

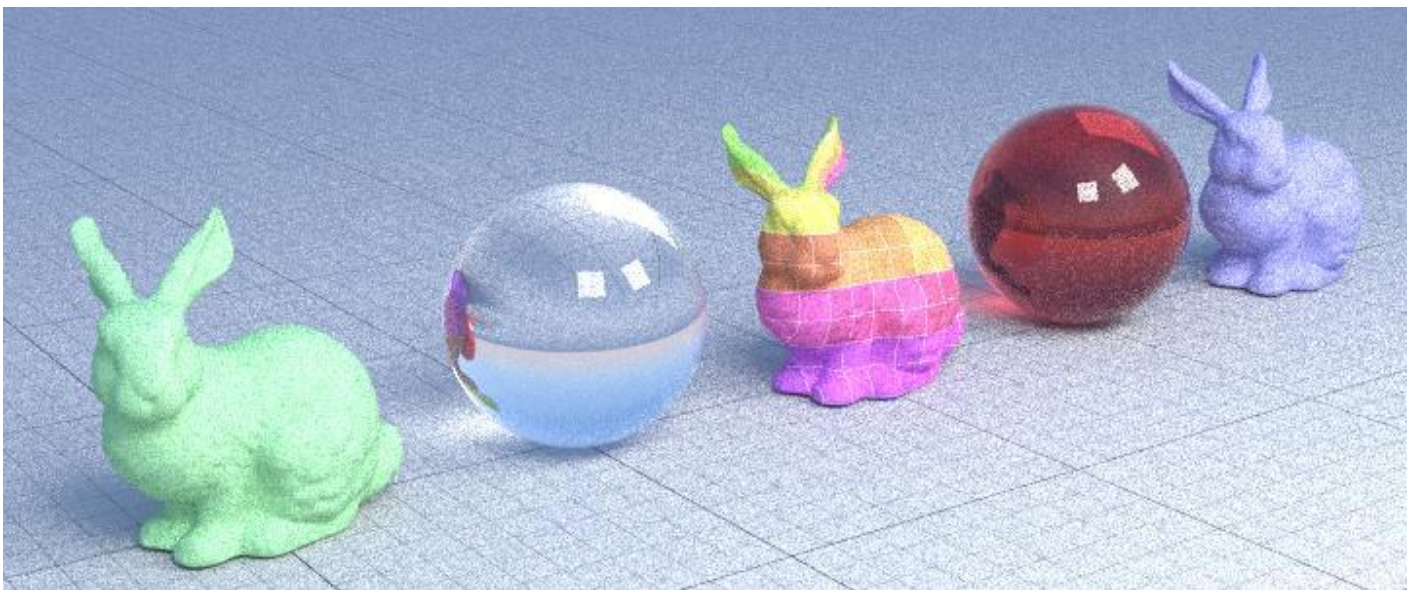
Extra Credit Facili

Refractive.

Per l'implementazione della rifrazione polished all'interno della nostra funzione *shade_raytrace* ho svolto i seguenti step di implementazione:

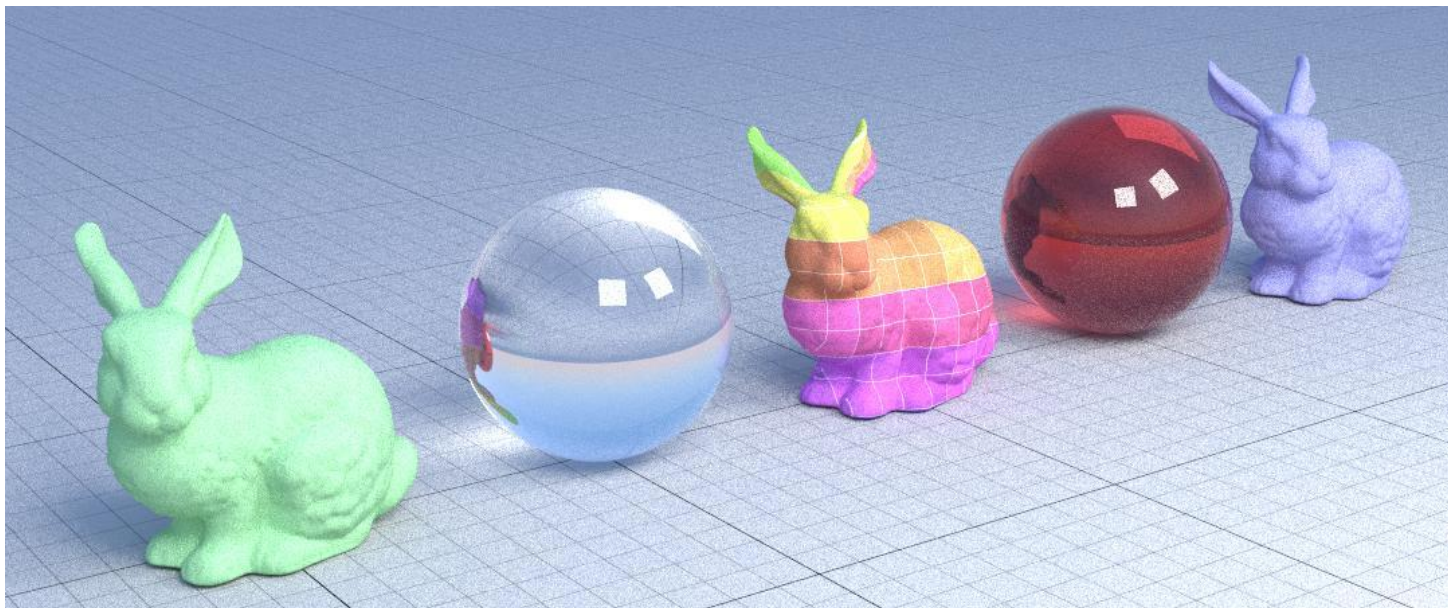
- Ho creato una nuova test folder *13_refractive*, che utilizza la stessa struttura del test *08_glass*. La differenza sta nel file json, dove ho modificato il materiale di *sphere 1* e *sphere 2* da transparent a refractive e settato un valore di $ior = 0.45f$
- All'interno dello switch{material.type} che itera sui casi del tipo di materiale, ho aggiunto un case *refractive*, per specificare il comportamento del renderer per quel tipo di materiale
- Il materiale *refractive* viene gestito in maniera simile ad un polished dielectric. Viene quindi eseguito un test con un valore fresnel_schlick che decide se eseguire la riflessione o la rifrazione (piuttosto che una diffusa come nei polished dielectrics)
- Seguendo il codice fornito dal sito specificato nel readme (credits alla fine di questa sezione), ho specificato due ulteriori check:
 - Il primo usa il dot product per decidere se invertire il valore della normale e dello ior. Questo check è stato implementato facendo riferimento alla implementazione di *eval_refractive* presente in yocto
 - Il secondo usa il prodotto per il \sin_{θ} o la funzione reflectance (come specificata nella pagina web di riferimento) per decidere sull'utilizzo della riflessione o della rifrazione

Il risultato iniziale di questo rendering è il seguente:



Samples: 256 - Resolution: 720

Il risultato è già visibile ma all'interno delle sfere c'è molta perdita di dettaglio, ho deciso quindi di generare una seconda immagine con maggiore risoluzione e samples:



Samples: 512 - Resolution: 1080

CREDITS:

Il codice per la rifrazione è stato implementato seguendo la guida fornita nel readme, consultabile al sito: <https://raytracing.github.io/books/RayTracingInOneWeekend.html#dielectrics>. Il resto del rendering è stato effettuato con le sole funzionalità built-in di Yocto.

Extra Credit Medi

Your Own Shader - Wet / Metallized Shader.

