**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Редактор за брандиране на 3Д модели

Дипломант: Научен ръководител:

*Виолета Георгиева Кирил Митов*

СОФИЯ

2 0 2 1

**TЕХНОЛОГИЧНО УЧИЛИЩЕ**



**ЕЛЕКТРОННИ СИСТЕМИ**

**към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

Дата на заданието: 10.11.2020 г. Утвърждавам:..............................

Дата на предаване: 10.02.2021 г. /проф. д-р инж. Т. Василева/

**ЗАДАНИЕ**

**за дипломна работа**

на ученичката Виолета Георгиева Георгиева 12 в клас

1. Тема: "Инструмент за брандиране на 3Д модели"
2. Изисквания:   
   - Зареждане на празна сцена с възможност за качване на 3Д модел и 3Д лого  
   - Да може да се запаметява редактираната сцена и да се зарежда  
   - Да може моделите да се местят и завъртат  
   - Инструмент за изкривяване на плоско лого  
   - Инструмент за "долепяне" (snap) на двата модела  
   - Инструмент за съединяване на двата модела в един  
   - Генериране на 3Д модел за парче текст  
   - Инструмент за автоматично залепване на логото върху една или повече повърхнини на модел
3. Съдържание 3.1 Обзор

3.2 Същинска част

3.3 Приложение

Дипломант :...........................................

/Виолета Георгиева/

Ръководител:..........................................

/Кирил Митов/

Директор:................................................

/ доц. д-р инж. Ст. Стефанова /

Мнение на научния ръководител

Научен ръководител:

/Кирил Митов/

Увод

Глава Първа

1. Какво е уеб приложение и как се разработва Уеб приложение

1.1. Уеб приложение

Уеб приложенията са изключително популярни в днешно време и са един от най-използваните софтуерни продукти. Причината за това е, че не е неободимо те да бъдат изтеглени на нашето устройство, за да можем да ги използваме. Необходимо ни е само устройството, което използваме да има връзка с интернет и уеб браузър, без значение дали това е настолен компютър, лаптоп, таблет или смартфон. Лесно е тези условия да бъдат изпълнени предвид, че живеем в XII век.

Уеб приложението е компютърна програма, която се съхранява на отдалечен сървър и се предоставя на потребителя през интернет, чрез интерфейса на браузъра. Според редактора на “Web.AppStorm” Джарел Ремик, всеки компонент на уебсайта, който изпълнява някаква функция за потребителя, се квалифицира като уеб приложение.

Повечето уеб приложения са написани на програмни езици като JavaScript, HTML5 или Cascading Style Sheets (CSS). Client-side програмирането(или по-често наричаният front-end) обикновено използва тези езици, които помагат за изграждането на видимата част от приложенията за потребителя. Server-side програмирането (или по-често наричания back-end) се използва за създаване на скриптове, които ще използва уеб приложението, с помащта на програмни езици като Python, Java и Ruby. ¹

Server-side скриптовете се използват в back-end частта, където изходния код е невидим или скрит от страна на клиента (браузъра). От друга старана, client-side скриптовете от страна на клиента се използват в front-end частта, която потребителите могат да видят от браузъра.

Когато се обработва скрипт от страна на сървъра, той комуникира със сървъра. За разлика от това, скриптове от страна на клиента не се нуждаят от взаимодействие със сървъра. Скриптове от страна на сървъра са по-сигурни от скриптове от страна на клиента, тъй като скриптовете от страна на сървъра обикновено са скрити от клиента, докато скриптът от страна на клиента е видим за потребителите.

