



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

**Jihočeská univerzita v ČB**

Pedagogická fakulta

Katedra informatiky

**Webové 3D aplikace s využitím open-source  
frameworku Babylon.js The use of open-source  
framework Babylon.js for creating 3D web  
application Teze bakalářské práce**

**Vypracoval:** Filip Chytra

**Vedoucí práce:** PaedDr. Petr Pexa, Ph.D.

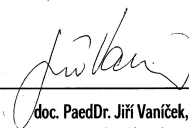
České Budějovice 2021



Datum zadání bakalářské práce: 7. dubna 2021  
Termín odevzdání bakalářské práce: 30. dubna 2022



doc. RNDr. Helena Koldová, Ph.D.  
děkanka



doc. PaedDr. Jiří Vaníček, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 7. dubna 2021

## Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 8. června 2023.

Filip Chytra

*Podpis*

## Abstrakt / Anotace

Cílem Tého bakalářské práce bude představení tvorby 3D modelů/scén a jejich vkládání do webových aplikací za pomoci open-source frameworku Babylon.js. V dnešní době se stále častěji setkáváme s využíváním trojrozměrných technologií, virtuálních a rozšířených realit (VR a AR) napříč mnoha medii a odvětvími jako jsou například kinematografie, herní průmysl, výcvikové trenažéry aj. Podobný trend implementace těchto technologií je předpokládán v následujících pár letech i u webových technologií a je podmíněn vývojem nových webových standardů a JavaScriptových API, jako jsou OpenGL, WebGL a v neposlední řadě nejnovější WebGPU, které umožňují využití fyziky a zpracování obrazu a efektů zrychlených na GPU jako součást plátna webové stránky. Autor se ve své práci zaměří na popis funkce samotného 3D enginu Babylon.js a porovnání s dalšími softwary pro tvorbu 3D počítačové grafiky a na její implementování do kódu webu. Základem práce bude na jednoduchých ukázkách důkladně demonstrovat možnosti základních nástrojů a funkcí pro založení scény, vytvoření základních objektů (světlo, kamera, polygonové sítě) a práci s materiály, ale také ukázání pokročilejších technik jako jsou průnik a sloučení objektů, animace, práci s uživatelskými vstupy, fyzikální engine, kostry, partikly a shadery. Cílem praktické části pak bude vytvoření vlastní 3D webové prezentace, na které bude demonstrováno vkládání jednotlivých praktických ukázek do kódu stránek v rámci HTML5 a popsáním možných uplatnění frameworku Babylon.js na webu.

## Klíčové slova

BABYLON.JS, 3D render, JavaScript, framework, HTML5, canvas, API, WebGL

# Abstract

The goal of the thesis is to introduce the creation of 3D models/scenes and their implementation into applications using the open-source framework Babylon.js. In recent years, we are increasingly encountering the use of three-dimensional technologies, virtual and augmented realities (VR and AR) across many media and industries such as cinema, gaming, training simulators, etc. A similar trend in the implementation of these technologies is expected in the next few years web technologies and is conditioned by the development of new web standards and JavaScript APIs, such as OpenGL, WebGL and last but not least the latest WebGPU, which allow the use of physics and image processing and GPU-accelerated effects as part of a web page canvas. The author will focus on the description of the function of the 3D engine Babylon.js itself and comparison with other software for creating 3D computer graphics and its implementation in the web code. The work will be based on simple demonstrations to thoroughly demonstrate the possibilities of basic tools and functions for creating a scene, creating basic objects (light, camera, meshes), and working with materials, but also demonstrating more advanced techniques such as collisions and intersections of objects, animation, working with user inputs, physics engine, skeletons, particles, and shaders. The aim of the practical part will be to create your own 3D web presentation, which will demonstrate the implementation of individual practical examples into the code of pages within HTML5 and a description of possible utilization of the Babylon.js framework on the web.