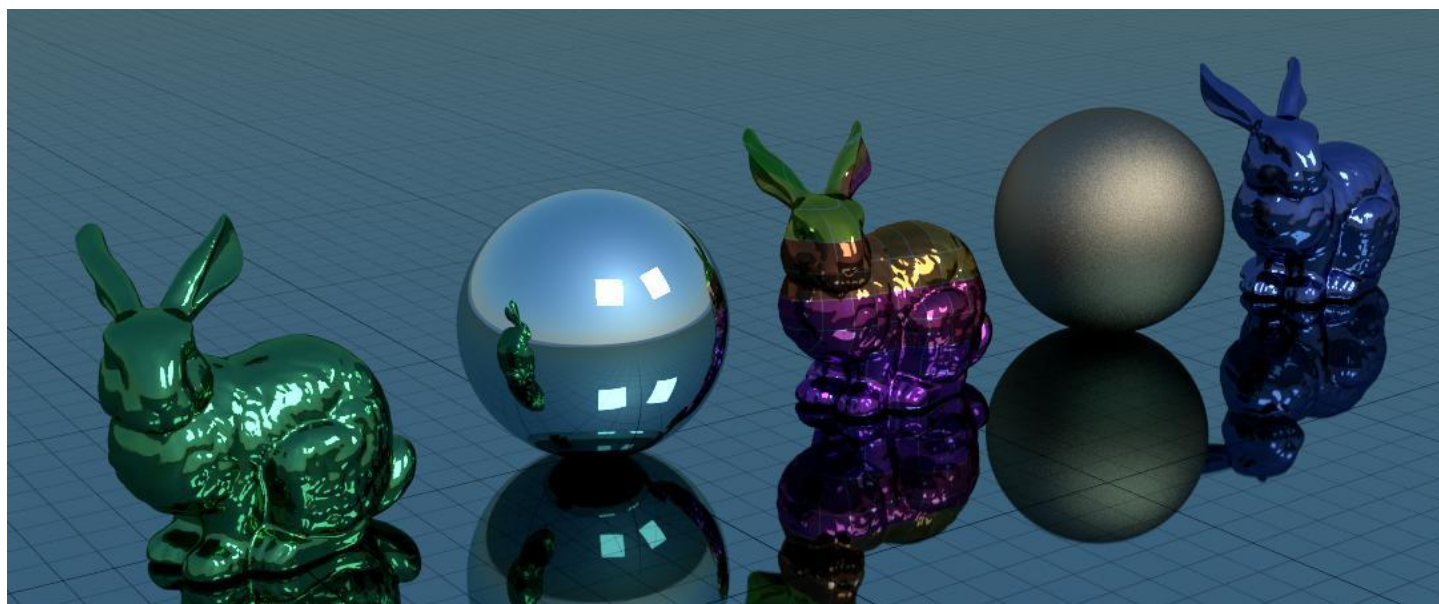
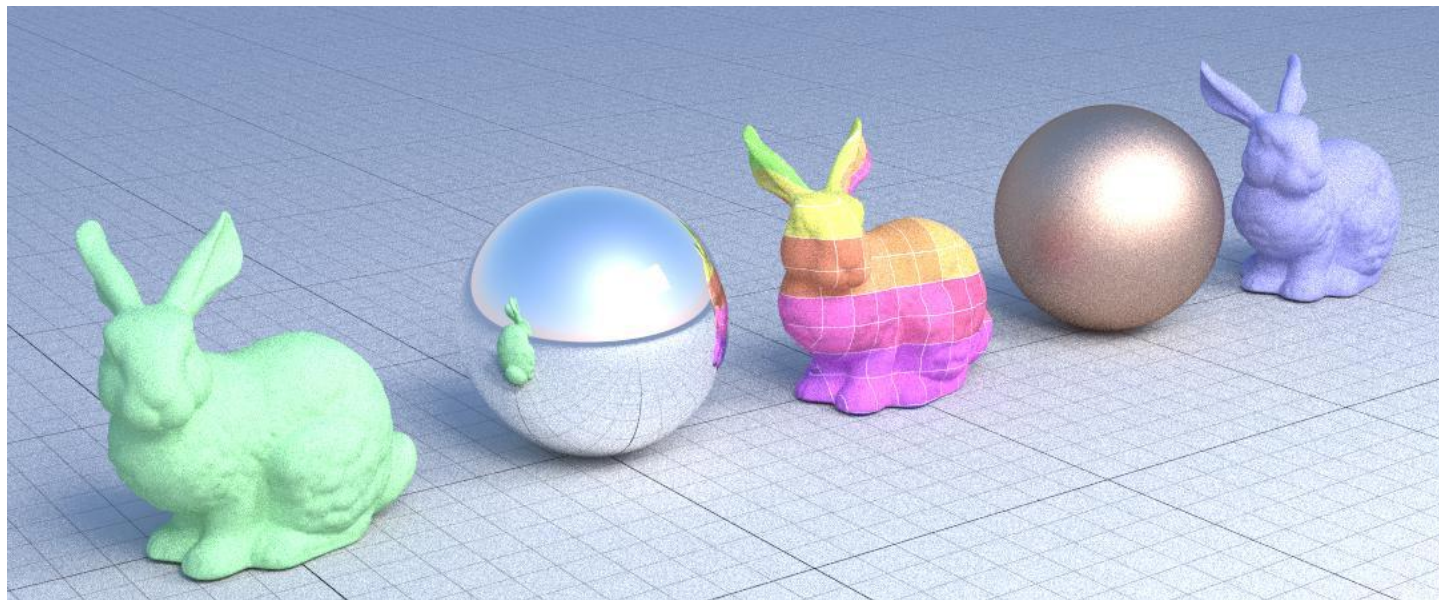
L'obiettivo di questo shader è quello di garantire a qualsiasi scena, a prescindere dal materiale usato, un effetto bagnato / metallizzato. Per accentuare ancora di più l'effetto bagnato ho cercato di ottenere uno scurimento generale della scena per dare l'impressione di un clima molto nuvoloso, come se avesse appena smesso di piovere. Ovviamente lo shader non cerca di puntare ad un effetto realistico al 100%, dato che sarebbero necessari molti altri step per ottenerlo, come ad esempio la realizzazione di piccole "macchie", come gocce che devono ancora essere asciugate. Data la natura di applicazione dello shader su qualsiasi materiale, le scene sono state comunque renderizzate attraverso la funzione principale *shade_raytrace*. Le modifiche dello shader wet sono controllate da un boolean *wet*, aggiunto in params e modificabile da linea di comando. Elenco di seguito alcuni dettagli di implementazione:

- Lo shader viene applicato prima della gestione dei materiali effettuata dal nostro render, su qualunque intersection, se il parametro *params.wet* è settato come *true* da linea di comando. In particolare, lo shader viene applicato su tutti i materiali **ad eccezione di quelli trasparenti**
- Per garantire un effetto bagnato/metallizzato, ogni intersezione gestisce un fattore di riflessione molto simile a quella per materiali reflective, in aggiunta alle altre eventuali riflessioni gestite da *shade_raytrace* sugli altri materiali, anche per motivi di semplicità di implementazione. Dopo alcuni test mi sono accorto che il vetro era particolarmente problematico e che avrebbe riflesso come un normale metallo, quindi è stato "escluso" dall'aggiunta di riflessione e subisce unicamente la variazione di colore dovuta agli altri oggetti della scena
- Come per i materiali reflective, viene calcolata una *wet_normal* attraverso la funzione *yocto sample_hemisphere_cospower*, con esponente basato sulla *roughness* del materiale
- La variazione di colore è calcolata attraverso diverse operazioni di saturation, smoothstep e mixing del colore del materiale, in relazione anche alla risoluzione dell'immagine in output
- Per migliorare l'effetto overall, il colore del pavimento viene calcolato con un fattore $0.75f$ invece che con un fattore $0.30f$ come il resto dei materiali della scena

Di seguito mostro alcuni risultati, tutti generati con samples: 512 e resolution: 1080, eccetto che per *hair*, generato con i valori standard samples: 256 e resolution: 720. Per ogni test riporto sia la versione senza shader wet generata dalla sola *shade_raytrace* e poi il risultato dell'applicazione di *shade_wet* sullo stesso testo, per comodità di comparazione.

