**The canvas element**

You've most likely played a video game or

watched an animation in your web browser before.

Perhaps you've wondered how it works Well in this video you'll be

introduced to the basics of graphics programming in the web browser.

There are many authoring tools available to produce animated and

video game content for the web browser.

But all of these are underpinned by one of four technologies GIF.

WebP 2D canvas and WebGL.

The graphics interchange format or

GIF along with the Web P image format are used for animated images.

An image using this format has no interactivity, but it's popular for

embedding animated content and for building two D video games and animations.

The two D canvas allows two D graphics to be drawn in the web browser.

It's popular for building two D video games and animations.

The logic and behavior of an animation or video game is defined using javascript.

But what about three D graphics.

Web G L is a popular technology for building two D and three D animations and

advanced graphics.

The benefit of WebGL is that a dedicated chip in your device called a graphics

processing unit?

Or GPU does the calculations when graphics are rendered to the screen.

This allows for complex calculations such as lighting simulation and

rendering massive three D worlds.

It is important to note that while WebGL will always use the GPU,

it's not guaranteed that the two D canvas will use the GPU.

This will depend on the web browser implementation.

WebGL is a complex technology and requires a lot of background information so

for now let's focus on the canvas element.

Let's examine how to animate some simple graphics.

Using the 2D canvas.

I'm going to animate a bouncing ball.

I've opened visual studio code and have set up two files index dot html and

animation dot Js inside index dot html.

I have added the canvas html element.

I have set its with attribute to 640 its height attribute to 480.

I have also set its ID attribute to my underscore canvas so that I can access

the element in my javascript code after the canvas element I have a script element

with its SRC attribute set to animation dot Js I now open animation dot Js at

the beginning of the file I look up the canvas element using the document dot

get element by ID function and pass in my underscore canvas at the parameter.

The result is stored in a variable named canvas.

Next I look up the two D canvas context using canvas dot get context and

pass in the parameter to D.

The result is stored in a variable named context below this I define some variables

to track the speed, position, movement and radius of the bouncing ball.

Next I define a function called move ball.

It is responsible for moving the ball across the screen and

changing its direction once it reaches the left or right side of the screen.

After that I define a function called draw ball.

The function clears the canvas area using context dot clear wreck it then

draws the ball in its current position using the context dot begin path,

context dot arc and context dot fill functions.

Finally, I define an animate function which calls the move ball and

draw ball functions after doing this, it then calls window dot request animation

frame to schedule the animate function to be called again at the end of the file I

call window dot request animation frame to start the animate function being called.

Using a small amount of javascript code.

I've animated a bouncing ball.

This is a straightforward example of the capabilities of the two D canvas.

There are many more possibilities to develop animations and video games and

there are also many libraries available to simplify development.

These are linked in the additional resources Reading

There are three main technologies underpinning the production of animated and video game content for the web browser. These are GIF or webP, 2D Canvas and WebGL. Which of the following statements about these technologies are true? Choose all that apply.



2D Canvas is used for building both 2D video games and animations.

Correct

That's correct! The 2D Canvas allows 2D graphics to be drawn in the web browser.



WebGL doesn't use the GPU to render graphics to screen.

This should not be selected

Not quite. WebGL is able to produce advanced graphics because of the GPU that can do calculations complex enough to allow for lighting simulation and rendering 3D worlds.



GIF and webP are used to animate images, but with no interactivity.

Correct

That's correct! GIF and webP are popular for embedding animated content but the format has no interactivity.