPLE CHOICE QUESTIONS

The multiple choice quiz is set in a field on the side of a cliff. The users are free to roam at the beginning and in order to start the minigame, they need to enter the red circle on the ground located next to the rock pile highlighted with a red circle in the screenshots below.



Figure 29: The location to start the quiz activity



Figure 30: The view when the quiz has started.



Questions appear on the top of the cliff and each field will contain of the four possible answers, alongside with the images, if any.

The exclamation mark at the bottom provides hints to the student, if specified.

The medal icon represents the current score of the student, based on the correct answers and the clock countdowns the seconds to end this challenge.

The arrow button at the bottom allows you to go to the next question.

If the timer runs out before you are finished answering the quiz, the following message will appear:



Figure 31: The timer ran out

This will happen in any minigame where a timer is involved.

## MATCH3

The Match3 game takes place in snowy, elevated region that the users are able to roam freely.



Figure 32: The initial player position in the Match3 game

The game can be started by locating and clicking on the red zone at the bottom of the hill with strange stones on it.



Figure 33: The location to start the Match3 game

The Match3 minigame is based on a typical Match3 activity where you have to drag and drop adjacent squares in order to make them disappear if they form a group of 3 tiles or more of the same type.



Figure 34: The Match3 minigame

## INVENTORY

The inventory game takes place in the fields, next to a small forest. There are colorful spheres floating around the area and the player needs to collect them, according to the instructions delivered by the teacher on the wooden sign.



Figure 35: A scene from the inventory minigame

The students can stack orbs of the same color, so they occupy less space in the inventory and they can also combine different orbs together to produce new ones.

