

Alejandro Aceytuno González

Practica 7

a) Se han reconocido y analizado las tecnologías relacionadas con la inclusión de contenido multimedia e interactivo.

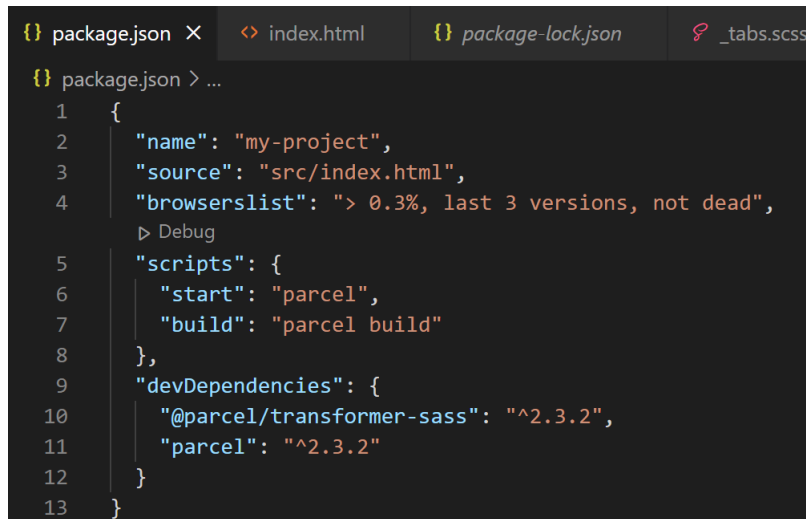
En este entorno de trabajo he usado Parcel y Sass.

Parcel es un Bundler de aplicaciones web, que usamos para programar en un entorno de trabajo familiar, pero con la capacidad de optimizar nuestros archivos de forma muy sencilla, también instala otras paquetes, como Sass.

Sass es una extensión de CSS. Tiene su propio lenguaje, con muchas ventajas, entre ellas anidamiento y modulación. Con un solo comando es posible transformar todo nuestro código en Sass (.scss) a css común, pero no es necesario gracias a Parcel.

b) Se han identificado las necesidades específicas de configuración de los navegadores Web para soportar contenido multimedia e interactivo.

Para estar conforme a lo pedido (los navegadores que tengan actualmente soporte, las últimas 3 versiones y su uso sea más del 0,3 % de la población.) se añade una sencilla línea de código en package.json



```
{} package.json X <> index.html {} package-lock.json ? _tabs.scss
{} package.json > ...
1  {
2    "name": "my-project",
3    "source": "src/index.html",
4    "browserslist": "> 0.3%, last 3 versions, not dead",
5    "scripts": {
6      "start": "parcel",
7      "build": "parcel build"
8    },
9    "devDependencies": {
10     "@parcel/transformer-sass": "^2.3.2",
11     "parcel": "^2.3.2"
12   }
13 }
```

“browserslist”: nos permite especificar a que rango de buscadores queremos hacer build.