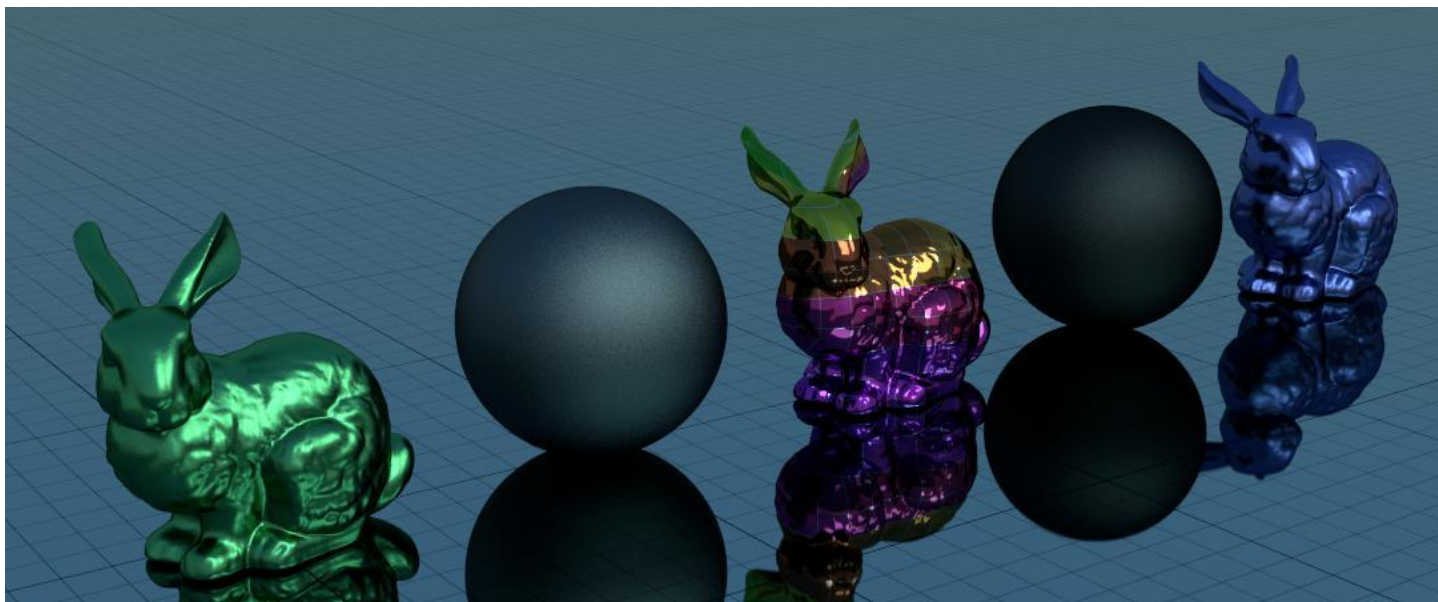
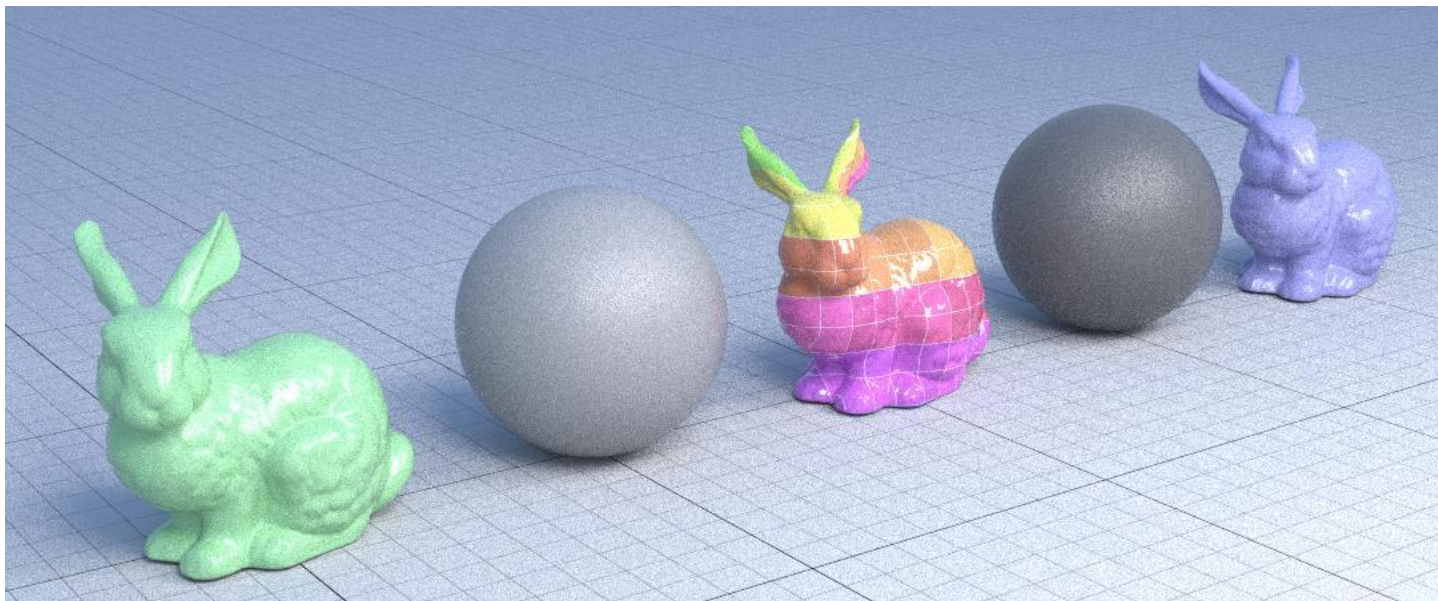
In ultimo vorrei evidenziare la velocità di elaborazione, estremamente vicina alle tempistiche di generazione delle immagini senza filtro wet, pertanto rendendo lo shader anche ragionevole in merito al fattore temporale.

