



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria
ANNO ACCADEMICO OFFERTA	2016/2017
ANNO ACCADEMICO EROGAZIONE	2017/2018
CORSO DILAUREA MAGISTRALE	INGEGNERIA INFORMATICA
INSEGNAMENTO	INFORMATICA GRAFICA
TIPO DI ATTIVITA'	B
AMBITO	50369-Ingegneria informatica
CODICE INSEGNAMENTO	08978
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	PIRRONE ROBERTO Professore Ordinario Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	144
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	81
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	2
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	PIRRONE ROBERTO Mercoledì 11:30 13:00 Studio del docente, Edificio 6, terzo piano, stanza 3025

PREREQUISITI	Conoscenze di geometria e algebra vettoriale; conoscenze avanzate di algoritmi e strutture dati; programmazione C++, Java e Javascript; elaborazione di immagini. Si consiglia di aver almeno seguito le lezioni della materia "Architetture e progetto di sistemi web".
RISULTATI DI APPRENDIMENTO ATTESI	Conoscenza e capacita' di comprensione Lo studente, al termine del corso, avra' acquisito conoscenze e metodologie per affrontare e risolvere in maniera autonoma le problematiche legate allo sviluppo di software per la Computer Graphics. Lo studente conoscerà i fondamenti matematici della disciplina, apprenderà il concetto di pipeline grafica e come i differenti algoritmi s'inseriscano all'interno di questo flusso di lavoro ideale. Lo studente conoscerà come la pipeline grafica viene implementata nei dispositivi hardware di rendering. Lo studente conoscerà le diverse tipologie di rendering fotorealistico nonché le teorie fisiche che stanno alla base di questi processi e la loro possibile composizione. Lo studente conoscerà le teorie alla base della modellazione solida e i principi dell'animazione. Per il raggiungimento di quest'obiettivo il corso comprende: lezioni frontali, analisi e discussione di casi di studio legati a temi avanzati di computer grafica, seminari e dibattiti guidati su temi di ricerca. Per la verifica di quest'obiettivo l'esame comprende la verifica scritta sugli argomenti del programma, la discussione sui casi di studio presentati e il colloquio orale.
	Capacita' di applicare conoscenza e comprensione Lo studente conoscerà tutte le principali tecniche algoritmiche impiegate nel campo della grafica ed avra' competenze specifiche per quanto riguarda l'uso delle librerie grafiche WebGL, OpenGL ES Shading Language e CUDA in linguaggio C++ e Javascript. Egli sara' in grado di selezionare ed utilizzare gli strumenti e/o i linguaggi piu' idonei allo sviluppo delle soluzioni software piu' adatte alla tipologia dei vari problemi che si trovera' ad affrontare. Infine sara' introdotto alla conoscenza elementare di un framework completo per lo sviluppo e la distribuzione di applicazioni grafiche qual e' Unity 3D. Per il raggiungimento di quest'obiettivo il corso comprende: esercitazioni teoriche e di gruppo per sviluppo di applicazioni grafiche complesse nonché la presentazione e discussione in aula dei progetti e implementazioni. Per la verifica di quest'obiettivo l'esame comprende la verifica scritta mirata a consentire il riutilizzo originale degli elaborati software gia' analizzati durante le esercitazioni teoriche.
	Autonomia di giudizio Lo studente sara' in grado di svolgere un'analisi comparativa delle caratteristiche di differenti ambienti di sviluppo per la grafica in relazione alla soluzione di problemi specifici. Egli sara' in grado di affrontare problemi non strutturati e prendere decisioni in regime d'incertezza. Attraverso l'approccio metodologico acquisito durante il corso, egli potra' modellare problematiche complesse nell'ambito della computer grafica. Per il raggiungimento di quest'obiettivo il corso comprende: analisi e discussione su casi di studio legati a temi avanzati della computer grafica, presentazioni e discussioni in aula di progetti e implementazioni legati alle esercitazioni pratiche di gruppo. Per la verifica di quest'obiettivo l'esame comprende la discussione sui casi di studio presentati e su possibili varianti proposte dal docente durante l'esame orale.
	Abilita' comunicative Lo studente sara' in grado di comunicare con competenza e proprieta' di linguaggio problematiche complesse di Computer Graphics anche in contesti altamente specializzati. Per il raggiungimento di quest'obiettivo il corso comprende: esercitazioni di gruppo per sviluppo di applicazioni grafiche complesse, la presentazione e discussione in aula dei progetti e implementazioni, seminari e dibattiti guidati su temi di ricerca. Per la verifica di quest'obiettivo l'esame comprende la discussione sui casi di studio presentati e su possibili varianti proposte dal docente.
	Capacita' d'apprendimento Lo studente sara' in grado di affrontare in autonomia qualsiasi problematica concernente la computer grafica. Sara' in grado di approfondire tematiche complesse quali lo sviluppo di sistemi avanzati per l'animazione di scene, il trattamento di modelli geometrici di elevata dimensione e così via. Per il raggiungimento di quest'obiettivo il corso comprende: analisi e discussione su casi di studio legati a temi avanzati della computer grafica, esercitazioni di gruppo per sviluppo di applicazioni grafiche complesse, presentazioni e discussioni in aula di progetti e implementazioni legati alle esercitazioni pratiche di gruppo.