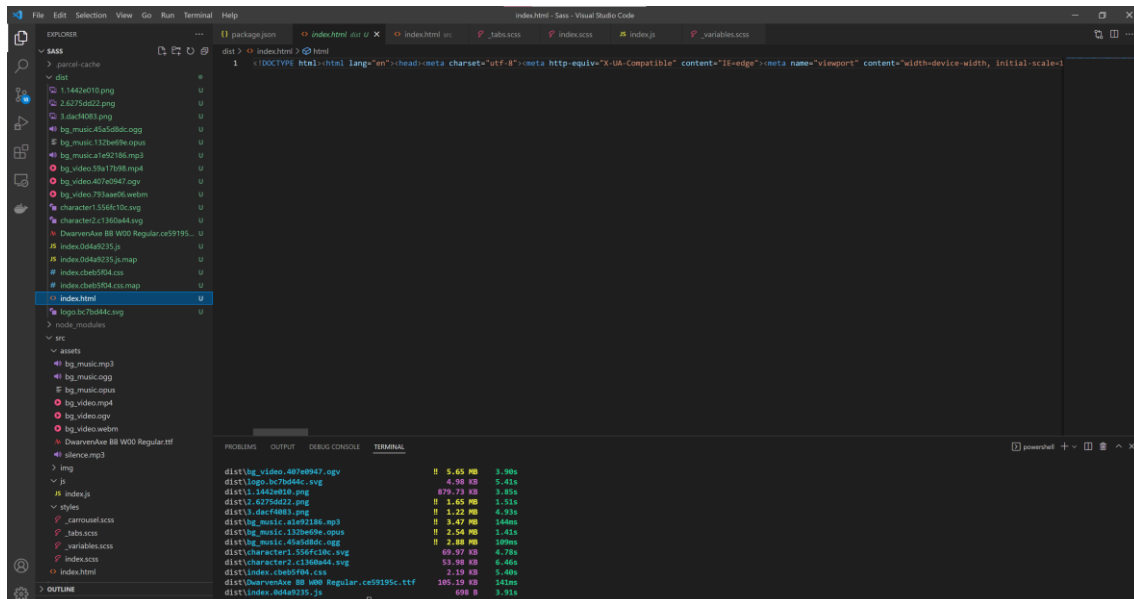
Todo esto nos lo proporciona Parcel

c) Se han utilizado herramientas gráficas para el desarrollo de contenido multimedia interactivo.

El Logo, fondos y personajes han sido realizados con Affinity Designer y Affinity Photo.

d) Se ha analizado el código generado por las herramientas de desarrollo de contenido interactivo.

Cuando ejecutamos `npx parcel build src/index.html` se nos crea una carpeta `dist`, con todos los archivos comprimidos, por ejemplo este `index.html` tiene todo el código en una sola línea.



e) Se han agregado elementos multimedia a documentos Web.

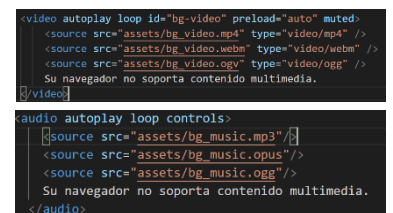
He incluido un video y un audio en la pagina web. El audio es una canción del juego original que se inspiro este trabajo ([link](#)), no he hecho que se reproduzca automáticamente porque no es una buena práctica, ya que muchos navegadores lo están prohibiendo, Google incluido.

El Video es usado de fondo de pantalla silenciado para que pueda ser autorreproducido por lo mismo que he mencionado antes el video original es este: ([link](#))(solo he usado los primeros segundos)

He usado la etiqueta audio y video

Ambos tienen tres tipos de extensiones de seguro , por si uno falla.

Mp3, opus y ogg para audio. Mp4, webm y ogv para video



f) Se ha añadido interactividad a elementos de un documento Web.

He añadido dos elementos interactivos en la página web, uno de ellos es un carrusel y el otro es un menú de pestañas, el carrusel muestra escenarios del videojuego y las pestañas muestran los dos personajes.

He decidido separar en módulos los estilos del carrusel y de las pestañas, para una mayor facilidad de cambio.

```

//carrousel code
var slideIndex = 1;
showSlides(slideIndex);
function plusSlides(n) {
    showSlides(slideIndex += n);
}
function showSlides(n) {
    var i;
    var slides = document.getElementsByClassName("fotos");
    if (n > slides.length) {slideIndex = 1}
    if (n < 1) {slideIndex = slides.length}
    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }
    slides[slideIndex-1].style.display = "block";
}
//tabs code
function openTab(evt, tabName) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace(" active", "");
    }
    document.getElementById(tabName).style.display = "flex";
    evt.currentTarget.className += " active";
}

```

El código de ambos elementos de javascript.

Ambos elementos cambian el display a none para ocultar o mostrar el contenido

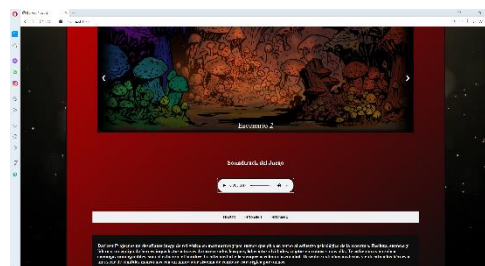
Estas funciones son llamadas por el evento onclick de los botones de los elementos

g) Se ha verificado el funcionamiento de los elementos multimedia e interactivos en distintos navegadores

He comprobado que funciona en Google:



En Opera:



Y en Firefox :

