Spilprogrammering

Who am I?

Marco Scirea

marcoscirea.com

- Associate Professor at the SDU Metaverse Lab
- PhD in music generation for games
- Studied Game Design at the IT University of Copenhagen
- I like to make weird things, especially if these make other things

Favorite games

- Most LoZ games
- Night in the woods
- The Longest Journey series
- Hat in Time
- any Zachtronics game
- the Witcher 3
- a lot more…

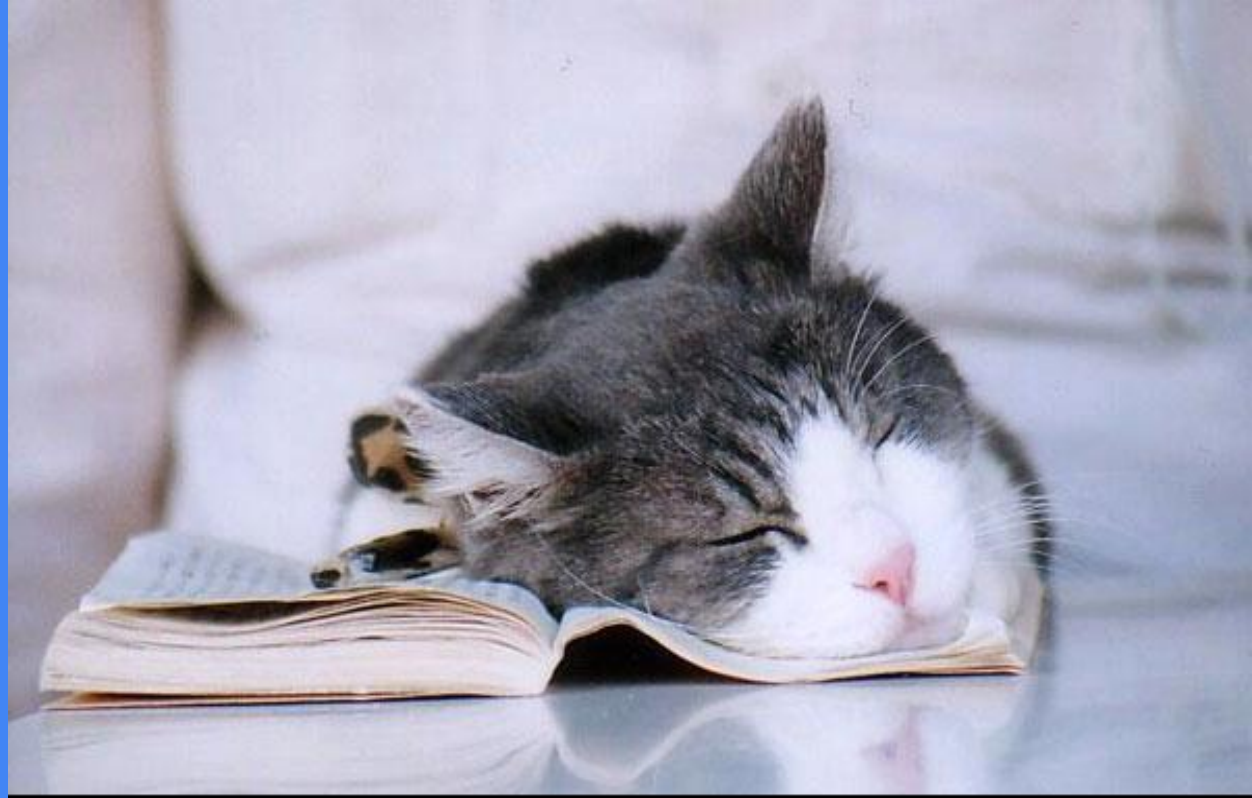
Favorite game I recently played:
Dredge/Signalis

What is this course about?