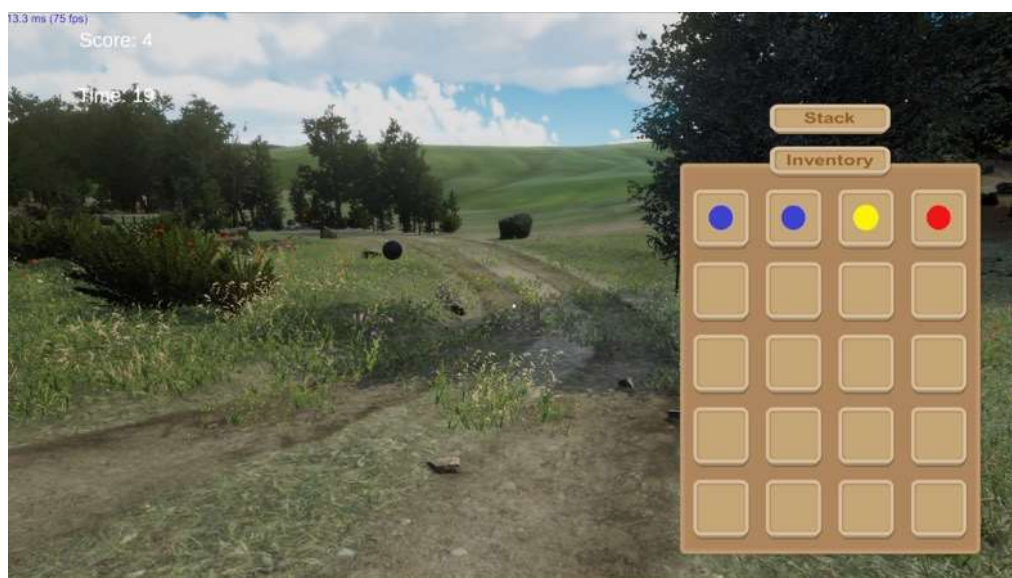


Figure 36: Gathered orbs, unstacked



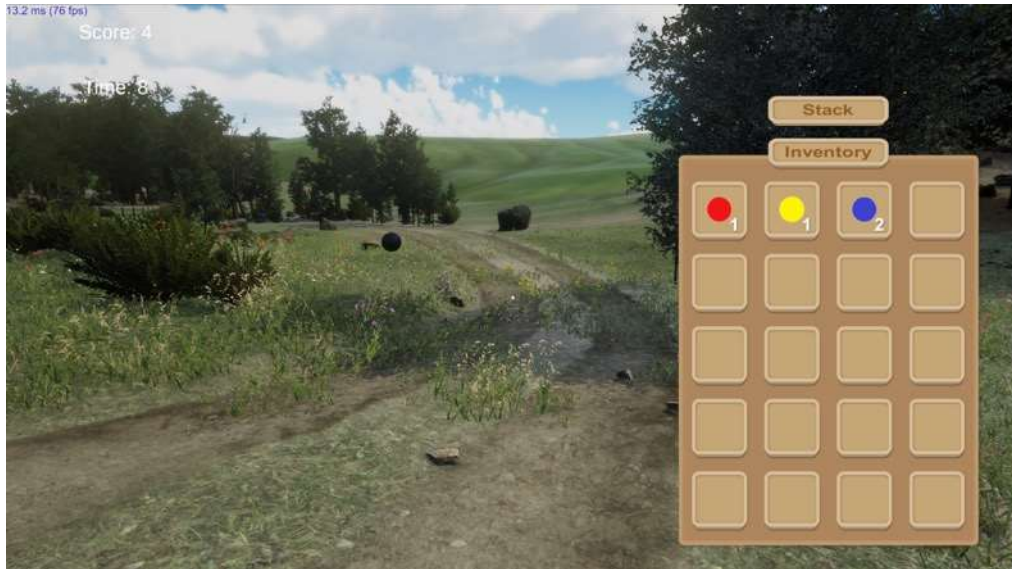


Figure 37: Gathered orbs, stacked

For example, a teacher can ask their students to have one purple orb (the combination of a blue and a red one) and three red orbs occupying only two slots in the inventory. In order to do so the students will need to pick up and stack the orbs.



Figure 38: A purple and a yellow orb in the inventory

## STEPPING GAME

This minigame takes place in one beach of the island at sunset. The player needs to answer the question written on the big rock in the center of the beach by stepping on the letter spelling the correct answer. A small radar view is available on the top down of the screen in order for the user to better see where they are what stone to step on.

When the user steps on one of the stone with a letter forming the answer, the letter will be highlighted in green and appear on the second big rock next to the one where the question is written. If the user steps on a wrong stone it will be highlighted red and they will need to start again the spelling of the answer.



Figure 39: The Stepping game



Figure 40: Stepping game after a letter has been pressed

## SOUND GAME

The sound game takes place in a tropical forest. The player needs to find the 5 scattered panels and solve a puzzle based on sounds. Below you can see one of the panels



Figure 41: A panel in the sound game

Each column of the panel represents one sound and each row represents the tone of the sound. The buttons on top are for the high tones and the ones on the bottom for the low. When the user is near a panel, a combination of 3 possible sounds will play (one low pitched bird call, one middle pitched bird call and one high pitched bird call). The combination of bird calls is repeated in a loop, each repetition separated by the others by a 3 seconds silence.

The player will need to pay great attention to the order in which the bird calls are played and reproduce it on the panel accordingly. For example on the first panel, the one illustrated in figure 36, the user can hear first a low pitched call and then a high pitched one. The correct answer is hence to press the low button on the left of the panel and the high button on the right.





Figure 42: The answer on the left is incorrect (black) and the upper button is selected on the right

To select a certain button, the user simply has to point the mouse cursor at it and the button will be highlighted in yellow. If the pressed button is the correct one then it will appear green, otherwise it will appear as black.

Once the entire solution is given, the whole panel will turn green and it will be unusable.



Figure 43: A solved sound game panel



## SNAKE GAME

The game is set near a lake and following the arrow sign leads you to a red circle you need to step into.



Figure 44: The snake's game landscape

The snake game takes place in the water. You control the snake using the 4 keyboard arrow and try to survive as long as possible. The snake increases in size by eating the bright green dots but can die if they circle on their own tail, hit the border of the screen or eat a red dot. Each eaten green dots adds one point to the score.



Figure 45: The snake game

## PUZZLE GAME

The puzzle game takes place in a rocky area of the map. The player needs to find the scattered panels in the area and solve the puzzles by creating a line starting from the white circle of the panel going to the red square.

If there are black squares, the line must pass through all of them.



