



Universidade do Porto

FEUP Faculdade de Engenharia



Creatively learning programming in Digital Media Education

G2T

António Coelho
acoelho@fe.up.pt

Outline

Motivation

Fundamental concepts

Gamification of the learning process

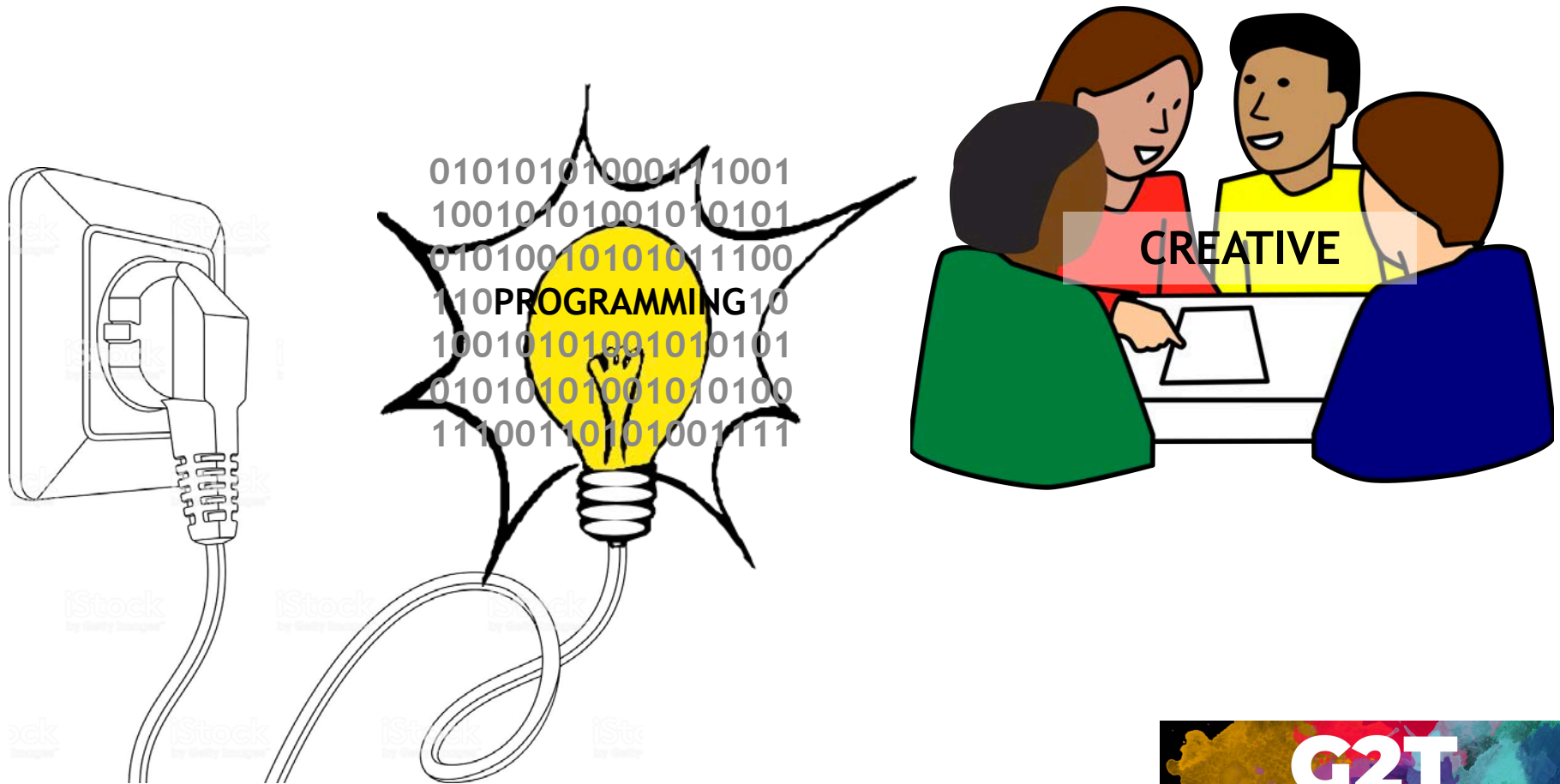
Creative Programming Matrix

Results

Conclusions and future work



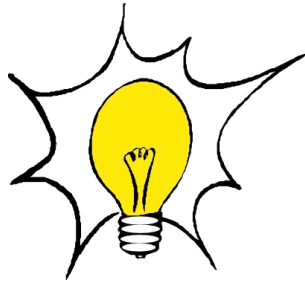
The motivation for this work is focused on Digital Media students.