- Apply control and decision-making structures
- Apply simple data structures
- Apply a modern development environment for the construction of 3D applications
- Apply specific game programming techniques, such as
  - management of points and lives
  - Level design
  - Collision detection
- Construction of multi-user games

You will learn how to use Unity to make different kinds of games.

I will *briefly* introduce you to some procedural content generation



tl;dr

Materials

- This course is mostly based on **Unity's video tutorials**
- Support book: **Game Development with Unity** by Michelle Menard
- Support book 2: **Illustrated C# 7** by Daniel Solis (especially for HUM students)
- Extra book: **Procedural Content Generation in Games** http://pcgbook.com/

What will you do during this course?

- Implement 5 games with the help of tutorials
- Expand on these games
- Learn a little about procedural content generation
- Combine PCG with one of the games you developed

Two main blocks

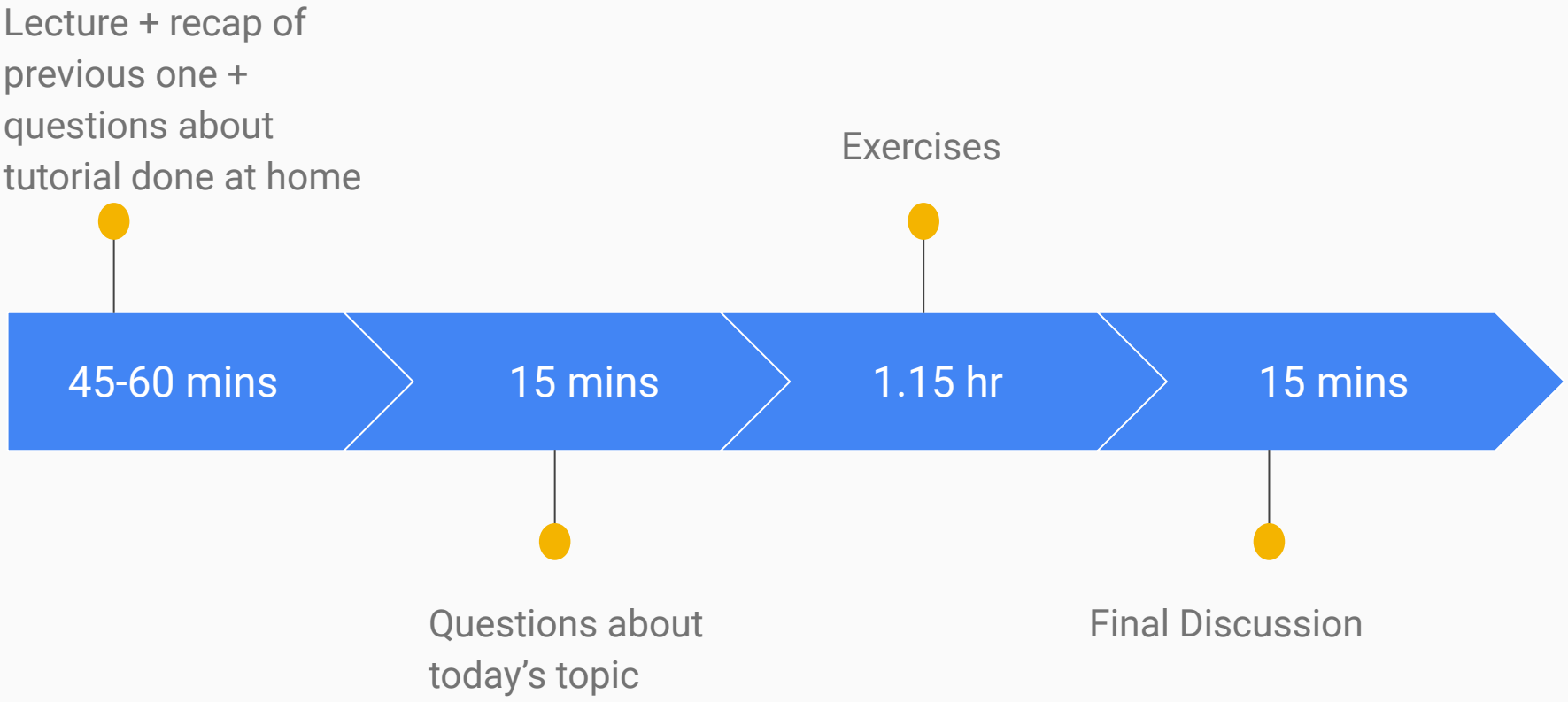
Who are you?