Клиентски скриптове и скриптове от сървър работят координирано помежду си. И двете техники за скриптиране обаче са много различни, където скриптове от страна на клиента наблягат на това интерфейсът на уеб приложението или уебсайта да стане по-привлекателен и функционален. И обратно, скриптове от страна на сървъра наблягат на методите за достъп до данни, обработка на грешки и т.н. ²

За разлика от десктоп приложенията, уеб приложенията нямат нужда от инсталация на компютър и не са зависими от операционната система на потребителя. Mогат да се ползват по всяко време, навсякъде, на всякакви устройства, които имат връзка с интернет. След като бъде разработена и качена нова версия на уеб сървъра, всички потребители могат да получат достъп до нея веднага, без нужда от обновяване. Ако приложението изисква повече ресурси, това засяга единствено хардуера на сървъра. Уеб апликациите предлагат по-голяма сигурност, тъй като нямат нужда от поддръжка на многобройни клиентски компютри, а единствено от мониторинг на уеб сървъра. Уеб базираният софтуер е по-лесен за ползване, разработка, поддръжка и надграждане. Почти всяка програма може да бъде реализирана като уеб базирано приложение.³

1.2.Разработка на уеб приложение

1.2.1. Framework (Софтуерна рамка)

Разработването на софтуерни продукти е възможно да бъде направено с помощта на framework(софтуерна рамка) или без, но не е нужно да започваме от нулата, когато има продукти, които могат да ни помогнат при разработката на проектите. Софтуерните рамки са софтуер, който се разработва и използва от разработчиците за изграждане на приложения. Използването на софтуерна рамка за разработване на приложения ни позволява да се съсредоточим върху функционалността на приложението на високо ниво. Това е така, защото за всяка функционалност на ниско ниво се грижи самата рамка. Софтуерните рамки улесняват живота на разработчиците, като им позволяват да поемат контрола върху целия процес на разработване на софтуер или по-голямата част от него от една платформа

Софтуерните рамки могат да включват: библиотеки; API​-та (това е съкращението на ​Application Programming Interface - Приложно-програмен интерфейс​, което е софтуерен посредник, който позволява предаване на данни между един софтуерен продукт и друг. Той също така съдържа условията на този обмен на данни); Scaffolding(техника, използвана от някои MVC рамки); AJAX (Някои JavaScript рамки са вградени в по-големи рамки, като включват бързата технология на AJAX във функционалността на сайта); Кеширане (което намалява натоварването на сървъра); Сигурност (чрез рамки за удостоверяване и упълномощаване); Компилатори (компилаторът е програма, която превежда изходната програма, написана на някакъв език за програмиране на високо ниво, в машинен код за някои компютърни архитектури. Генерираният машинен код може по-късно да бъде изпълнен много пъти срещу различни данни всеки път).

1.2.2. Рамки за уеб приложения(**Web application frameworks)**

Рамките за уеб приложения са софтуерни рамки, които се използват за рационализиране на разработването на уеб приложения и уеб сайтове, уеб услуги и уеб ресурси. Популярен тип рамка за уеб приложения е архитектурата Model-View Controller (MVC), наречена така заради начина, по който разделя кода за всеки компонент на приложението на модули

1.2.3. Софтуерна рамка – Библиотека (Framework vs library)

Някои може да приемат, че софтуерната рамка е колекция от библиотеки. Това обаче не е вярно, тъй като не всички софтуерни рамки използват или зависят от библиотеките. Разликата между библиотека и рамка е, че последната извиква кода. Противоположно на това кодът извиква софтуерната библиотека. Това е технически известно като Inversion of Control(IoC – Инверсия на контрола).

1.2.4. Език за програмиране – Софтуерна рамка

Програмният език казва на компютъра какво трябва да направи. Всеки език за програмиране има синтаксис и определен набор от правила, които трябва да се спазват всеки път, когато се пише код. Софтуерната рамка е изградена върху език за програмиране. Например, Rails, известен също като Ruby on Rails, е уеб рамка, изградена върху езика за програмиране Ruby. Django и Flask са две различни уеб рамки, изградени върху езика за програмиране Python. Следователно, те са известни и като рамки на Python. React и Angular са front-end уеб рамки, изградени върху езика за програмиране JavaScript

1.2.5. Модел-Изглед-Контролер (Model-View-Controller / MVC)

Model-View-Controller (MVC) е архитектурен шаблон, който разделя приложението на три основни логически компонента: модел, изглед и контролер. Всеки от тези компоненти е създаден, за да се справя със специфични аспекти на разработката на дадено приложение. MVC е една от най-често използваните индустриални рамки за уеб разработка за създаване проекти MVC разделя бизнес логиката и презентационния слой един от друг. Традиционно се използва за графични потребителски интерфейси. В днешно време MVC архитектурата стана популярна за проектиране на уеб приложения, както и на мобилни приложения. Визуализацията на тази архитектура може да се види на **фиг. 1**