	Per la verifica di quest'obiettivo l'esame comprende: la verifica scritta mirata a consentire il riutilizzo originale degli elaborati software già analizzati durante le esercitazioni teoriche al fine di affrontare un nuovo tema applicativo, nonché la discussione sui casi di studio presentati e su possibili varianti proposte dal docente.
VALUTAZIONE DELL'APPRENDIMENTO	L'esame finale consta di due parti: la prima è una prova scritta ovvero la discussione del lavoro di gruppo. La seconda è un colloquio orale. Svolgono la prova scritta solo quegli studenti che, per esigenze personali, non riescono a garantire una frequenza assidua che consenta loro l'interazione proficua all'interno di un gruppo di lavoro. La prova scritta consta di 6 quesiti su argomenti riguardanti il programma teorico e di un quesito di programmazione WebGL. Ogni quesito teorico consente di acquisire da 0 a 3 punti secondo la seguente griglia di valutazione: •Punti 0: risposta assente o sbagliata •Punti 1: risposta appena sufficiente •Punti 2: risposta più che sufficiente, ma parziale •Punti 3: risposta corretta e completa Il quesito di programmazione consente di acquisire 12 punti che saranno attribuiti valutando i seguenti aspetti del codice prodotto: •Completezza •Sviluppo delle funzionalità grafiche principali •Originalità •Capacità di integrazione di codice già noto dalle esercitazioni teoriche •Sviluppo di funzionalità aggiuntive e/o di interfaccia. La prova scritta si intende superata se la valutazione è di almeno 18/30 ed è preclusiva dell'accesso alla prova orale. I partecipanti ai gruppi di lavoro non affrontano la prova scritta perché gli obiettivi formativi che andrebbero valutati tramite quest'ultima si intendono verificati attraverso la presentazione dei casi di studio e lo sviluppo dell'applicazione grafica proposta dal docente. Essi ricevono, in fase di presentazione, una valutazione complessiva in trentesimi che tiene conto, per quanto riguarda il codice sviluppato, degli stessi aspetti già presentati per la prova scritta. L'esposizione orale dei casi di studio viene valutata rispetto ai seguenti aspetti •grado di comprensione mostrato in relazione al programma teorico svolto •proprietà del linguaggio utilizzato •capacità di approfondimento del tema assegnato mediante letture autonome. Il colloquio orale tende a verificare le conoscenze dei temi esposti nel programma teorico svolto. Il voto finale risulta da una media delle valutazioni riportate nelle due parti dell'esame poiché queste coprono aspetti diversi ed egualmente importanti della preparazione dello studente.
OBIETTIVI FORMATIVI	L'esame finale consta di due parti: la prima è una prova scritta ovvero la discussione del lavoro di gruppo. La seconda è un colloquio orale. Svolgono la prova scritta solo quegli studenti che, per esigenze personali, non riescono a garantire una frequenza assidua che consenta loro l'interazione proficua all'interno di un gruppo di lavoro. La prova scritta consta di 6 quesiti su argomenti riguardanti il programma teorico e di un quesito di programmazione WebGL. Ogni quesito teorico consente di acquisire da 0 a 3 punti secondo la seguente griglia di valutazione: •Punti 0: risposta assente o sbagliata •Punti 1: risposta appena sufficiente •Punti 2: risposta più che sufficiente, ma parziale •Punti 3: risposta corretta e completa Il quesito di programmazione consente di acquisire 12 punti che saranno attribuiti valutando i seguenti aspetti del codice prodotto: •Completezza •Sviluppo delle funzionalità grafiche principali •Originalità •Capacità di integrazione di codice già noto dalle esercitazioni teoriche •Sviluppo di funzionalità aggiuntive e/o di interfaccia. La prova scritta si intende superata se la valutazione è di almeno 18/30 ed è preclusiva dell'accesso alla prova orale. I partecipanti ai gruppi di lavoro non affrontano la prova scritta perché gli obiettivi formativi che andrebbero valutati tramite quest'ultima si intendono verificati attraverso la presentazione dei casi di studio e lo sviluppo dell'applicazione grafica proposta dal docente. Essi ricevono, in fase di presentazione, una valutazione complessiva in trentesimi che tiene conto, per quanto riguarda il codice sviluppato, degli stessi aspetti già presentati per la prova scritta. L'esposizione orale dei casi di studio viene valutata rispetto ai seguenti aspetti