Let’s find out!

- Are you a humanistic or tech student?
- Why did you choose this bachelor programme?
- Is there one topic you’d really like to be covered in the course?
- What, if anything, do you hope to get out of this class?
- What would be the weirdest combination of human and animal?

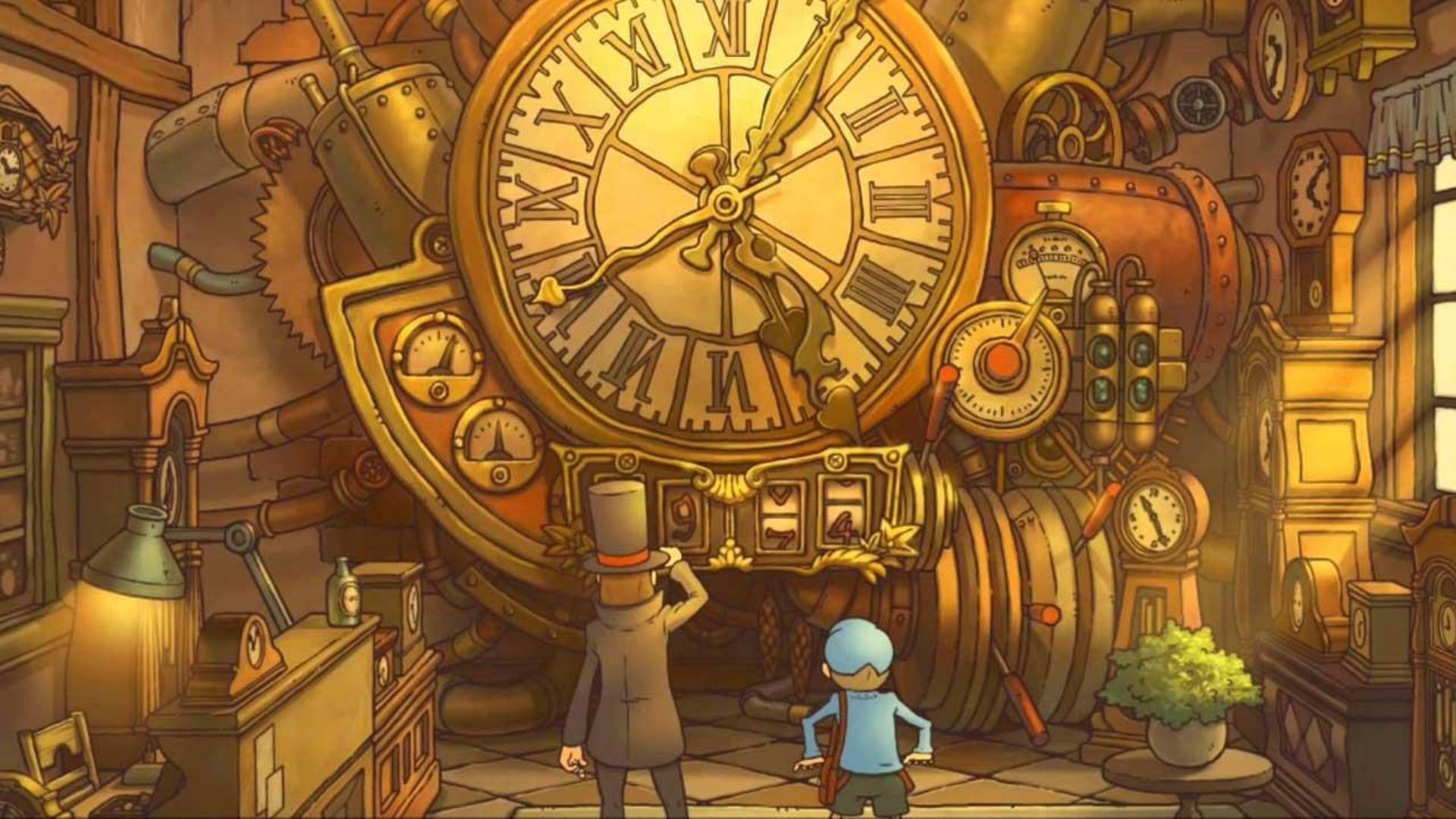
https://goo.gl/forms/WqMydvGouzc0Ja963

What about lecture structure?


Lecture + recap of previous one + questions about tutorial done at home
45-60 mins
15 mins
Questions about today's topic
Exercises
1.15 hr
15 mins
Final Discussion

How are we going to use the tutorials?

- You will implement the tutorials (or part of it) at home
- In class we will discuss them, and you will have time to fix them or finish them
- The main task during class will be to extend the tutorial (e.g. add a new feature)



Is the exam going to kill you?

3 parts

7-point scale grade

- Small project
- Short report
- Oral exam

Small Project + Report

What do I want to evaluate?

- That you can implement some mechanic in Unity
  - You can do some simple programming
  - You understand some of the principles and structures Unity gives you
- That you can shortly and clearly describe what you did (the report)
  - Writing forces you to reflect!

Small Project + Report

How much work is it?

- Not much, you will implement a new mechanic/feature in a game of your choice
- The report should be very short (around 3 pages)

Why not a predefined project?

- I want to give you freedom to come up with your own ideas, it’s more fun for you that way :)

The oral exam

What do I want to evaluate?

- That you can understand the basics of programming introduced in this course
- That you know what kind of Unity structures to use to solve a problem

The oral exam

How does it work?

- There will be 5 questions
- You will pick a random question
- Then you have some time to answer (I recommend you have a small presentation)
- Finally we will ask you some questions as discussion, which might go on topics not covered by your chosen question

What do I expect from you?

- Be curious: don’t be afraid of asking questions!
- Be responsible: I won’t force you to do homeworks or assignments
- Be creative: show me some cool things :)

