Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Scuola d’arti mestieri Trevano |
| Data | 26.03.2021 |

|  |
| --- |
| Lavori svolti |
| Oggi ho trovto una libreria javascript che mi permette di visualizzare i pdf sul browser. Quando si apre un documento pdf sul browser viene aperto con la modalità visualizzatore pdf dei browser (ogni browser ha il proprio visualizzatore). La libreria PDF.js è una piattaforma generica basata su standard web per l’analisi e il rendering pdf. Nel mio progetto ho creare un nuovo script che si chiama pdfViewer che permette di visualizzare dei documenti pdf. All’interno ho implementato il seguente codice.    La variabile pdfDoc corrisponde al documento pdf che deve essere visualizzato sullo schermo. La variabile pageNum corrisponde la numero della pagina del documento pdf. La variabile pageRendering permette sapere se deve essere fatto il rendering della pagina del documento pdf. La variabile pageNumPending permette di aspetttare il rendering della pagina del documento pdf. La variabile scale corrisponde alla scala di rendering di ogni pagina del documento pdf. La variabile canvas corrisponde all’area nella quale viene stampato i pdf. La variabile ctx corrisponde al contesto grafico. Per finere la variabile isFirst permette di sapere se il documento è stato già stampato.  In questa libreria il rendering permette di andare analizzare il file pdf (analizza il contenuto della pagine del documento) e di stampare sullo schermo il contenuto delle varie pagina come se fosse un’immagine.  Immagine che contiene testo  Descrizione generata automaticamente  La funzione loadPdf permette di visualizzare il file pdf. In effetti quest’ultimo riceve il parametro url che corrisponde all’url del documento pdf da rendere visibile e riceve anche il parametro numbber che corrisponde all’identiifcativo del filmato/presentazione. Il primo codice che c’è scritto al suo interno permette di caricare tramite il tag “<script>” la libreria PDF.js. La libreria PDF.js utilizza le promesse e quanto sopra restituirà un'istanza PDFDocumentLoadingTask che ha una proprietà promise che viene risolta con l'oggetto documento. Utilizzo anche la variabile isFirst per stampare solo una volta il documento pdf.  Immagine che contiene testo  Descrizione generata automaticamente  La funzione renderPage permette di ottienere le informazioni sulla pagina dal documento, ridimensionando l’aera (canvas) di conseguenza e visualizza la pagina. Quest’ultimo deve ricevera anche il numero della pagina attuale del documento pdf. Ogni pagina PDF ha il proprio viewport che definisce la dimensione in pixel (72DPI) e la rotazione iniziale. Per impostazione predefinita, il viewport viene ridimensionato alla dimensione originale del PDF, ma questo può essere cambiato modificando il viewport. Quando viene creata la finestra, verrà creata anche una matrice di trasformazione iniziale che tiene conto della scala e della rotazione desiderate e trasforma il sistema di coordinate (il punto 0,0 nei documenti PDF in basso a sinistra mentre quella del canvas 0,0 è in alto -sinistra).  Immagine che contiene testo  Descrizione generata automaticamente  Ho creato anche questo metodo che permette di mostrare la pagina succesiva del documento pdf. Ho creato anche il metodo queueRenderPage che permette di attendere il rendering che è in corso e inseguito esegue immediatamente il successivo rendering. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Oggi ho ricontrato un problema nel stampare i documenti PDF. In effetti quando stampo un solo documento sullo schermo, funziona in modo corretto. Invece quando sono presneti due documenti PDF, viene stampato solo il file. In teoria quando appare il secondo documento il primo documento dovrebbe essere visibile fino alla data di fine. Nel mio caso scompare il documento, ma resta visibile ancora la canvas.    Questo problema è dovuto che nello script pdfViewer le variabile che permettono di fare funzionare il rendering del documeto (url del pdf, numero attuale della pagina, documento pdf, ..) vengono sovrascritte quando viene inserito un nuovo documento pdf. Ho provato a creare una classe ma non funzionava e quindi l’unico sistema e di creare degli array che contengono le informazioni di ogni documento pdf per potere relizzare in modo corretto il rendering. Non ho ancora risolto il problema, ma molto probabilmente lascerò perdere questo requisito, perché non resta molto tempo. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Rispetto alla pianificaziono sono ad un buon punto, perché devo solomanete sistemare diversi aspetti di sicurezza e di grafica del sito web. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Per la prossima giornata di lavoro devo riuscire a sistemare il problema legato alla libreria PDF.js. Se non riesco a sistemare questo problema lascio perdere questo requisito e mi concentro sulla sicurezza dela sito web che devo ancora sistemare. Una volta terminato la sicurezza posso continuare tranquillamente la documentazione del progetto. |