1.2.5.1. Модел - Model

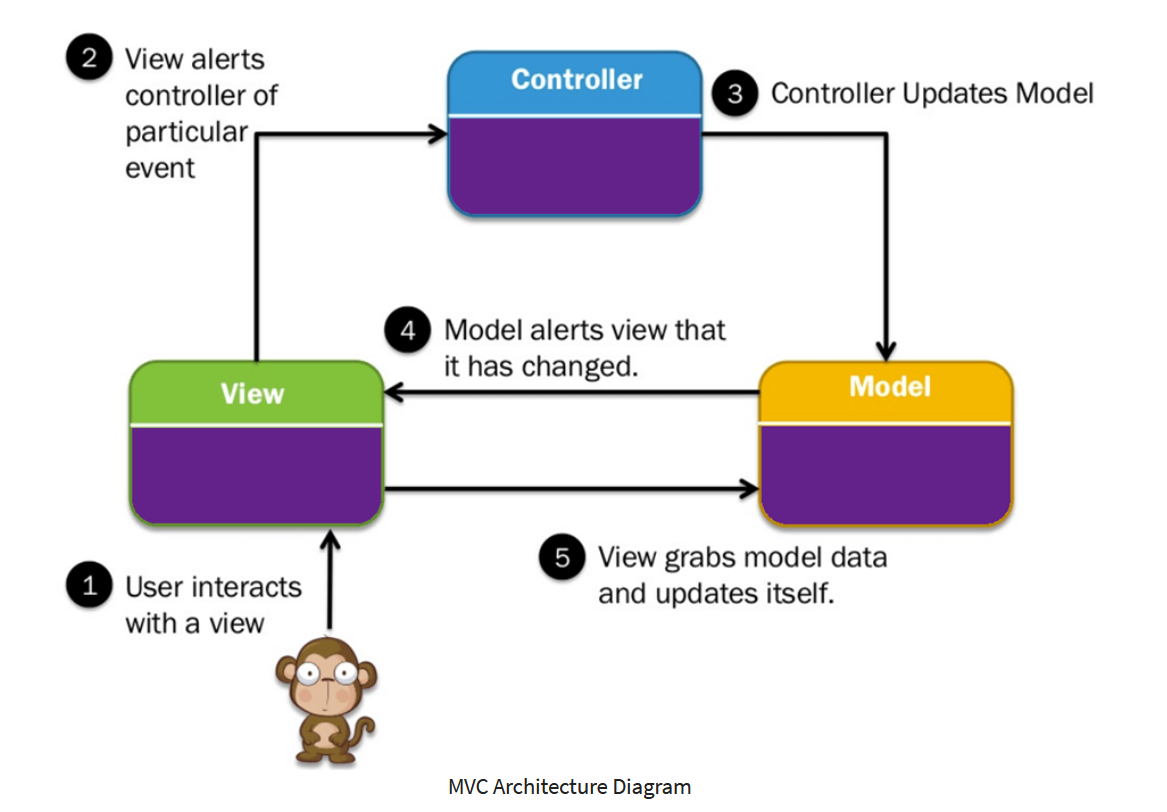
Компонентът Model отговаря на цялата свързана с данните логика, с която потребителят работи. Той представлява данни, които се прехвърлят между компонентите на контролера или друга свързана бизнес логика. Например обект Controller ще извлече информацията за клиента от базата данни. Той манипулира данните и ги изпраща обратно в базата данни или ги използва, за да изобрази същите данни. Той отговаря на заявката от изгледите(views) и също така отговаря на инструкциите на контролера за самообновяване. Това е и най-ниското ниво, което е отговорно за поддържането на данните

1.2.5.2. Изглед - View

Компонентът View се използва за цялата UI(user interface – потребителски интерфейс) логика на приложението. Например изгледът на клиента ще включва всички компоненти на потребителския интерфейс, като текстови полета, падащи менюта и т.н., с които крайният потребител взаимодейства