... and Rasmus

Questions and feedback

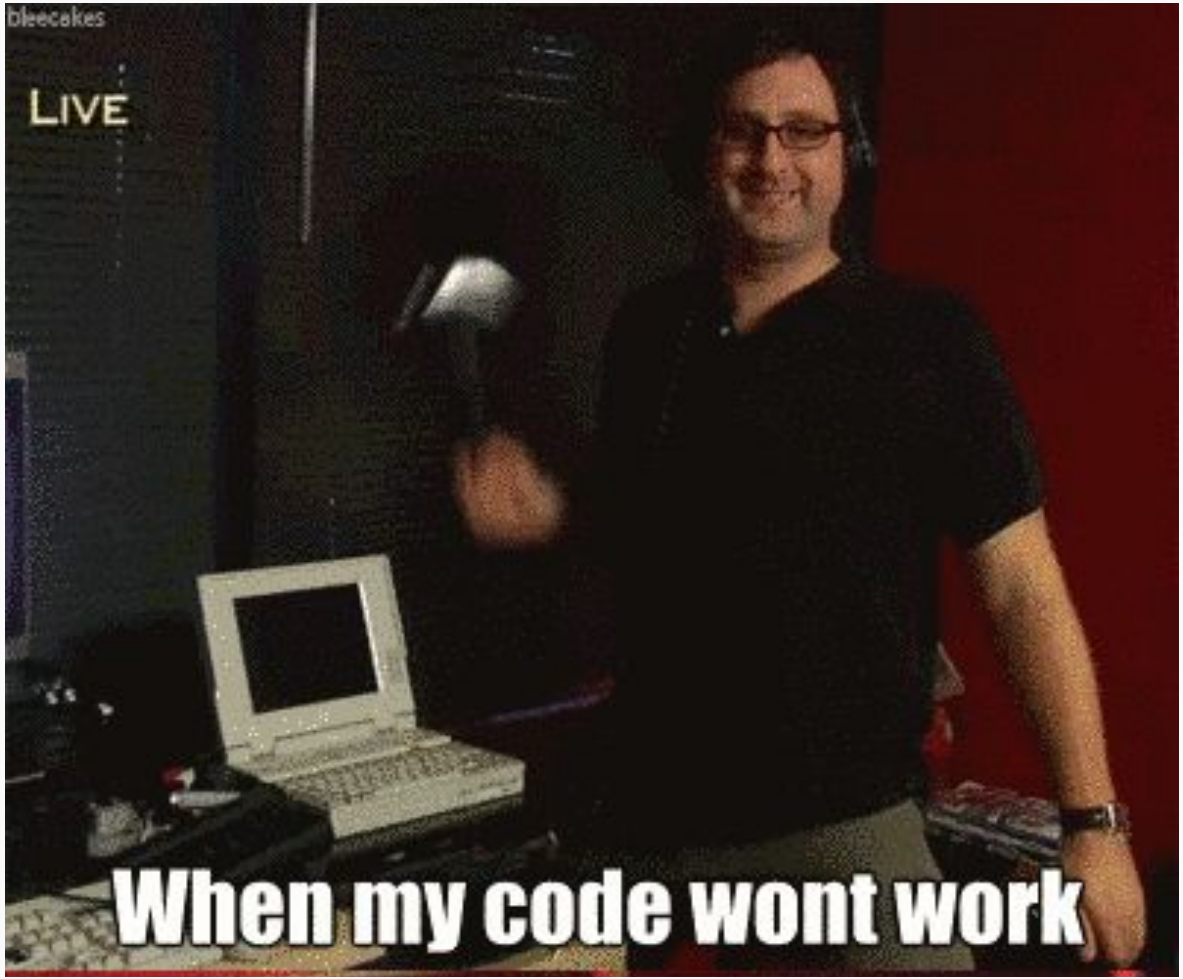
- Any general question about something I didn’t cover?
- We can also adjust the course later on, after we see what works better and what doesn’t

Disclaimer

Programming is hard!

During this semester you will at times think:

- “It’s too hard for me!”
- “I’m not smart enough!”
- “I’m just not cut out for coding!”
- “Why is ClassmateX so good at this? I must be untalented”



Why should you keep at it?

Programming is a valuable skill! (and skills are not easily obtainable)

It will be helpful throughout your education and beyond!