Fundamental concepts:



B-Learning - *Blended-learning*



PBL - *Problem-based Learning*



Gamification



Gamification of the learning process

Following the “6 Steps to Gamification” [Werbach]

1. Define business objectives

- Acquire computational thinking skills;
- Autonomy to develop low/medium complexity programs.

2. Delineate target behaviors

- Read the learning contents before the class;
- Practice programming regularly;
- Promote adaptation to the learner’s progression;
- Assiduity and punctuality.

3. Describe your players

- Heterogeneous profile & creativity.



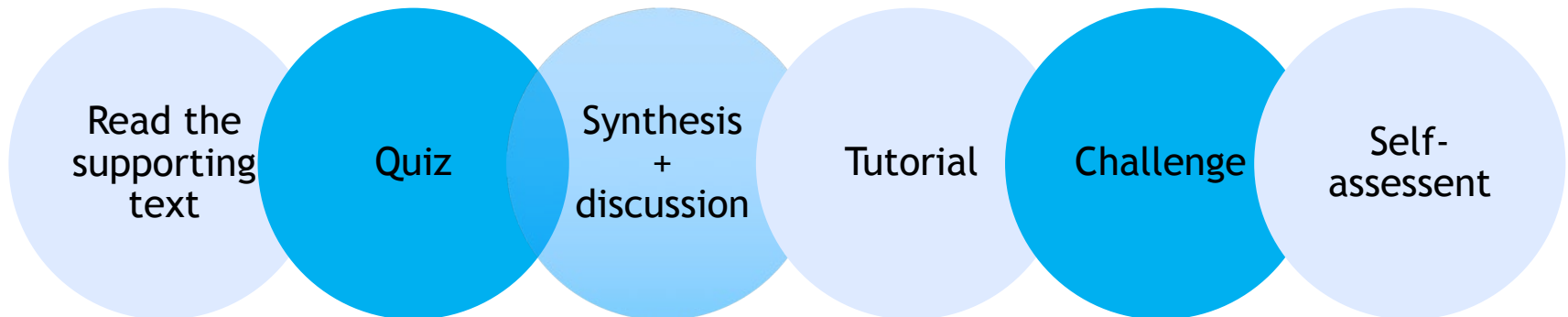
Gamification of the learning process

4. Devise your activity loops

- Semester

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Onboarding		Scaffolding										Endgame	