1.2.5.3. Контролер – Controller

Контролерите действат като интерфейс между компонентите Model и View, за да обработят цялата бизнес логика и входящите заявки, да манипулират данни, използвайки компонента Model, и да взаимодействат с Views, за да изобразят крайния изход. Например, клиентският контролер ще обработва всички взаимодействия и входове от изгледа на клиента и ще актуализира базата данни, като използва модела на клиента. Същият контролер ще се използва за преглед на данните на клиента



**фиг.1**

1.2.6. Приложно-програмен интерфейс - API

API е набор от програмен код, който позволява предаване на данни между един софтуерен продукт и друг. Той също така съдържа условията на този обмен на данни. API се състои от два компонента:

* Техническа спецификация, описваща опциите за обмен на данни между решенията със спецификацията, направена под формата на заявка за обработка и протоколи за доставка на данни
* Софтуерен интерфейс, написан по спецификацията, която го представлява

API служат за множество цели. Като цяло те могат да опростят и ускорят разработването на софтуер. Разработчиците могат да добавят функционалности от други доставчици към съществуващи решения или да създават нови приложения, използвайки услуги от трети страни доставчици. Във всички тези случаи специалистите не трябва да се занимават с изходния код, опитвайки се да разберат как работи другото решение. Те просто свързват своя софтуер с друг. С други думи, API служат като абстракционен слой между две системи, скривайки сложността и работните детайли на последната

1.3. Текстови редактори. Интегрирани среди за програмиране. Редактори на код.

Редакторите на код са софтуерът, който е създаден специално, за да помогне на разработчиците в кодирането. Те са текстови редактори, с някои допълнителни функции, за управление и редактиране на код. Редакторите са различни в зависимост от програмния език. Някои редактори поддържат един или два програмни езика, докато други поддържат множество програмни езици. Редакторите на кода не компилират кода, а просто ни позволяват да пишем и редактираме изходния код. Когато разработчиците пишат код с помощта на тези редактори, той се грижи за синтаксиса, отстъпа от началото на реда, довършването на думите и поставянето на скоби, така че да се съчетават.

Редакторите на код имат повече функционалности от текстовите редактори. Текстовите редактори не предлагат функции като подчертаване на синтаксиса и автоматични отстъпи. Също така редакторите на код не са IDE. IDE включва функционалности за отстраняване на грешки, генератори на код и много други сложни функционалности, които помагат на разработчиците, докато редакторите на код помагат на разработчиците в кодирането

1.3.1. Atom

Atom е стартиран от гениалните разработчици на Github, но сега е собственост на Microsoft, хубавото на Atom е, че е безплатен, с отворен код и също така много персонализиран, най-добрата част за atom е неговата функция Teletype, която позволява на няколко / множество разработчици да работят по един и същ проект, по един и същ код, от всяка част на света, на различно място.

Atom е малко подобен на Sublime Text, но много разработчици, които предпочитат Atom, се дължат на неговия прост интерфейс в сравнение с Sublime Text.

Atom може да се използва в 3 операционна система (OS), Windows, Mac и Linux, така че каквато и платформа да имате Atom, е налична.

1.3.2. Vim

Vim осигурява поддръжка за стотици езици за програмиране. В UNIX и Mac той е известен като vi. Последната му налична версия е 8.1.

Предимства - лек и бърз в сравнение с други редактори, той използва минимална памет на системата и се зарежда незабавно. Можете да намерите приставки за почти всичко, от версия до контрол до предпочитание на цветовете. Той е безплатен и е с отворен източник и може да се използва в терминален или GUI режим. Базиране е на клавиатурата и вероятно ще осъзнаете, че вече не се нуждаете от мишка, всичко е налично на клавишните комбинации.

Недостатъци - Обучението, ще ви отнеме известно време, за да го научите, и ще имате толкова много команди и преки пътища, които да запомните. Слаба поддръжка за инструментариума му. Не е много ефективен, когато става въпрос за механизмът за изрязване / поставяне.

1.3.3. Brackets

Модерен текстов редактор с отворен код, който разбира уеб дизайна.

С фокусирани визуални инструменти и поддръжка на препроцесор, Brackets е модерен текстов редактор, който улеснява разработването на дизайн в браузъра. Той е създаден от нулата за уеб дизайнери и разработчици от front-end-a.