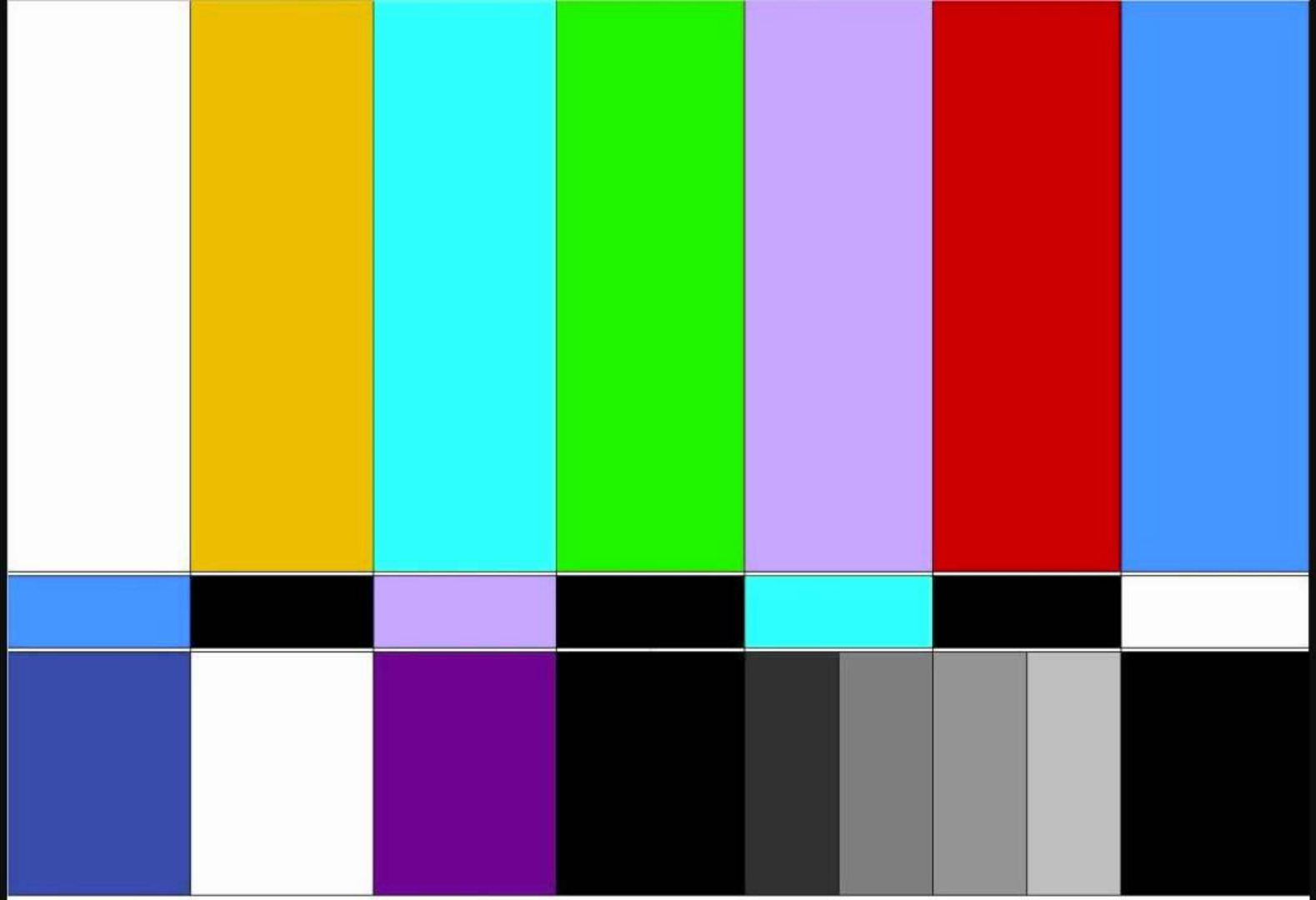
How to make the journey easier?

Incremental: start with easy things and expand them gradually. The course structure should help :)

Visualizing what you want to achieve can help! Don’t shun pen&paper

Ask people for help, often a fresh perspective can help you figure out a solution

Fail, and fail quickly!



Now to install Unity! (hopefully you already did)

https://store.unity.com/download?ref=personal

Let’s jump right into it :)

PS: you can take a look at these videos as well: https://learn.unity.com/project/beginner-gameplay-scripting

Warning!

- Today we will focus on looking at things practically
- There is a lot of information that will probably be new to you
- Next time we will review what we see today with text description in the slides as well :)

Intro to Unity

Exercises: let’s make a cookie clicker

1. Scripts as Behaviour Components - Input.GetKeyDown(KeyCode.R)
2. Variables and Functions - add counter to your cookie clicker
5. IF Statements - make something happen only on condition, like every 10 cookies
6. Loops - multiple cookies! (array) Add to them using a loop
9. Update and FixedUpdate – make cookies increase by 1 each second (new Behaviour)
11. Enabling and Disabling Components - enable auto-increase on condition
12. Activating GameObjects - Add some eye-candy!
13. Translate and Rotate – Make the cookie spin/move!