Figure 46: The scenery of the puzzle game



Figure 47: Across the bridge in puzzle game





Figure 48: A panel in the puzzle game

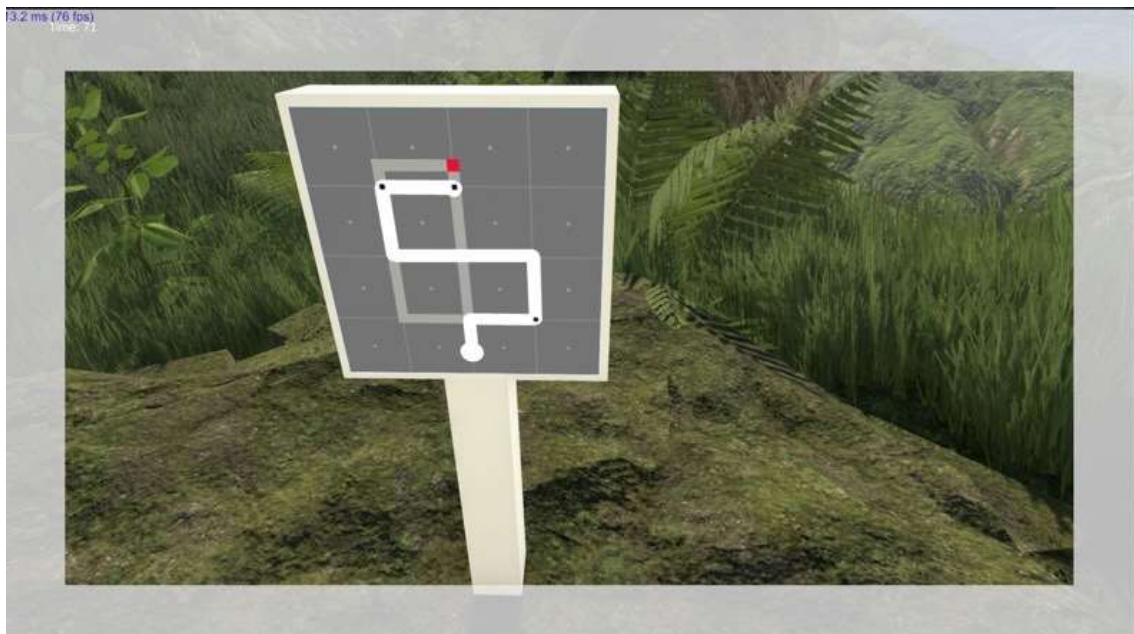


Figure 49: A closeup of a panel in sound game

## PATTERN MATCHING GAME

The pattern matching game takes place on the banks of a river. Once the player steps in the red circle, the game starts.



Figure 50: The scenery of the pattern matching game

Boxes that have numbers (all kind of numbers) written on their side will start floating on the river and the current will make them head towards the sea. The goal of the game is for the player to pick them up in order to complete a certain math operation.

A big blue square on opposite side of the player indicates the target result. Three blue boxes below define the available operation that will be applied to number boxes that will be added in them.

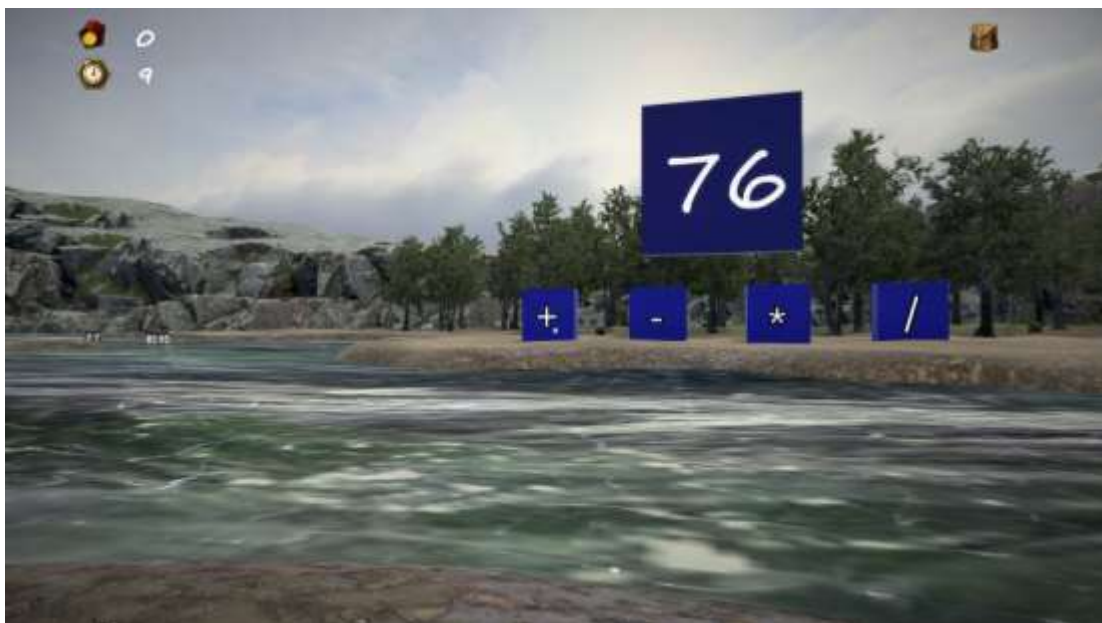


Figure 51: The target and the operation boxes in the pattern matching game

In other terms, the players need to use the floating numbers and available operations to reach the target number.

In the example displayed on figure 46, the player could pick up a box labeled 26, put it in the + operation box, pick another floating box labeled 30, put it also in the plus to increase their score. That would represent  $46+30=76$ .

But they could also pick a box labeled 38, put it in the \* operation box and then pick a box labeled 2 and put in in the \* operation box. This would net them the same amount of points, as  $36*2=76$ .

The target numbers and the floating numbers are generated randomly but in a way that makes sure there is always a possible combination to reach the target number.

The available operations depend on the teacher and can be one of the following:

- Basic operations (+, -, \*, /)
- Advanced operation (power, square root, modulo)
- Trigonometry (Cos, Sin, Tan)
- Booleans (AND, OR, XOR)

A box that is pointed by the mouse and ready to be picked up by a click is highlighted with a white halo.



