

Diario progetto G-Gen

Data : 19 febbraio 2018

Autore : Cristiano Colangelo

Luogo: SAM Trevano

Lavori svolti

- Continuato a lavorare sul sito. Adesso comprendo molto di più il funzionamento di Go, anche perchè ci sono stati alcuni esempi vecchi e troppo complicati per fare cose che ora richiedono molta meno fatica.
 - `http.HandlerFunc("path", handler)` dove il primo parametro è il percorso (es. /generator o /) mentre il secondo è un metodo del tipo `func IndexHandler(w http.ResponseWriter, r *http.Request)`. Se l'utente accederà al percorso specificato, verrà invocato l'handler associato
 - `templates = template.Must(template.ParseGlob("templates/*"))` mette in cache tutti i file template contenuti nella cartella `templates` (specificato come parametro di `ParseGlob`), sarà possibile accedervi come spiegato sotto
 - `templates.ExecuteTemplate(w, "nometemplate", nil)` esegue il template specificato (messo in cache grazie al codice precedente). Il primo argomento `w` è dove deve stampare l'HTML finale (in questo caso `w` è la risposta del server HTTP specificato come parametro dell'handler), il secondo è il nome del template specificato con `{{define "nometemplate"}} ... html ... {{end}}` mentre il terzo argomento è la variabile contenente i dati da passare al template (in questo caso null, ma devo vedere come passare questi dati ancora)
- <https://github.com/kenwheeler/slick/> Questo mi servirà per la pagina contenente i form, farò vari step a cominciare dalla selezione dell'immagine con modifica della luminosità/contrasto continuando verso le impostazioni del modello 3D e della stampante, la visualizzazione del risultato e il download.
- <https://stackoverflow.com/questions/12102464/how-do-i-make-iframes-load-only-on-a-button-click> Questo mi servirà per caricare il visualizzatore solo quando l'utente avrà completato tutte le informazioni richieste. Alla fine del form, farò un controllo sui dati inseriti e se tutto è presente, caricherò un iframe contenente il visualizzatore. L'utente potrà anche visualizzare il risultato in una pagina a parte in modo da vederlo più grande.
- <http://evanw.github.io/glfw.js/> Questo mi servirà per modificare luminosità e contrasto delle foto. E' una libreria eccezionalmente semplice e molto veloce, il problema è che non è retrocompatibile con i browser più vecchi, ma non penso sia un problema ormai.

Problemi riscontrati e soluzioni

-

Punto di situazione del lavoro

Stesura sito web

Programma per la prossima volta

- Continuare a lavorare sul sito