- Week

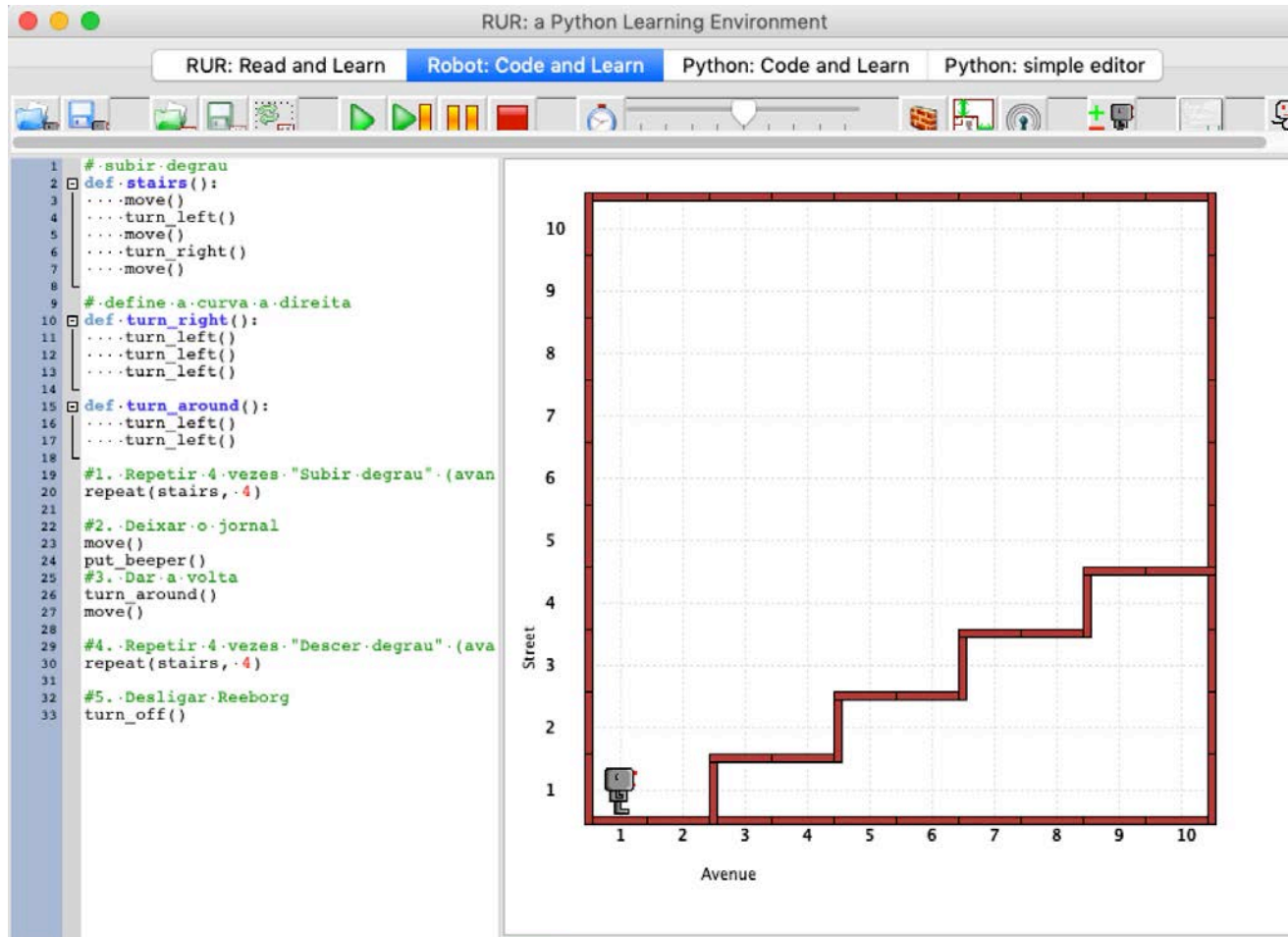


Gamification of the learning process

5. Don't forget the fun (engagement)



Onboarding: Introduction to computational thinking with RUR-PLE.



The screenshot displays the RUR-PLE Python Learning Environment interface. The window title is "RUR: a Python Learning Environment". The interface includes a menu bar with "RUR: Read and Learn", "Robot: Code and Learn", "Python: Code and Learn", and "Python: simple editor". Below the menu bar is a toolbar with various icons for file operations, execution, and help. The main area is split into two panes. The left pane shows Python code for a robot named "Reeborg" in a grid world. The code defines functions for moving, turning, and repeating actions. The right pane shows a 10x10 grid world with a robot at the bottom-left corner (1,1). The robot's path is shown as a red line, starting at (1,1) and moving up to (1,2), then right to (2,2), up to (2,3), right to (3,3), up to (3,4), right to (4,4), up to (4,5), right to (5,5), up to (5,6), right to (6,6), up to (6,7), right to (7,7), up to (7,8), right to (8,8), up to (8,9), right to (9,9), and finally up to (9,10).