Figure 52: The gameplay of the pattern matching game

## 5. UI AND PAUSE MENU

### PAUSE MENU

#### GENERAL PANEL



Figure 53: Pause menu, general panel

Here, the student can change the language of the game, return to the main entrance (after the login procedure) or quit completely the application.

#### AUDIO PANEL



Figure 54 : Pause menu, audio panel



The student can adjust the sound of various aspects of the game using this panel. The main volume slider adjusts the volume of all the sounds, regardless of the other two sliders. Changing the music volume slider adjusts the volume of the music in the lobby section and inside the minigames. Finally, the SFX volume slider, controls the volume of special effects sounds, found mostly in mini-games.

## VIDEO PANEL

In the video panel, the student can change the quality of the graphics in the game



Figure 55 : Pause menu, video panel

The graphical quality slider affects the quality of all the 3D models that appear in the game and all the visual effects of the game.

The anti-aliasing makes the graphics have smoother edges. Higher values mean better graphics quality but requires high performance graphics card. By default the value is set to none

Mouse sensitivity affects the speed of the mouse. Higher values make the mouse cursor move faster. Be careful with adjusting this parameter as it directly affects the user experience.

Enabling the FPS counter will show you the current frames per second on the bottom right corner of the screen.

## GAME RULES

The game rules panel contains useful information on how to play the current minigame.



Figure 56 : Pause menu, game rules panel



## PERSONALISATION PANEL

On this panel, students can change the decoration of the lobby rooms and the music that plays when they are there. The sliders can be adjusted to see and listen the current selection pool and by pressing the little icon next to them, they will be saved, reducing their coins by one each time.

Also, there is the possibility to play the available minigames in free-mode (without time or score restrictions), again reducing their coins by one.



Figure 57 : Pause menu, personalization panel

## TABLE OF FIGURES

Figure 1: C4G project methodology.....	3
Figure 2: Structure of a C4G course.....	7
Figure 3: The C4G patcher .....	8
Figure 4: Game configuration window .....	9
Figure 5: The introduction scene .....	10
Figure 6: The first room .....	10
Figure 7: The terminal .....	11
Figure 8: The login panel .....	11
Figure 9: The register panel .....	12
Figure 10: Unsuccessful authentication message.....	12
Figure 11: The panel after successful login.....	13
Figure 12: The unlocked door in the first room.....	13
Figure 13: The second room .....	14
Figure 14: The terminal to join a lobby.....	14
Figure 15: The lobby portal and the selection terminal .....	15
Figure 16: The terminal that selects a course.....	15
Figure 17: The course panel.....	16
Figure 18: A selected challenge in course panel.....	16
Figure 19: Teacher's instructions.....	18
Figure 20: Snap! canvas for the global problem .....	19
Figure 21: Snap! canvas for a coding activity .....	19
Figure 22: Brainstorm zone .....	20
Figure 23: anonymized solution database .....	20
Figure 24: The loading screen.....	21
Figure 25: The abandoned cabin .....	22
Figure 26: The inside of the cabin.....	22
Figure 27: The wooden board in the Dice Game .....	23
Figure 28: Dice game in action .....	23

Figure 29: The location to start the quiz activity .....	24
Figure 30: The view when the quiz has started. ....	24
Figure 31: The timer ran out.....	25
Figure 32: The initial player position in the Match3 game .....	26
Figure 33: The location to start the Match3 game .....	26
Figure 34: The Match3 minigame.....	27
Figure 35: A scene from the inventory minigame .....	28
Figure 36: Gathered orbs, unstacked .....	28
Figure 37: Gathered orbs, stacked.....	29
Figure 38: A purple and a yellow orb in the inventory .....	29
Figure 39: The Stepping game .....	30
Figure 40: Stepping game after a letter has been pressed .....	30
Figure 41: A panel in the sound game .....	31
Figure 42: The answer on the left is incorrect (black) and the upper button is selected on the right .....	32
Figure 43: A solved sound game panel .....	32
Figure 44: The snake's game landscape .....	33
Figure 45: The snake game .....	33
Figure 46: The scenery of the puzzle game .....	34
Figure 47: Across the bridge in puzzle game .....	34
Figure 48: A panel in the puzzle game.....	35
Figure 49: A closeup of a panel in sound game .....	35
Figure 50: The scenery of the pattern matching game .....	36
Figure 51: The target and the operation boxes in the pattern matching game .....	36
Figure 52: The gameplay of the pattern matching game .....	37
Figure 53: Pause menu, general panel .....	38
Figure 54 : Pause menu, audio panel.....	38
Figure 55 : Pause menu, video panel.....	39
Figure 56 : Pause menu, game rules panel.....	40
Figure 57 : Pause menu, personalization panel.....	41