# Keywords

BABYLON.JS, 3D render, JavaScript, framework, HTML5, canvas, API, WebGL

# Obsah

<b>1</b>	<b>Úvod</b>	<b>10</b>
1.1	Východiska práce . . . . .	10
1.2	Cíle práce . . . . .	11
1.3	Metody práce . . . . .	12
<b>2</b>	<b>Trojrozměrná počítačová grafika</b>	<b>13</b>
2.1	Vývoj . . . . .	13
2.2	Základní procesy tvorby . . . . .	14
2.3	Uplatnění technologie . . . . .	14
<b>3</b>	<b>3D webové aplikace</b>	<b>15</b>
3.1	Počátky 3D technologií pro web . . . . .	15
3.1.1	Přínos . . . . .	16
3.1.2	Nevýhody . . . . .	16
3.2	Současné standardy pro tvorbu 3D webových aplikací . . . . .	17
3.2.1	HTML5 a canvas element . . . . .	17
3.2.2	WebGL . . . . .	17
3.2.3	WebXR . . . . .	17
3.2.4	WebGPU . . . . .	18
3.2.5	High level frameworky . . . . .	18
<b>4</b>	<b>BabylonJS</b>	<b>19</b>
4.1	Historie . . . . .	19
4.2	Alternativy . . . . .	20
4.2.1	ThreeJS . . . . .	20
4.2.2	A-Frame . . . . .	20
4.2.3	Unity WebGL . . . . .	20
<b>5</b>	<b>Bybylon.js Rendering engine</b>	<b>21</b>
5.1	Vykreslovací jádro . . . . .	21

5.2	Graf sceny . . . . .	21
5.3	Materiály a textury . . . . .	21
5.4	Animační systém . . . . .	22
5.5	Fyzikální engine . . . . .	22
5.6	Input systém . . . . .	22
5.7	Zvukový systém . . . . .	22
<b>6</b>	<b>Architektura frameworku</b>	<b>22</b>
6.1	Základní moduly . . . . .	23
6.2	Doplňkové moduly . . . . .	24
6.3	. . . . .	25
<b>7</b>	<b>Online nástroje</b>	<b>26</b>
7.1	Playground . . . . .	26
7.1.1	Editor kódu . . . . .	26
7.1.2	Náhled 3D scény . . . . .	27
7.1.3	Ukládání změn . . . . .	28
7.1.4	Export a import . . . . .	28
7.2	Sandbox . . . . .	29
7.3	Node Material Editor . . . . .	30
<b>8</b>	<b>Implementace</b>	<b>32</b>
8.1	Babylon.js Viewer . . . . .	32
8.1.1	Přidání Vieweru do projektu . . . . .	33
8.2	Export HTML souboru z nástroje Playground . . . . .	33
8.2.1	Export z nástroje Playground . . . . .	34
8.2.2	Přidání do projektu . . . . .	35
8.3	Přímá implementace frameworku . . . . .	35
8.3.1	CDN . . . . .	36
<b>9</b>	<b>Inspektor</b>	<b>38</b>



10	Nástin osnovy kapitol práce	40
11	Seznam přečtené literatury a zdrojů	43

# 1 Úvod

Bakalářská práce se bude zabývat open-source frameworkem Babylon.js, který umožňuje na moderních internetových prohlížečích vytvářet a vkládat scény s trojrozměrnými modely.

## 1.1 Východiska práce

Možnost vkládání různých formátů, rastrové či vektorové grafiky, videí a animací nepomohla k podobě webů jak je známe teď. Ovšem v poslední době se nabízí otázka, jakým směrem se vývoj webů vydá dále. Jedna z možností, která se nabízí převážně v posledních letech, a jejíž úspěch byl možný být pozorován i v dalších odvětvích, je rozšíření webu o trojrozměrné technologie. Díky příchodu nových technologií jako WebGL do moderních prohlížečů, výpočetní výkonnosti dnešních zařízení a rychlého gigabitového internetu tak jediné co zatím stálo v cestě nástupu 3D grafiky na scénu moderních stránek bylo jednoduchý, intuitivní a uživatelsky přívětivý způsob implementace pro vývojáře. A právě tyto poslední zásadní faktory zastupuje JavaScriptový open-source framework Babylon.js.