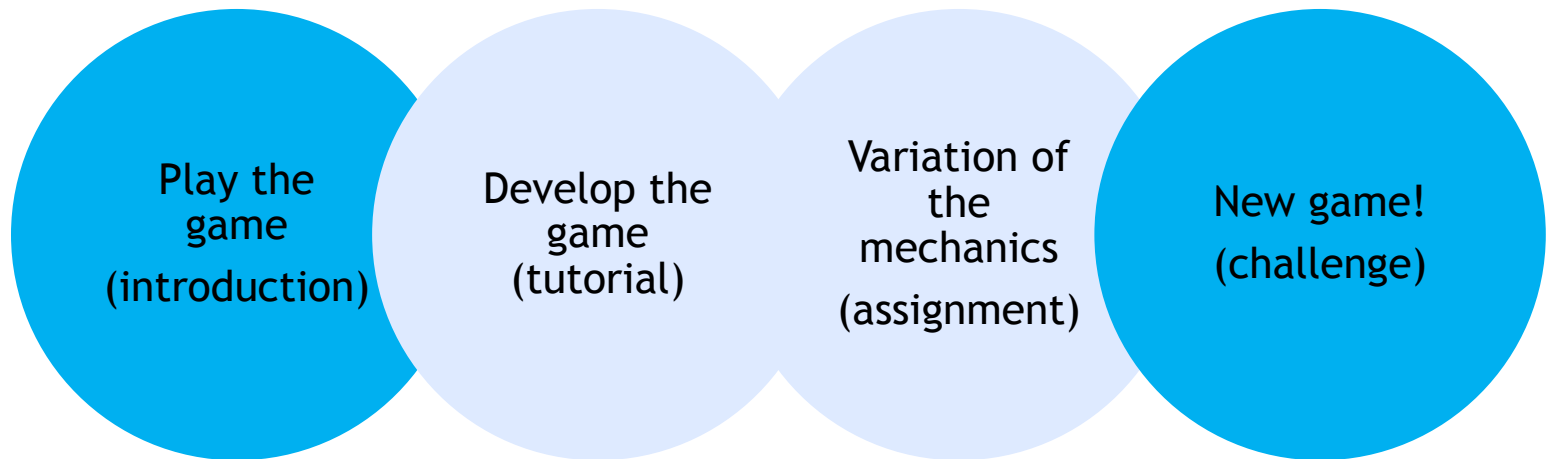
```
1 #.subir-degrau
2 def stairs():
3     ...move()
4     ...turn_left()
5     ...move()
6     ...turn_right()
7     ...move()
8
9 #.define-a-curva-a-direita
10 def turn_right():
11     ...turn_left()
12     ...turn_left()
13     ...turn_left()
14
15 def turn_around():
16     ...turn_left()
17     ...turn_left()
18
19 #1. Repetir 4 vezes "Subir-degrau" (avan
20 repeat(stairs, 4)
21
22 #2. Deixar o jornal
23 move()
24 put_beeper()
25 #3. Dar a volta
26 turn_around()
27 move()
28
29 #4. Repetir 4 vezes "Descer-degrau" (ava
30 repeat(stairs, 4)
31
32 #5. Desligar Reeborg
33 turn_off()
```

RUR-PLE



Scaffolding: PBL approach.

Game-based learning activities:



Mapping of programmatic contents to games

1	Introduction to the programming language: Development environment (IDE) and command line execution; lexicon and syntax; variables and primitive data types; expressions. A first game. Conditional control structures. <ul style="list-style-type: none">• Flip the coin (Heads or Tails)• Game "Rock, paper, scissors"
2	Recursion and iteration: loops in programming and the game loop. Procedural abstraction: functions. Definition of tests and detection of semantic errors (debugging). <ul style="list-style-type: none">• Game to guess the number from 1 to 100• Blackjack
3	Data structures: Vectors. <ul style="list-style-type: none">• Mastermind• Slot machine
4	Data structures: Strings. <ul style="list-style-type: none">• Hangman• Word puzzle
5, 6, 7, 8	Data structures: lists, queues, stacks and tables. <ul style="list-style-type: none">• Card game - lists• Car racing game - queues• Towers of Hanoi - stacks• Adventure text game - tables
9	Anatomy of a Game Engine. Graphics and visualization. <ul style="list-style-type: none">• Pong
10	Data abstraction: Classes of objects. <ul style="list-style-type: none">• Snake

