



# Re-Evaluating Digital Technology to Enhance Learning

## Lesson Plan

---

**Lesson Title:** TEACHING APPLIED PHYSICS FOR TRADITIONAL VIDEOGAME DEVELOPMENT

**Lesson Duration:** 3 HOURS

**Level:** Vocational School – Multimedia and Mobile Programming

**Student Age Range:** 18-25

**Learning Style(s) of Students:**

This lesson is intended for students with any learning style, but it is specifically adapted for visual learners, individual learners and logical learners.

**Additional Student Information:**

Students attending this lesson must have advanced knowledge about computer programming and a basic level of maths and physics.

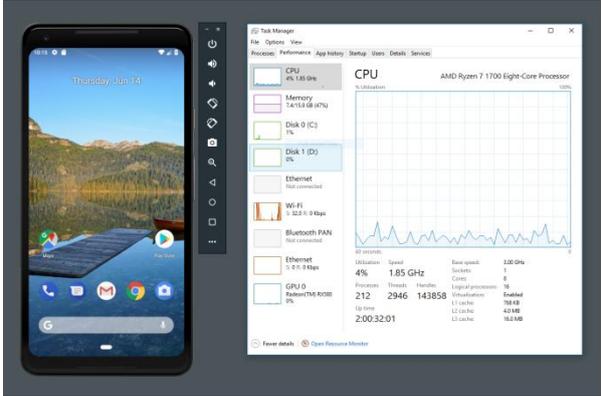
**Digital Technology Hardware Required:**

<p>A PERSONAL COMPUTER with virtualization capabilities and network connection: one for the teacher. The students can use another personal computer or a laptop to replicate the teacher's actions</p>	
<p>SIMPLE SCREEN PROJECTOR</p>	

**Software Required:**



<p>Android Studio for the development environment</p>	
---	--

<p>OBS and Google Meet for broadcasting and recording the class so the students can review the full lesson. Besides, students under confinement with the COVID-19 situation can follow the lesson as in normal circumstances</p>	
<p>Mobile phones emulator (Android virtual devices)</p>	
<p>Additional Software</p>	<p>Powerpoint, pdf reader and Moodle</p>

**Other Equipment Required:**



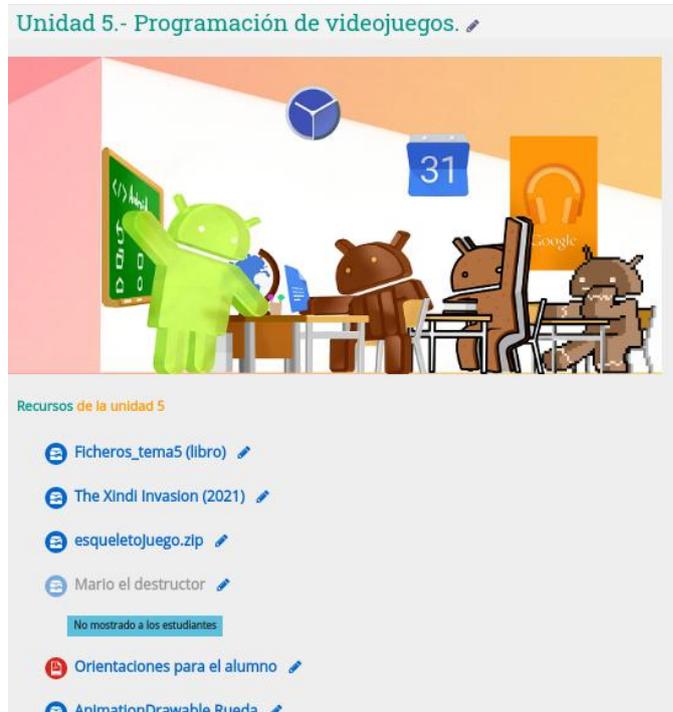
<p>Tablet or smartwatch with android operating system is optional as debug platform.</p>	
--	--

**Lesson Plan:**

The objective of the lesson is to learn how to apply physics through simulations for the development of traditional arcade videogames. This lesson is part of the unit 5 **Videogame Programming** of the module Multimedia and Mobile Programming. The students will follow teachers explanation to produce a small version of SuperMario Bros for the Nintendo Entertainment System with applied physics (movement + gravity) using vectors and mathematical measures. For this kind of lesson it is essential that students can visualise all the effects of applying different mathematical operations to the physics of the videogame. A Virtual Device is used to test the effect of these changes in the flow of the videogame. By modifying certain parameters and entering snippets of code in their Android Studio Project students can experiment how the main character of the videogame (Mario) moves throughout the screen and collide with different objects and enemies. Optionally these effects can also be tested on an smartphone or a tablet.