METAL_vs WET_METAL



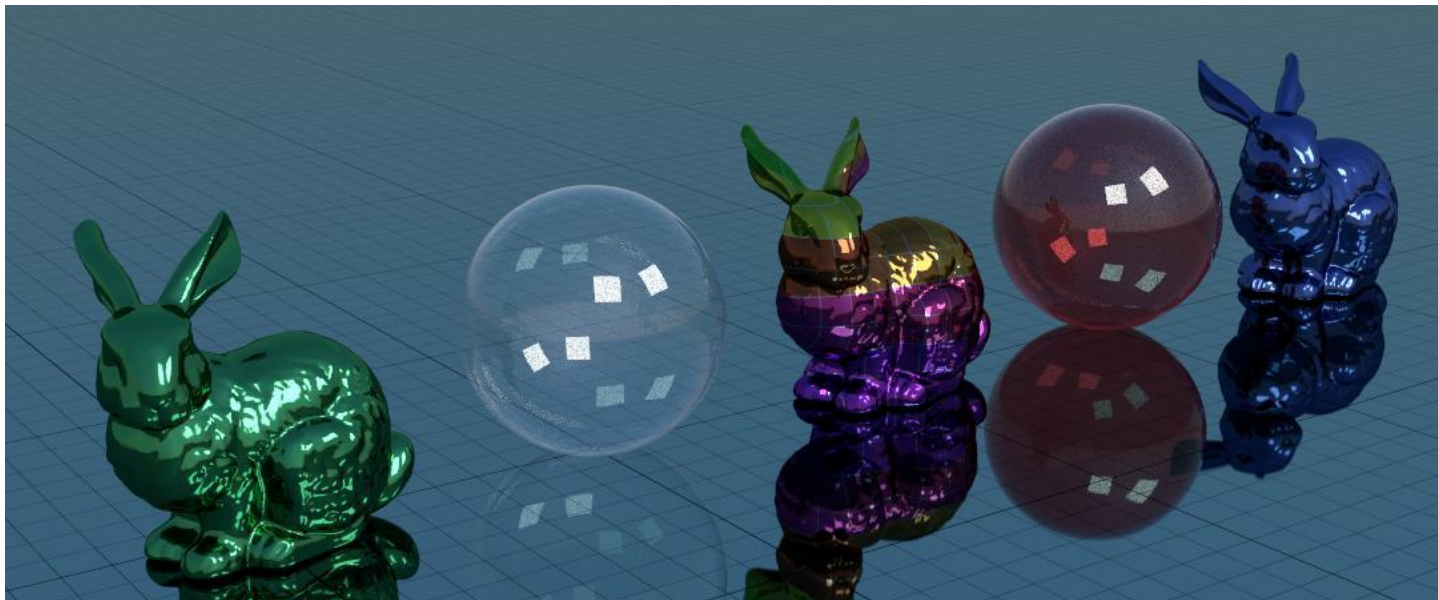
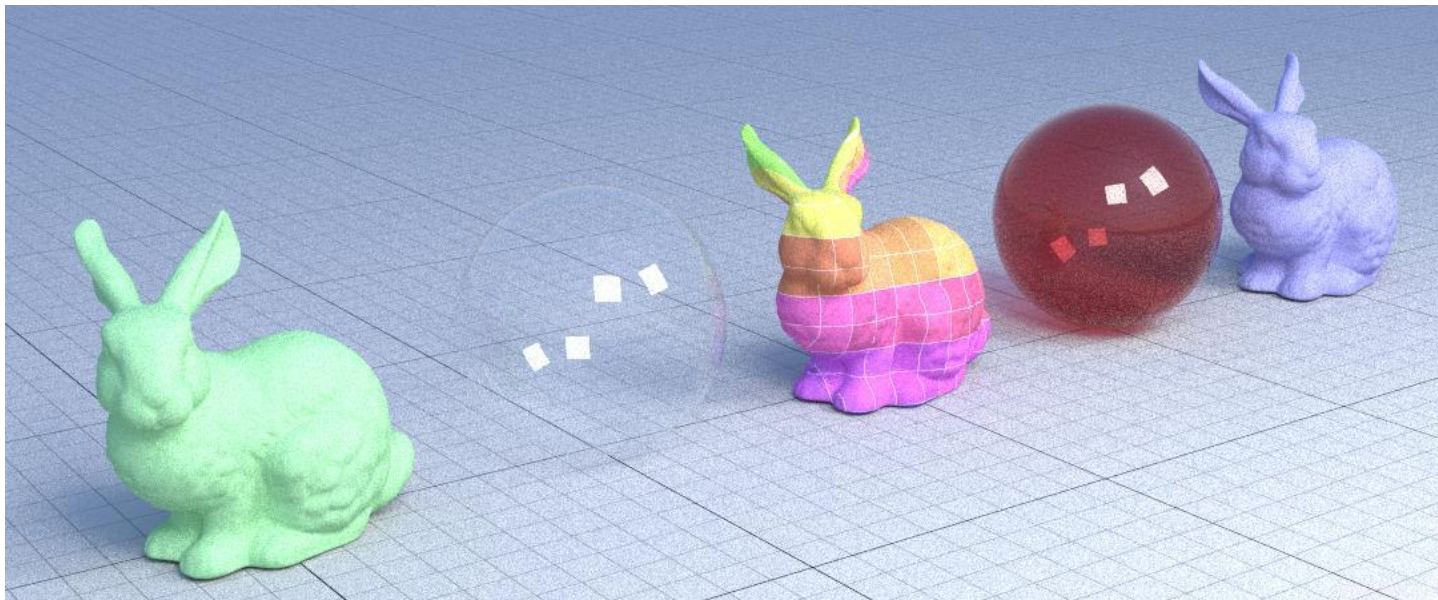
Per rendere l'effetto wet del pavimento è molto visibile il riflesso generato a prescindere dal materiale, nella scena infatti i conigli sono di materiale matte ma gestiscono comunque alcuni riflessi. Questo è particolarmente utile anche per rendere le piccole chiazze bianche presenti sui conigli per dare l'impressione di bagnato/lucidato.