Creative Programming Matrix

Creative Programming Matrix																			
				tutorial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Syllabus	Bloom	N	Avg	Max \ week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1. Fundamentals of Programming																			
1.1. Lexicon and syntax of a programming language	3	14	2,64	3	1	2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.2. Primitive data types and variables	3	14	2,64	3	1	2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.3. Expressions	3	13	2,77	3		2	2	2	3	3	3	3	3	3	3	3	3	3	3
1.4. Basic I/O including files	3	14	2,43	3	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3
1.5. Conditional Control Structures	4	14	3,07	4	1	2	2	2	3	3	3	3	4	4	4	4	4	4	4
1.6. Functions and parameter passing	4	12	3,17	4			1	1	3	3	3	3	4	4	4	4	4	4	4
1.7. Recursion and iteration: conditional and counted loops	4	12	3,17	4			1	1	3	3	3	3	4	4	4	4	4	4	4
1.8. Program testing and debugging	4	11	3,36	4				1	3	3	3	3	4	4	4	4	4	4	4
2. Data Structures																			
2.1. Vectors	3	13	2,31	3	1	1	1	1	2	3		3	3	3	3	3	3	3	3
2.2. Character Strings	3	14	1,71	3	1	1	1	1	1	1	2	3	2	2	2	3	2	2	2
2.3. Lists, Stacks, Queues and Tables	3	6	2,50	3									1	2	3	3	3	3	3
2.4. Data abstractions and their implementation	3	6	2,00	3									1	2	2	2	2	2	3
2.5. Selecting the most appropriate data structure	2	6	1,33	2									1	1	1	1	1	2	2
3. Software Project																			
3.1. Basic concepts;	3	12	2,42	3			1	2	2	2	2	2	3	3	3	3	3	3	3
3.2. Structured programming;	3	14	2,57	3	1	1	2	2	3	3	3	3	3	3	3	3	3	3	3
3.3. Object oriented programming;	3	2	2,50	3														2	3
3.4. Abstraction-based programming: libraries and frameworks	3	4	2,25	3										1	2			3	3
4. Graphics and Visualization																			
4.1. Image representation: vector and raster model;	3	4	2,25	3										1	2			3	3
4.2. Color and color models	3	4	2,25	3										1	2			3	3
4.3. Development of graphic applications	3	4	2,25	3										1	2			3	3
4.4. Graphic User Interfaces (GUI)	3	4	1,75	3										1	1			2	3
5. Fundamentals and Design of Digital Games																			
5.1. Concept of game and human factors;	3	14	2,79	3	1	2	3	3	3	3	3	3	3	3	3	3	3	3	3
5.2. Interactive and non-linear narratives;	3	1	3,00	3														3	
5.3. Game mechanics;	4	14	2,86	4	1	2	2	2	3	3	3	3	3	3	3	4	4	4	4
5.4. Design of game worlds (levels);	5	4	2,25	4										1		2	2	2	4
6. Programming of Digital Games																			
6.1. Architecture of a game engine	2	4	1,50	2										1	1			2	2
6.2. Graphic programming	3	4	2,25	3										1	2			3	3
6.3. Audio Programming	2	3	1,67	2										1				2	2
6.4. Game Logic	4	13	2,08	4	1	1	1	1	1	2	2	2	2	3	3			4	4
6.5. Input and event management	2	13	1,31	2	1	1	1	1	1	1	1	1	1	2	2			2	2
6.6. Physics and collision detection	2	3	1,67	2										1				2	2
6.7. Artificial Intelligence	2	2	1,50	2														1	2