## 1.2 Cíle práce

Cílem Tého bakalářské práce bude představení tvorby 3D modelů/scén a jejich vkládání do webových aplikací za pomoci open-source frameworku Babylon.js. V dnešní době se stále častěji setkáváme s využíváním trojrozměrných technologií, virtuálních a rozšířených realit (VR a AR) napříč mnoha medii a odvětvími jako jsou například kinematografie, herní průmysl, výcvikové trenažéry aj. Podobný trend implementace těchto technologií je předpokládán v následujících pár letech i u webových technologií a je podmíněn vývojem nových webových standardů a JavaScriptových API, jako jsou OpenGL, WebGL a v neposlední řadě nejnovější WebGPU, které umožňují využití fyziky a zpracování obrazu a efektů zrychlených na GPU jako součást plátna webové stránky. Autor se ve své práci zaměří na popis funkce samotného 3D enginu Babylon.js a porovnání dalšími softwary pro tvorbu 3D počítačové grafiky a na její implementování do kódu webu. Základem práce bude na jednoduchých ukázkách důkladně demonstrovat možnosti základních nástrojů a funkcí pro založení scény, vytvoření základních objektů (světlo, kamera, polygonové sítě) a práci s materiály, ale také ukázání pokročilejších technik jako jsou průnik a sloučení objektů, animace, práci s uživatelskými vstupy, fyzikální engine, kostry, partikly a shadery. Cílem praktické části pak bude vytvoření vlastní 3D webové prezentace, na které bude demonstrováno vkládání jednotlivých praktických ukázek do kódu stránek v rámci HTML5 a popsáním možných uplatnění frameworku Babylon.js na webu.

### 1.3 Metody práce

Úvodem bakalářské práce popíši jednotlivé potřebné technologie, které stojí za umožněním vykreslování trojrozměrných scén na webových stránkách. Dále se přímo zaměřím na popsání jaké úlohy v tomto procesu zástava framework Babylon.js a na popsání jeho funkcí. V další části se zaměřím na způsoby tvorby 3D scén a objektů v prostředí Babylonu. Uvedu jaké jsou jeho výhody a omezení oproti jeho alternativám a popíši, jak následně scénu implementovat do HTML kódu. Základem praktické části pak již bude důkladná demonstrace jednotlivých funkcí a nástrojů, ale také pokročilejších funkcí a technik, které framework poskytuje. Veškeré funkce pak budou představeny na jednotlivých ukázkách na vlastní webové prezentaci, a následně vytvořím ukázkou možné reálné 3D webové aplikace s implementováním vlastní trojrozměrné grafiky vytvořené kombinací jednotlivých technik a funkcí.

## 2 Trojrozměrná počítačová grafika

Trojrozměrnou počítačovou grafikou se rozumí vytváření, manipulace a zobrazování trojrozměrných objektů pomocí specializovaného počítačového softwaru a hardwaru. 3D grafika zahrnuje reprezentaci virtuálního objektu nebo scény v počítači jako soubor geometrických tvarů a povrchů, které lze vykreslit a vytvořit tak realisticky vypadající obraz.

### 2.1 Vývoj

Historie 3D počítačové grafiky sahá až do 60. let 20. století, kdy vědci začali experimentovat s wireframe neboli drátovými modely pro vytváření 3D objektů. V průběhu let byly vyvinuty různé metodiky, včetně polygonového modelování, dělení povrchů a modelování skulptur, pro vyjmenování alespoň několika z nich.

V současné době je 3D počítačová grafika vysoce specializovanou oblastí informačních technologií, která vyžaduje odborné znalosti algoritmů počítačové grafiky, geometrie, lineární algebry a počítačového hardwaru. Vývoj nových technik a algoritmů pro vytváření a vykreslování 3D grafiky je aktivní oblastí výzkumu a inovací.

Mezi nejnovější vývoj v této oblasti patří trasování světelných paprsků v reálném čase, volumetrické vykreslování a přístupy založené na strojovém učení.

## 2.2 Základní procesy tvorby

Proces vytváření 3D počítačové grafiky zahrnuje několik kroků, včetně 3D modelování, texturování, nasvícení a vykreslování (renderování). Trojrozměrné modelování zahrnuje vytvoření virtuálních objektů definováním jejich tvaru, velikosti a struktury. Proces texturování zahrnuje přidání povrchových charakteristik, jako je barva a textura, do 3D objektů. Osvětlení zahrnuje přidání zdrojů světla do virtuální scény a jejich umístění tak, aby vytvořily požadovaný efekt. A konečně vykreslování zahrnuje použití matematických algoritmů k převodu 3D objektů na 2D obrázky, které lze zobrazit na obrazovce.

