



Inclusive tools to create ESL video games

Design a traditional-looking video game for ESL

Introduction

Engagement is key in education, however, we tend to forget that we are not equal in face of engagement. Students with SLDs can encounter more difficulties in staying engaged in educational content due to the drain or frustration caused by their disabilities. Which is why educational games should first aim at being fun and entertaining.

What do we mean by a traditional-looking video game?

By traditional-looking game, we mean a non-serious game. The key to success is that the games must feel like games. A bad example would be to present students with the fiction of being

thieves that must unlock a safe but they have to do it by answering math questions. Despite pretending to be thieves, a worksheet of math problems is still a worksheet.

What is a game engine?

Overview

A game engine is a software that acts as an orchestra conductor. The instruments are visual and audio assets, and code that handles game physics, etc. They offer an interface that simplifies the use and combination of these assets. They also streamline the things that are required for games like handling user inputs.

Game engines that you can use

Construct3

It is usable online, so, from any device. The interface is a bit dry but straightforward. Some very nice games were made with it. It's powerful and simple but a bit pricey.

Gdevelop

It is very similar to Construct, but the interface is a bit more pleasing and friendly. Also, it is open-source software. It's a bit less powerful than Construct but it is totally free.

Core

Game creation interface based on Unreal Engine 4 and focused on multiplayer and online characteristics. However, there is no no-code tools. The scripting system is simple but you'll need code.

Stencyl

Stencyl build-up on the Scratch philosophy. It uses a similar coding system: puzzle-like elements. So students that learned scratch can have some familiarity with it but with a much

more powerful and pleasing interface. The interface is very intuitive. However, there may be lacking features like an interface managing system, .

Flowlab

Very nice for low-level game-making. It's an all-in-one tool. You can create sprites. The interface is intuitive and the coding system as well. Something nice in Flowlab is that you can export game elements or code you made. So if you need something another user made you can simply copy it!

It is powerful and intuitive, fun to work with and not too pricey!

RPG maker

RPG maker is specialized in role-playing games. It streamlines the creation of all the elements these games feature: navigating a map, encountering and fighting monsters in a turn-by-turn manner... It requires no coding but limits you in what you can do (RPGs). Resources are not free but some people share some for free and it is possible to buy a lot for not too much money.

BuildBox

BuildBox is a no-code game engine that uses a Mindmap to organize game elements. It is ideal for non-programmers who want to do some 3D. Recently released with a free version. It has a very nice interface and is accessible. The low code features (drag and drop nodes) are quite nice however you'll end up finding limitations. But you can use custom script nodes to overcome these. However, the coding experience is not great using those.

Minetest

Minetest is a very specialized engine. It is aimed at copying Minecraft but in an open-source and customization-oriented way.

It is possible to modify the base game using the Lua scripting language (easy).

A page dedicated to how to use Minetest for education is available [here](#).

Engine	Access-ibility	Hard-ware	Power	Docum-entation	Time	Money	2D	3D	OS	Low code	Platform	Asset store
FlowLab	++	-	+	+	-	29/month for 35 students	v	x	web	yes	All but consoles	
Construct3	+	--	=	+	-	9€ /month/ seat	v	x	web	yes	All but consoles	yes
Gdevelop	+	--	-	+	-	0	v	x	Win, Linux, Mac, web	yes	All but consoles	yes
Game-Maker	=	-	++	++	=	10€/month	v	x	Win, mac	yes	All	yes
Core	-	+	++	++	=	0	v	v	Win, mac	no	Core ptaform	Yes, huge
Godot	-	-	+++	+++	=	0	v	v	Pc, linux	yes	All	Limited
Stencyl	++	-	=	+	+	0	v	x	Win, Linux, Mac,	yes	All but consoles	no
Minetest	+	--	=	+	+	=	0	x	v	yes	Win, Linux, Android	No-need
Buildbox	=	=	+	=	+	+	0	v	v	yes	Win, mac	no

Conclusion

We presented how a game engines work and several easy-to-use, low or no-code engines that you can use to create engaging games.

Resources and references

- Serrano, H. (2016, January 9). How does a Game Engine work? An Overview. haroldserrano.com Retrieved from <https://www.haroldserrano.com/blog/how-do-i-build-a-game-engine>
- [Ask Gamedev]. (2018, June 19). 'Game Engines Explained' [Video File]. Retrieved from <https://www.youtube.com/watch?v=LMRZBKhQcRc>
- Chen, S. (n.d.). Facing Edutainment's Dark Legacy. gamesandlearning.org Retrieved from <http://www.gamesandlearning.org/2016/01/25/facing-edutainments-dark-legacy/>



Co-funded by the European Union

This project has been funded with support from the European Commission. This publication 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.

Project code: 2021-1-BE01-KA220-SCH-000027783

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<http://creativecommons.org/licenses/by-nc-sa/4.0/>).

Learn more about D-ESL at: <https://www.d-esl.eu>