Creative Programming Matrix

Scale:
 1 - Remember
 2 - Understand
 3 - Apply
 4 - Analyze
 5 - Evaluate

Syllabus	Bloom	N	Avg	Max \ week						
					tutorial	1	2	3	4	5
1. Fundamentals of Programming										
1.1. Lexicon and syntax of a programming language	3	14	2,64	3	1	2	2	2	2	3
1.2. Primitive data types and variables	3	14	2,64	3	1	2	2	2	2	3
1.3. Expressions	3	13	2,77	3		2	2	2	2	3
1.4. Basic I/O including files	3	14	2,43	3	1	1	1	1	1	3
1.5. Conditional Control Structures	4	14	3,07	4	1	2	2	2	2	3
1.6. Functions and parameter passing	4	12	3,17	4			1	1	1	3
1.7. Recursion and iteration: conditional and counted loops	4	12	3,17	4			1	1	1	3
1.8. Program testing and debugging	4	14	3,26	4				1	1	3
2. Data Structures										
2.1. Vectors	3	13	2,31	3	1	1	1	1	1	2
2.2. Character Strings	3	14	1,71	3	1	1	1	1	1	1
2.3. Lists, Stacks, Queues and Tables	3	6	2,50	3						
2.4. Data abstractions and their implementation	3	6	2,00	3						
2.5. Selecting the most appropriate data structure	2	6	1,33	2						

https://www.fe.up.pt/~acoelho/creative_programming_matrix.xlsx



Gamification of the learning process

6. Deploy the appropriate tools(2/3)

Levels

- **Apprentice (0 points)**
You have just started...
- **Pro (500 points)**
You can **eliminate your worst** practical assignment.
- **Expert(1000 points)**
Also **Pro** and...
You can **eliminate your 2nd worst** practical assignment replacing it by a project.



Gamification of the learning process

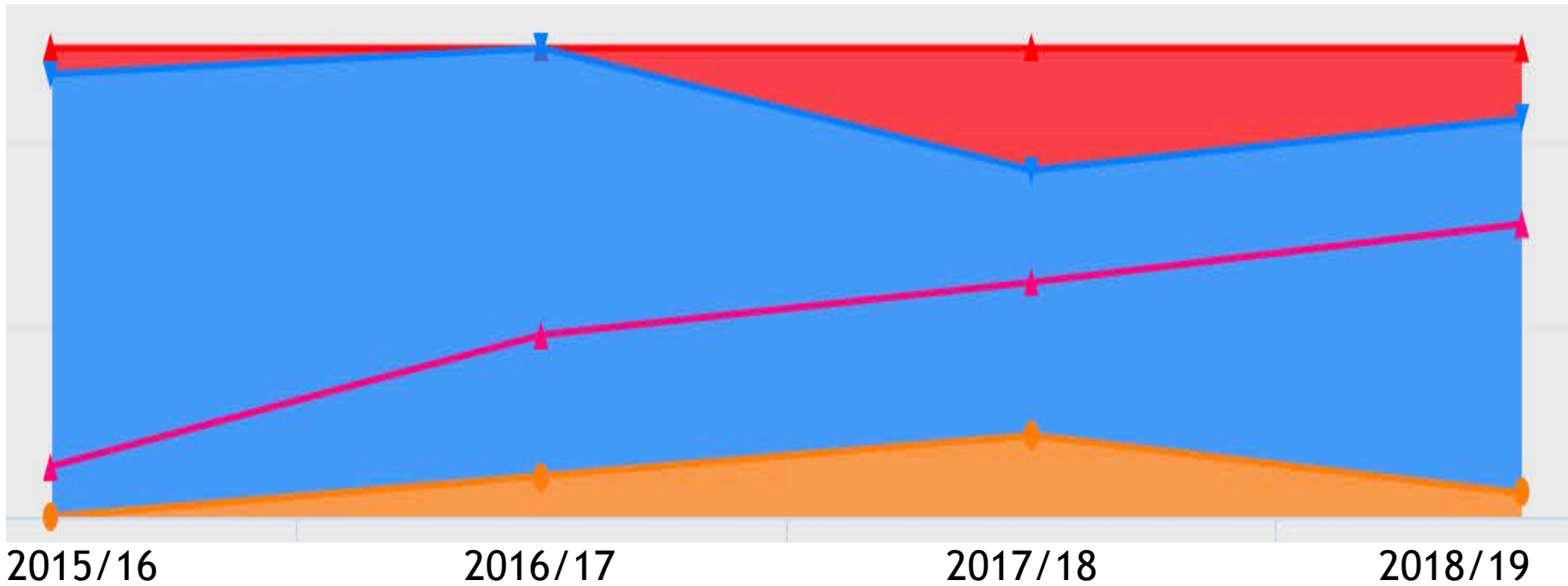
6. Deploy the appropriate tools(3/3)