## 2.3 Uplatnění technologie

3D počítačová grafika se používá v široké škále oblastí, včetně videoher, filmů, architektonické vizualizace, produktového designu, vědecké vizualizace a virtuální reality. Používání 3D grafiky způsobilo revoluci v mnoha odvětvích tím, že umožnilo vytvářet realistická a pohlcující virtuální prostředí, která nebyla s tradiční 2D grafikou možná.

## 3 3D webové aplikace

Vzestup internetu výrazně ovlivnil způsob, jakým zacházíme s technologiemi. S příchodem webu 3.0 se internet změnil z pouhého zdroje informací na platformu pro imerzní zážitky. Jedním z nejvýznamnějších trendů posledních let v této oblasti je vznik 3D webových aplikací.

3D webové aplikace jsou webové aplikace, které využívají 3D grafiku a umožňují uživatelům interakci s 3D prostředím v reálném čase. Tyto aplikace mají potenciál revolučně změnit způsob, jakým pracujeme, učíme se a hrajeme, jelikož poskytují interaktivní, pohlcující zážitek, který je dostupný odkudkoli na světě.

Samotný render ve webovém prostředí se týká procesu vytváření a zobrazování trojrozměrného obrazu nebo animace pomocí webových technologií a standardů.

### 3.1 Počátky 3D technologií pro web

Mezi některé z významných předchůdců moderních technologií, které vydláždily cestu moderním 3D webovým aplikacím patří brzký standard VRML (Virtual Reality Modeling Language). Dále Java Applet, který umožnil vývojářům vkládat interaktivní 3D grafiku a animace přímo do webových stránek pomocí programovacího jazyka Java, jenž je následně spouštěn v prostředí Virtual Java Machine. A v neposlední řadě pak také za zmínku stojí multimediální softwarové platformy Adobe Shockwave a Flash Player, které vývojářům umožňovaly vytvářet interaktivní animace, hry a další multimediální obsah, který bylo možné vkládat přímo do webových stránek.

### 3.1.1 Přínos

Tyto technologie přinesly na web nové možnosti a umožnily vývojářům vytvářet interaktivní a dynamický obsah, který dříve nebyl možný pouze pomocí HTML a CSS. To podpořilo experimentování a inovace při vývoji webových stránek. Umožnily vytvářet interaktivní aplikace a hry na webu, což přineslo poutavější uživatelský zážitek

### 3.1.2 Nevýhody

Jednalo se o dodatečný proprietární software a technologie, jejich vývoj byl řízen vždy jedinou společností nebo organizací. To omezovalo jejich propojitelnost a ztěžovalo vytváření konzistentních aplikací v různých prohlížečích a zařízeních. Často byly kritizovány za problémy s výkonem. Dalším problémem pak bylo mnoho možných bezpečnostní chyb, které mohly ohrozit data a soukromí uživatelů. Z tohoto důvodu mnoho webových prohlížečů jejich podporu ve výchozím nastavení zakázalo nebo zcela odstranilo.



## 3.2 Současné standardy pro tvorbu 3D webových aplikací

Moderní přístupy k vytváření 3D webových aplikací jsou proto často postaveny na několika webových standardech a technologiích jejíž otevřenost a vzájemná kompatibilita umožňuje usnadňuje vývojářům webových aplikací vytvářet konzistentní webové aplikace, které fungují na široké škále zařízení a platform. Mezi hlavní webové standardy a technologie pro tvorbu 3D webových aplikací patří:

### 3.2.1 HTML5 a canvas element

HTML5 je nejnovější verze hypertextového značkovacího jazyka, který se používá pro strukturování webových stránek a začlenění 3D grafiky na webovou stránku. V této verzi byl představen `<canvas>` element, který poskytuje nízkoúrovňový rastrový kreslicí povrch, který lze použít k vytváření složité 2D a 3D grafiky, včetně vykreslování 3D scén prostřednictvím WebGL.