Предимства - Open Source и Cross-Platform, което означава, че е безплатно и може да се използва в Mac, Windows и Linux. Той има автоматично довършване на кода, което ни дава автоматично предложения за допълване на нашия код. Уникална функция на скобата - бързото редактиране, когато маркирате който и да е HTML елемент и натиснете CMD / Ctrl + E, след това скобата ще отвори всички CSS, свързани с тези определени елементи в същия прозорец, след като го затворите, той автоматично ще се приложи към CSS таблица със стилове. Възможност за добавяне на няколко разширения и пакети, за да направите вашия редактор по-мощен и полезен. Редакторът също така има избор на цвят, тъй като цветните шестнадесетични кодове са трудни за запомняне.Наличен е и предварителен преглед на живо, за да видите лесно резултатите или резултатите от вашето редактиране.

Недостатъци – Съсредоточен е върху езиците за разработка на front-end, а не много от страна на езиците за сървъра.

1.3.4. Sublime Text

Много разработчици все още избрат да използват Sublime Text, защото неговата скорост, простота, както и клавишните комбинации, са много полезни при работа на различни скриптови езици, като Javascript.

Предимства - Open Source и Cross-Platform, което означава, че е безплатно и може да се използва в Mac, Windows и Linux. Автодовършване. Персонализируем - налични са над 3800 пакета. Лек, бърз и стабилен

Недостатъци - Все още няма приличен Git плъгин. Бавно развитие, тъй като sublime няма голям екип зад себе си. Не е толкова подходящ за начинаещи.

1.3.5. VSCode

Visual Studio Code предоставя на разработчиците нов набор от инструменти, който съчетава простотата и рационализирания опит на редактор на код с най-доброто от това, което разработчиците се нуждаят за техния основен цикъл за редактиране и отстраняване на грешки в кода. Visual Studio Code е първият редактор на код и първият инструмент за разработване на различни платформи - поддържащ OS X, Linux и Windows - в семейството на Visual Studio. Едно от предимствата на използването на VSCode е способността му да се интегрира с Git, това е лесно и бързо, друго нещо е способността му да се интегрира дистанционно в различни екипи, няколко разработчици могат да работят по един и същ проект или дори един и същ код с помощта на този инструмент: Visual Studio Live Share помага на няколко разработчици да работят заедно по-ефективно в сравнение с Sublime Text и Atom.

Предимства - Лек и бърз, безплатен, има прост и изчистен потребителски интерфейс, вграден терминал. IntelliSense - това е функция за автоматично довършване на суперзареждане, помощник за подчертаване на синтаксиса, за да се елиминират печатарските грешки и възможните синтаксисни грешки. Огромна общност за подкрепа и изграждане на разширения и допълнителни ресурси. Има най-добрите функции като Git, Terminal, WebView, Live Share. Вътрешна поддръжка за съвместна работа с Git, отворена директория лесно се свързва с Github и няма сложни настройки. Има разширение Emmet, супер бързо HTML, CSS и Javascript кодиране.

Недостатъци - Няма поддръжка за по-стара версия за програмиране като ASP.NET. Функциите за отстраняване на грешки са ограничени.

VSCode е най-популярният редактор.

1.3.6. Notepad++

Notepad ++ е безплатен текстов редактор за Microsoft Windows, който предоставя допълнителни функции, които не се намират в стандартния текстов редактор на Windows, Notepad. За първи път е издаден на 24 ноември 2003 г. от разработчика Дан Хо и все още е в активно развитие. Notepad ++ е специално проектиран за редактиране на изходния код. „++“ в името е препратка към оператора за увеличаване в програмни езици като C, C ++, Java и JavaScript. Notepad ++ е полезен по всяко време, когато трябва да направите значителни промени в текстов файл

1.4. Софтуерни рамки и библиотеки за графичен интерфейс

Най-забележимото за потребителя в едно уеб приложение е графичния дизайн. Изграждането на собствен дизайн от нулата е трудна и времеемка задача, точна поради тази причина се използват библиотеки и софтуерни рамки. Освен софтуерни рамки за уеб приложения има и множество софтуерни рамки за графичен интерфейс, които позволяват използването на готови силове на шрифтове и оразмеряване, които варират. Предварително изградени компоненти на уеб сайта като странични панели, бутони и навигационни ленти. В зависимост от рамката има множество токива, които са способни на много повече.