Scoring table:

Challenge	score	n. of challenges	total	duration
Reading + quiz	15	10	150	120 min
Class	10	12	120	180 min
Homework	30	10	300	120 min
Self-evaluation	20	7	140	60 min
PE1, PE2	200	2	400	60 min



Results



Conclusions

A creative approach to learning programming

- for Digital Media students.

Creative Programming Matrix

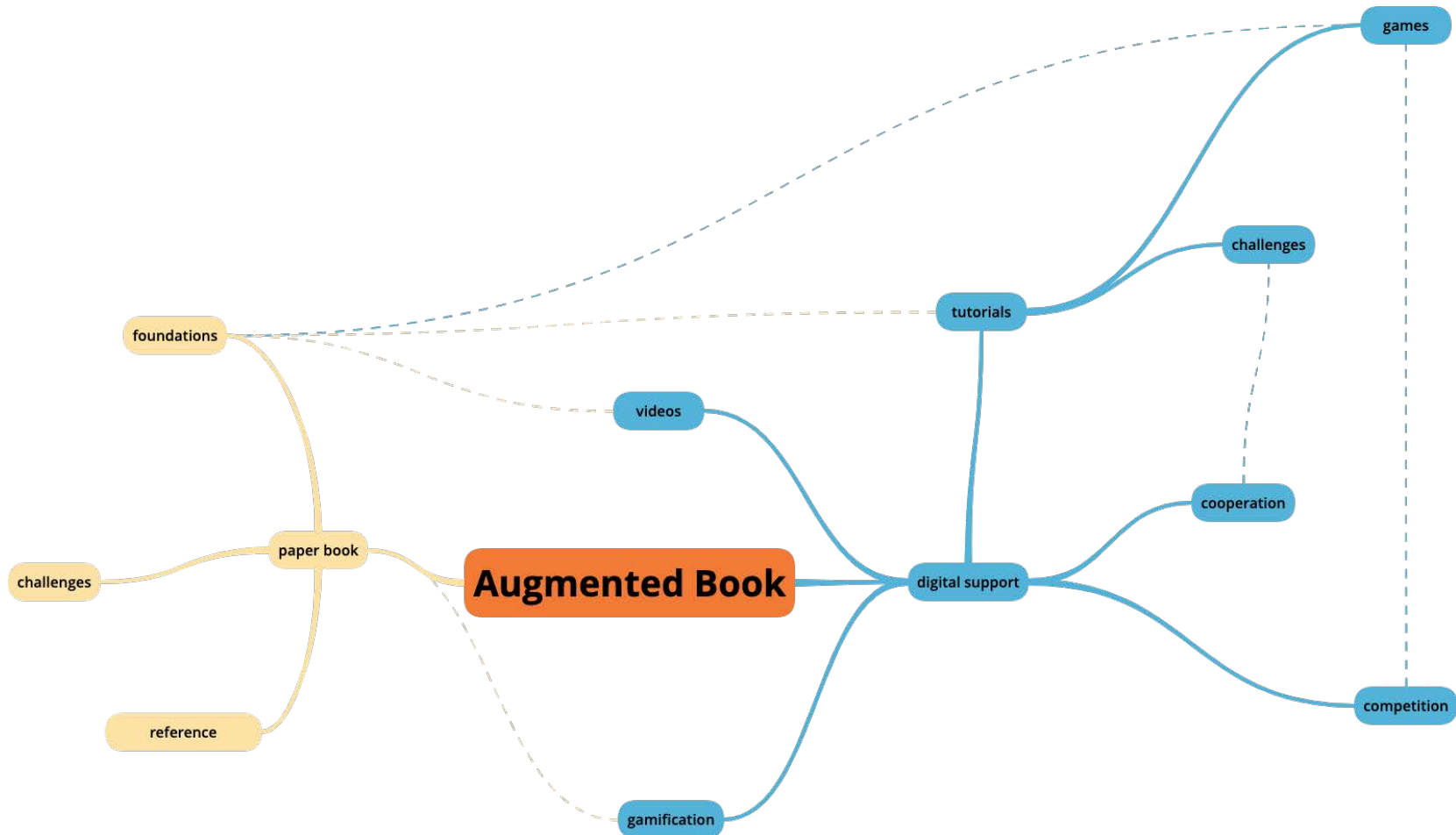
- a framework to create the tutorials to support the PBL approach.