	•grado di comprensione mostrato in relazione al programma teorico svolto •proprietà' del linguaggio utilizzato •capacità' di approfondimento del tema assegnato mediante letture autonome. Il colloquio orale tende a verificare le conoscenze dei temi esposti nel programma teorico svolto. Il voto finale risulta da una media delle valutazioni riportate nelle due parti dell'esame poiché queste coprono aspetti diversi ed egualmente importanti della preparazione dello studente.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali; Esercitazioni teoriche; Creazione di gruppi di lavoro per l'analisi di casi di studio e per lo sviluppo di applicazioni grafiche complesse; Presentazione e discussione in aula dei risultati prodotti dai gruppi di lavoro; Dibattiti guidati in aula su temi di ricerca.
TESTI CONSIGLIATI	- Alan Watt, 3D Computer Graphics - Third Edition, Addison-Wesley Publishing Company, ISBN: 978-0201398557 - Alan Watt & Mark Watt, Advanced Animation and Rendering Techniques, Addison-Wesley Publishing Company, ISBN: 978-0201544121 - Diego Cantor and Brandon Jones, WebGL Beginner's Guide, Packt Publishing, ISBN: 978-1849691727 - Andreas Anyuru, Professional WebGL Programming, Wiley, ISBN: 978-1-119-96886-3 Materiale elettronico sul sito web del docente. Siti di riferimento per HTML5, WebGL, ESSL, CUDA: http://www.w3schools.com/html5/default.asp http://www.khronos.org/ http://www.khronos.org/webgl/wiki/MainPage http://learningwebgl.com/blog/ http://www.khronos.org/registry/gles/specs/3.0/GLSLESSpecification3.00.3.pdf http://www.khronos.org/opengles/sdk/docs/manglsl/ http://www.khronos.org/opengl.org/ http://developer.nvidia.com/object/gpucomputing.html Sito di Unity 3D http://unity3d.com

PROGRAMMA

ORE	Lezioni
1	Introduzione al Corso
2	Grafica Vettoriale e Raster, Pipeline di Rendering, Shaders e kernel per elaborazione parallela. Architettura e funzionamento delle moderne GPU.
4	Matematica per la Computer Graphics
5	Mesh di poligoni, curve e superfici parametriche
2	Modelli geometrici e loro proprietà
2	Trasformazioni geometriche bidimensionali e tridimensionali
3	Trasformazioni proiettive e Viewing
3	Algoritmi di determinazione delle superfici visibili e clipping tridimensionale
5	Algoritmi di illuminamento e di ombreggiatura, BRDF
3	Algoritmi per il tracciamento, antialiasing, riempimento ed il clipping sullo schermo
1	Trasformazioni spaziali di immagini
4	Algoritmi di visualizzazione realistica: applicazione di trame, mutue riflessioni tra oggetti, trasparenza
5	Algoritmi di Raytracing
5	Algoritmo di Radiosity
2	Principi di animazione
1	Rendering di volumi
ORE	Esercitazioni
2	Introduzione alla programmazione WebGL; Riepilogo su Javascript e DOM; cenni di HTML5 e dell'elemento <canvas> per il recupero del contesto grafico; Struttura della libreria WebGL e sue relazioni con gli altri ambienti grafici; ambienti di sviluppo per uso di WebGL.
2	Struttura di un'applicazione WebGL. Creazione di modelli e caricamento di modelli geometrici in WebGL tramite oggetti JSON.

ORE	Esercitazioni
2	Visualizzazione della scena in WebGL. Introduzione agli shader ESSL; scrittura di fragment shaders per shading di Gouraud e Phong; gestione di luci direzionali e posizionali.
2	Impostazioni della camera e matrici di proiezione: uso della libreria Javascript glMatrix.js e uso diretto delle trasformazioni di proiezione all'interno dei vertex shaders.
2	Trasformazioni di modellazione per animazione della scena.
2	Effetti fotorealistici: illuminamento da sorgenti multiple, depth testing, blending e trasparenza.
2	Texture mapping: gestione delle tessiture in WebGL e all'interno di uno shader; tecniche di filtraggio, mipmaps, warping di tessiture.
2	Introduzione al framework Unity 3D.
2	Introduzione alla libreria CUDA. Struttura dei kernel. Gestione della memoria.
ORE	Laboratori
4	Sviluppo di un'applicazione WebGL completa.
4	Introduzione alla libreria CUDA. Esempi di applicazione di kernel CUDA.
ORE	Altro
4	Discussione in aula di progetti e implementazioni preparati durante le esercitazioni di gruppo.
3	Seminari e dibattiti guidati su temi di ricerca e applicazione della computer grafica anche con la partecipazione di esperti del settore.