Примери за Front-End Frameworks:

1.4.1. Bootstrap

Bootstrap е най-популярниата рамка. Той има много звезди в Github и много ресурси. Един от най-лесните за използване, но някои казват, че има много характерен „Bootstrap“-ски външен вид.

1.4.2. Foundation

Предлага много гъвкавост и адаптивност. Добре за тези, които са опитни в разработката на front-end и обичат да обхващат основите, като същевременно запазват много творчески контрол.

1.4.3. Stylus

Експресивен и стилен CSS език. Тази рамка може да се използва само за Node.js приложения.

1.4.4. Semantic UI

Кратък, интуитивен и прави отстраняването на грешки в кода приятно и лесно. Дава много свобода на дизайна и се адаптира към нуждите на разработчика.

1.4.5. UI Kit

Рамката е подходяща при разработването на iOS приложения.UI Kit има основен стил, който улеснява разработването на собствен външен вид.

Изборът на софтуерна рамка зависи на първо място от нашите собствени умения и ако сме начинаещи е по-добре да изберем библиотека като Bootstrap. Ако сте по-опитни, може би ще ви е по-добре с по-проста рамка, която ви дава много място за персонализиране. Другото важно нещо, за което трябва да гледаме е така наречения Responsive design – Адаптивен дизайн. Всеки сайт трябва да се визуализира правилно на всички устройства, тъй като все повече холра получават достъп до интернет чрез мобилни устройства. Според StarCounter към декември 2020г. мобилният трафик е надминал трафика от настолни компютри с 55,73% от трафика, идващ от мобилни устройства. Затова е добра практика да се придържаме към фронтенд рамки, които поддържат отзивчив уеб дизайн, така че да имаме едно нещо по-малко, за което да се притесняваме. Не по-малко важен е и външния вид, затова е хубаво да си придържаме към рамка, която ни позволява да постигнем желания външен вид с възможно най-малко усилия.

Повече от всичко, правилната интерфейсна рамка опростява, рационализира и ускорява процеса на проектиране и разработване на уеб приложението, като същевременно ни дава гъвкавостта и характеристиките, необходими ни за постигане на изключителни резултати.

1.5. Преглед на съществуващи подобни програмни систими/продукти – 3,4 стр blender-1, online editor близки до темата

Как се разработво един 3д редактор кои са библиотеките webgl

* 1 стр
* 1 стр three.js
* 1 babylon.js
* 1 друг допълнителен редактор
* Каква е целта на дипломната- осн изисквания
* 1 стр програмни езици и среди с които се разработват горните неща
* 1 стр 3д като цяло геометрия ветрекс материал текстура сцена куатърньон и
* 1 осн формати .obj и др 3д формати

3 глава какво сме направили

4 показване на приложението

Представлява проучвателната част на ДР. Прави се преглед на съществуващи подобни програмни системи и продукти и преглед на известните развойни средства и среди ( максимум 8-10 страници ).

¹ What is web app? - <https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>

² Difference Between Server-side Scripting and Client-side Scripting –

<https://techdifferences.com/difference-between-server-side-scripting-and-client-side-scripting.html>

## ³ Предимства на уеб базирания софтуер- <https://www.simdim.com/services/web-app>

What is Framework - <https://hackr.io/blog/what-is-frameworks>

What is API - <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>

What is a compiler - <https://lambda.uta.edu/cse5317/notes/node3.html>

What is framework - <https://www.upwork.com/resources/what-is-a-framework>

MVC - <https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm>

MVC - <https://www.guru99.com/mvc-tutorial.html>

Code editor - <https://www.softwaretestinghelp.com/best-code-editor/>

# The 5 Best Code Editors For Web Development - <https://medium.com/for-self-taught-developers/the-5-best-code-editors-for-web-development-deaaa0b68fd5>

What is Notepad++ - <https://www.computerhope.com/jargon/n/notepad-plus-plus.htm>