**REFERENCES**

**1. Window: requestAnimationFrame() method**

The window.requestAnimationFrame() method tells the browser you wish to perform an animation.

It requests the browser to call a user-supplied callback function before the next repaint.

The frequency of calls to the callback function will generally match the display refresh rate.

The most common refresh rate is 60hz, (60 cycles/frames per second), though 75hz, 120hz, and 144hz

are also widely used. requestAnimationFrame() calls are paused in most browsers when running in background

tabs or hidden <iframe>s, in order to improve performance and battery life.

**Link:** https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame

**2. HTMLCanvasElement: getContext() method**

The HTMLCanvasElement.getContext() method returns a drawing context on the canvas, or null

if the context identifier is not supported, or the canvas has already been set to a different context mode.

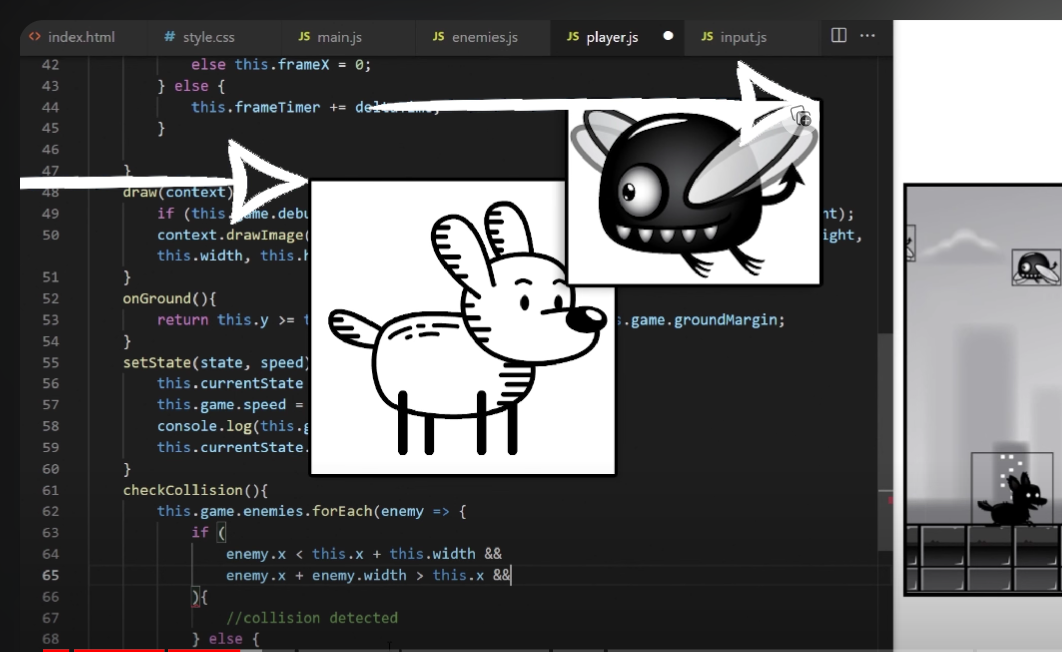
Later calls to this method on the same canvas element, with the same contextType argument, will always return

the same drawing context instance as was returned the first time the method was invoked. It is not possible to

get a different drawing context object on a given canvas element.

**3.** JavaScript automatically creates references to all elements with IDS into the global namespace, using it's ID as a variable name

**4. Hitbox and collision:**

****

**Code:**

      if (

                enemy.x < this.x + this.width &&

                enemy.x + enemy.width > this.x &&

                enemy.y < this.y + this.height &&

                enemy.y + enemy.height > this.y

            )

* Check if 2 pictures are overlapping

**5. Use unshift() instead of push() in particles:**

Use unshift() for add one or more elements and return new lengths, instead of push only one in time as push() -> delete the gap() of using slice(0, all) in main -> smoothen the particles’ effects.