PLASTIC vs WET PLASTIC



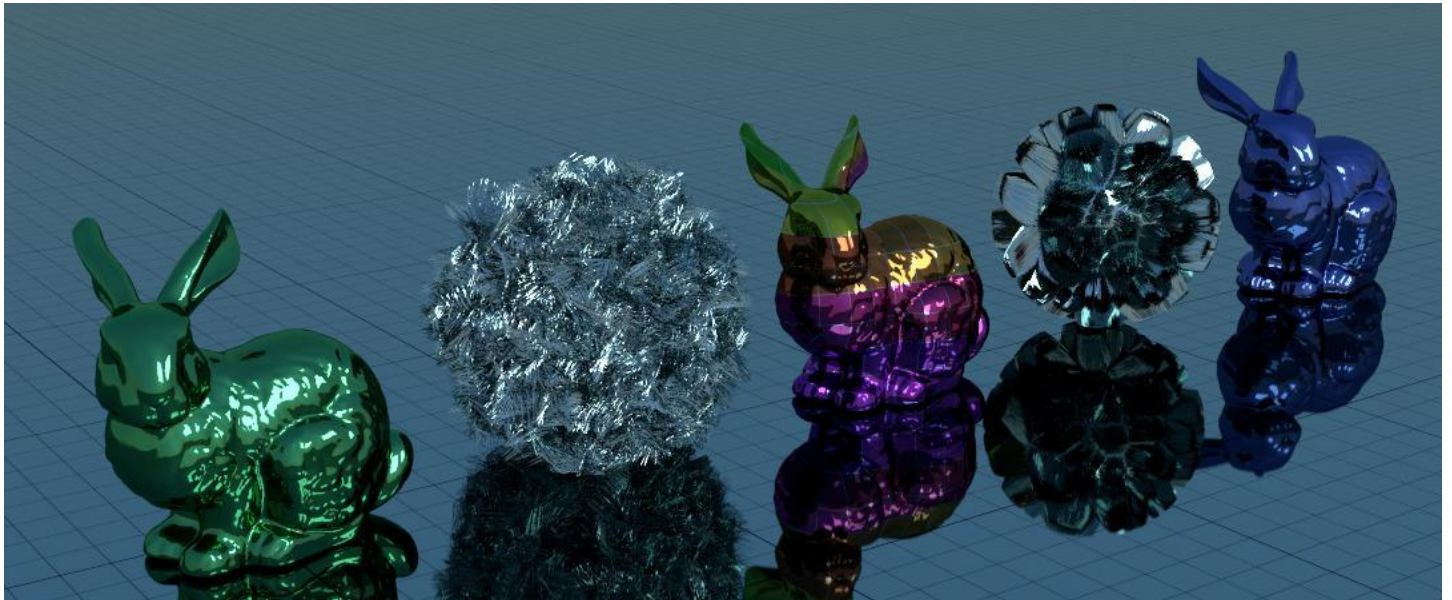
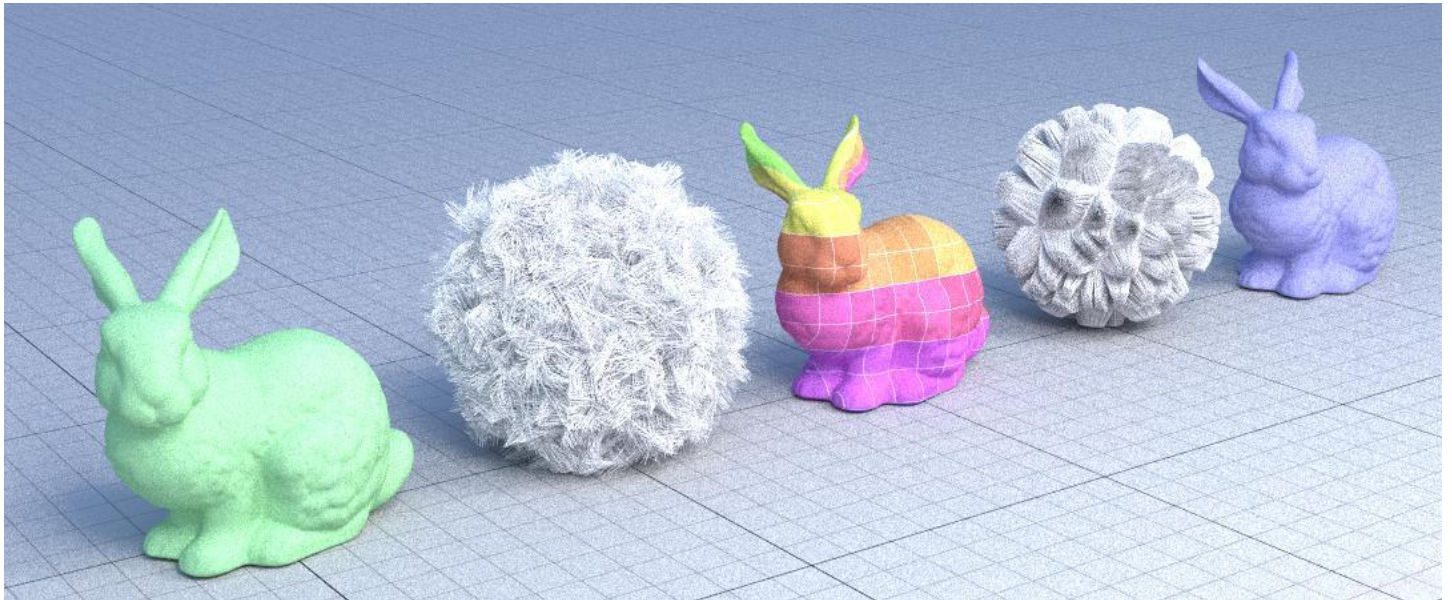
È visibile da questa scena dove le sfere non riflettono come in metal, una minore distribuzione delle sopra citate “chiazze” sui bunnies. Questo dimostra anche come l’effetto reagisca in base ai materiali di tutta la scena e non unicamente sulla singola istanza.

GLASS vs WET GLASS



Come specificato, il materiale trasparente non viene processato dallo shader, dato che l'applicazione del riflesso dovuto all'effetto *wet* porta a risultati molto "sgradevoli" e poco corretti. Notiamo comunque una leggera variazione di colore dovuta al resto della scena, che aiuta comunque a rendere la sensazione di bagnato.

HAIR vs WET HAIR



Il caso di *hair* è un po' più particolare, includo questo risultato perché mi è sembrato interessante analizzare il risultato dello shader sulle due sfere "di pelo" generate dal nostro renderer. Con la gestione dei riflessi e l'effetto *wet* infatti, più che bagnate, le due sfere sembrano più cristallizzate. Il risultato non è, a mio parere, totalmente spiacevole, in quanto potrebbe rendere questo shader adatto alla renderizzazione di alcune scene ghiacciate, come cave o simili, utilizzando la struttura *hair* per rendere meglio dei materiali cristallizzati.

CREDITS

L'idea per la creazione dello shader è stata presa dalla seguente pagina ShaderToy, con delle modifiche sostanziali. Credits all'utente **TDM** per l'implementazione: <https://www.shaderToy.com/view/ldSSzV>.

È stata principalmente sfruttata la modifica del colore applicata da TDM, in particolare le funzioni *saturation* e *mix*. Il resto dello shader è stato scritto da me, sfruttando i concetti di riflessione già sviluppati per i materiali reflective dell'homework.