Time to complete the workshop (3hours)

[Preparation]: Before the workshop begins the teacher must have prepared all the documentation for the workshop in the moodle platform and set an **assignment** link for students to upload their work at the end of the lesson.



The documentation provided consists of:

- A text with all necessary explanation for the unit
- Snippets with source code in Java language

```
//mario corre
if(controles[DERECHA].pulsado) {
    if(contadorFrames%3==0){
        estado_mario++;
        if(estado_mario==4) estado_mario=1;
    }

    if(posicionMario[X]<0.7f*maxX) {
        posicionMario[X] = posicionMario[X] + deltaT * velocidadMario[X];
        velocidadMario[X] = velocidadMario[X] + deltaT * gravedad[X];
    }else{
        pos_inicial_mapa=pos_inicial_mapa + deltaT * velocidadMario[X];
        velocidadMario[X] = velocidadMario[X] + deltaT * gravedad[X];
    }
}
```



- A Powerpoint presentation with graphical explanations of the vectors and algorithms to be used.



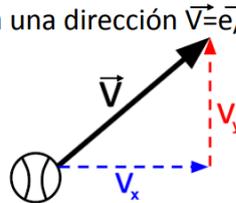
- 3. [20 mins] How to draw sprites and let them move on the screen



[30

minutes]: Once the videogame is ready to be launched, the teacher opens the powerpoint and explains the students the basics of the movement that are going to be applied to the main character of the videogame. He explains to all students (in class and at home) the actions to perform, so Mario can move horizontally on the screen of the virtual mobile device.

- En cada iteración  $\vec{M}' = \vec{M} + \Delta t \vec{V}$
- Vector velocidad  $\vec{V}$ : Proporciona una dirección  $\vec{V} = \vec{e} / t$ 
  - $V_x > 0, V_y = 0$  
  - $V_x < 0, V_y = 0$  
  - $V_x = 0, V_y > 0$  
  - $V_x = 0, V_y < 0$  



[15 minutes]: Students are given some time to think about how to perform a jump by using a gravity vector.

[50 minutes]: The teacher now demonstrates how Mario can jump by explaining and using the snippets of code. He shows the students how to add the gravity vector to the horizontal movement and simulate a jump. Students

## Movimiento parabólico

- $\vec{V}' = \vec{V} + \Delta t \vec{G}$       $\vec{M}' = \vec{M} + \Delta t \vec{V}$
- Vector gravedad `private float [] gravedad=new float [2];`
- $\vec{V}$  y  $\vec{G}$  Iniciales
 

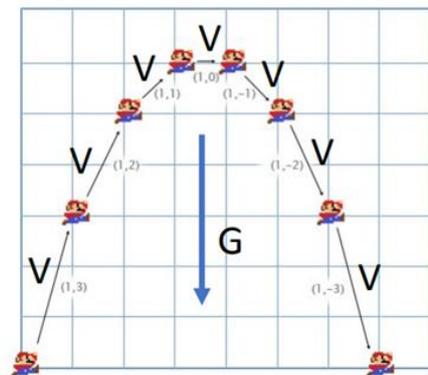
```

      velocidadMario[X]=maxX/tiempoCrucePantalla; //espacio / tiempo
      velocidadMario[Y]=-velocidadMario[X]*2; //componente Y dos veces componente X
      gravedad[X]=0;
      gravedad[Y]=-velocidadMario[Y]*2; //DOS VECES VELOCIDAD Y
      
```
- RENDERIZAR:  $\vec{V}$  depende de  $\vec{G}$ 

```

      velocidadMario[X]=velocidadMario[X]+deltaT*gravedad[X];
      velocidadMario[Y]=velocidadMario[Y]+deltaT*gravedad[Y];

      //rebote vertical
      if(posicionMario[Y]>=mInicialMario[Y])
        velocidadMario[Y]=- (maxX/tiempoCrucePantalla)*2; //velocidad inicial Y
      
```



are asked to use some of the snippets of code to replicate Mario's movement in their own simulators.

[10 minutes]: Students are asked to compress their android studio projects and upload it to the corresponding assignments in Moodle.

[After the lesson] The teacher downloads the assignments, evaluates them and gives them feedback through the moodle platform.