Constructionist approach

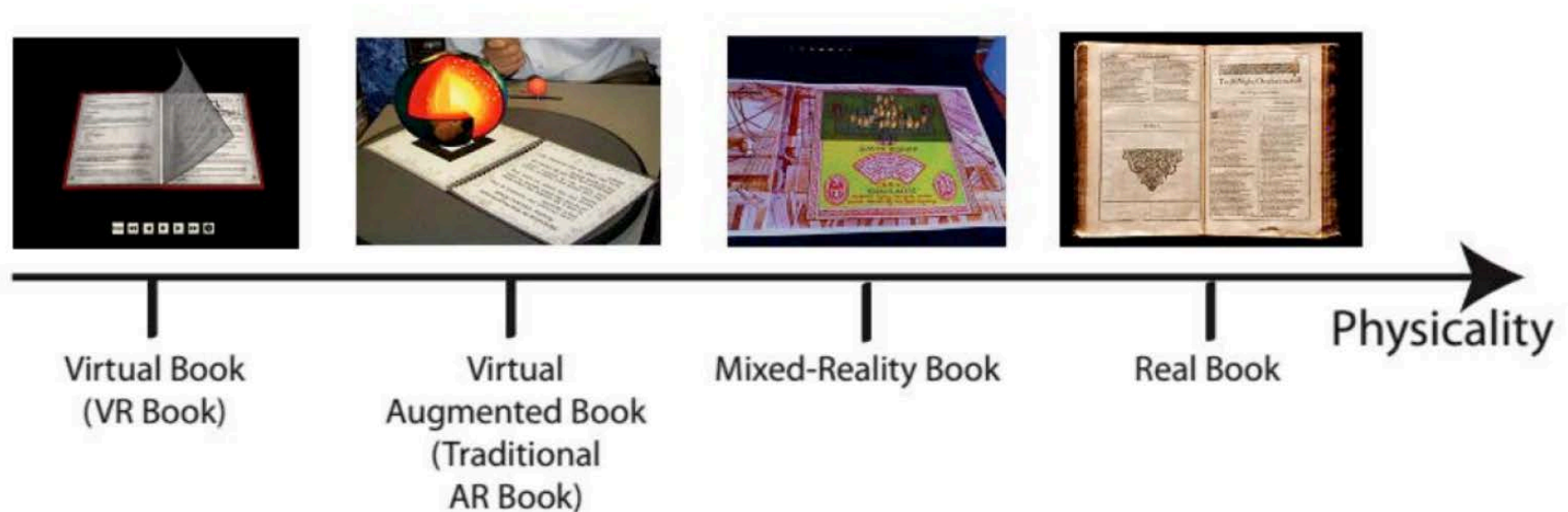
- Learning by making.



Future work - an augmented book



The Physicality Continuum (Grasset et al., 2008)



The Augmented Book

Gamified APP

tutorials

videos

games

challenges

sharing





Universidade do Porto

FEUP Faculdade de Engenharia



Creatively learning programming in Digital Media Education

G2T

António Coelho
acoelho@fe.up.pt