Numbers point to relevant videos at https://learn.unity.com/project/beginner-gameplay-scripting

Structure of scripts in Unity
+
variables, functions, and loops

Scripts in Unity

- Scripts are executed in Unity as Behaviours, which have to be attached to a GameObject present in the scene

- To attach a script drag-and-drop it on the object or add it in the Inspector
- The : MonoBehaviour part is important, we'll see why later

Unity Behaviours functions

Start(): this function is executed only once when the object to which the script is connected is activated.
Use for initializations.

Update(): this function is executed every frame of the game execution, this is where you will write most of your code.

Variables

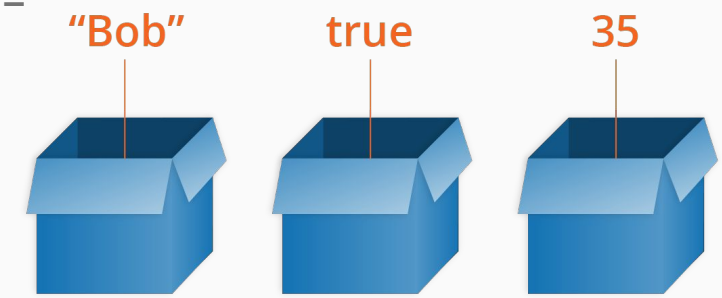
A variable is a container for values

You define (or initiate) a variable by:

```
TYPE NAME;        //e.g: int myVariable;
```

You can put values in a variable using the operator =

```
myVariable = 5;
```



Types

You need to use different types for different information you want to store, the most common ones are:

- Integers: int
- Real numbers: float or double (for bigger numbers)
- True or False statements: bool
- Sentences: string

Functions

Functions allow you to write small reusable parts of code, which take some input and return an output.

```
OUTPUT_TYPE FUNCTION_NAME(ARGUMENTS){

    CODE

    return OUTPUT;

}
```

Functions

Example: function to multiply by 2 a number

```
int Double(int number){
	int result;
	result = number * 2;
	return result;
}
```

Then you can call it like this:

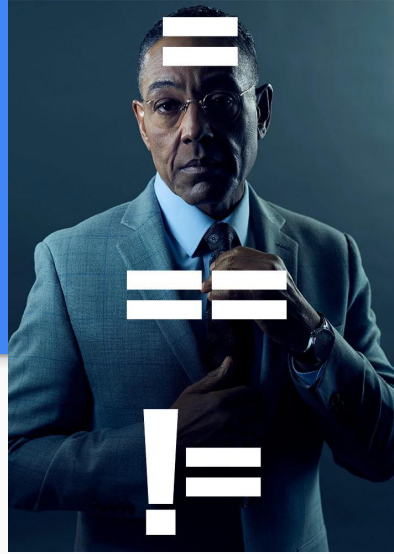
```
int x = Double(5);
```

If statement

Allows you to change the execution of the program depending on a condition. The else part of the statement is optional.

```
if (myVariable == true){
	//Code to execute when the condition is true
}
else{
	//Code to execute when the condition is false
}
```

Boolean operators



In IF statements you will need to use boolean operators, these are:

- == equal: 5 == 5 -> true, 4 == 5 -> false
- > greater than: 4 > 4 -> false, 5 > 4 -> true
- < lesser than: 4 < 4 -> false, 3 < 4 -> true
- >= greater or equal than: 4 >= 4 -> true, 5 >= 4 -> true, 3 >= 4 -> false
- <= lesser or equal than: 4 <= 4 -> true, 5 <= 4 -> false, 3 <= 4 -> true
- != not equal: 5 != 5 -> false, 4 != 5 -> true
- ! not: !false -> true, !(5 == 5) -> false

Feedback?

https://padlet.com/marco_prolog/challenges

Next lecture

No homework!

Take another look at the scripts we've seen today, if there are any doubts ask next time