### 3.2.2 WebGL

Rozhraní API jazyka JavaScript pro vykreslování interaktivní 3D a 2D grafiky v kompatibilních webových prohlížečích. Poskytuje nízkoúrovňový přístup ke grafickému procesoru a umožňuje rychlé vykreslování složitých 3D scén. I přes širší podporu starší verze standardu WebGL 1.0 napříč verzemi webových prohlížečů je v současné době pro své vylepšení výkonnosti, bezpečnosti a poskytnutí dodatečných nástrojů a funkcí pro vytváření pokročilých 3D grafických aplikací novější verze WebGL 2.0 používanější.

### 3.2.3 WebXR

Webový standard, který vychází a který nahradil standard WebVR, podporuje zážitky ve virtuální i rozšířené realitě v rámci prostředí prohlížeče. Poskytuje jednotné rozhraní API pro přístup k zařízením AR a VR a umožňuje vytvářet zážitky ve smíšené realitě.

### 3.2.4 WebGPU

\*\*\*\*

### 3.2.5 High level frameworky

BabylonJS, stejně jako další 3D frameworky jako ThreeJs nebo A-Frame, poskytuje vysoko-úrovňovou abstrakci nad rozhraním WebGL API, což vývojářům usnadňuje vytváření a správu 3D scén v prohlížeči.

## 4 BabylonJS

BabylonJS je JavaScriptový framework, který se zařadil mezi oblíbené nástroje pro tvorbu 3D webové grafiky. BabylonJS je postaven na rozhraních API HTML5 Canvas a WebGL, která poskytují základ pro vytváření 3D webové grafiky.

Babylon.js je projekt s otevřeným zdrojovým kódem, který je hostován na serveru GitHub a je napsán převážně v jazyce TypeScript, což je nadstavba jazyka JavaScript, která přidává volitelné statické typování a další funkce jazyka. To umožňuje lepší organizaci kódu, zvýšení produktivity vývojářů a zlepšení kvality kódu.

Jednou z klíčových předností vykreslovacího jádra Babylon.js je jeho podpora široké škály platforem a zařízení. Je navržen tak, aby byl kompatibilní s různými webovými prohlížeči, včetně desktopových a mobilních prohlížečů. Podporuje také řadu vstupních zařízení, včetně klávesnic, myši, dotykových obrazovek a herních ovladačů.

### 4.1 Historie

BabylonJS byl původně vytvořen Davidem Catuhem v roce 2012, softwarovým inženýrem společnosti Microsoft. V té době Catuhe pracoval na projektu, který vyžadoval použití WebGL, nízkoúrovňového 3D grafického API, které umožňuje hardwarově akcelerované vykreslování 3D grafiky ve webových prohlížečích. Catuhe však zjistil, že práce přímo s WebGL je složitá a časově náročná a vyžaduje značné znalosti nízkoúrovňového programování.

Aby tento problém vyřešil, vytvořil Catuhe BabylonJS jako vysoko-úrovňový framework, který by zjednodušil proces vytváření 3D webové grafiky pomocí WebGL. Jako jazyk pro tento framework zvolil JavaScript, protože byl již široce používán při vývoji webových aplikací a měl velkou komunitu vývojářů.

BabylonJS byl poprvé uvolněn jako open-source projekt na GitHubu v roce 2013 a díky snadnému použití, výkonné sadě nástrojů a obsáhlé dokumentaci

si rychle získal oblibu mezi vývojáři..

V průběhu let se BabylonJS dále vyvíjel a zdokonaloval a pravidelně přibývaly nové funkce a vylepšení. V roce 2016 byl BabylonJS oceněn jako nejlepší open-source herní engine v soutěži Develop Awards, což ještě více upevnilo jeho pozici předního frameworku pro 3D webovou grafiku.

## 4.2 Alternativy

### 4.2.1 ThreeJS

Dalším oblíbeným frameworkem je Three.js. Jde o flexibilní framework, který lze použít pro širokou škálu projektů 3D webové grafiky, od jednoduchých animací až po komplexní interaktivní zážitky. Nabízí řadu funkcí pro práci s 3D modely, včetně podpory fyziky, animací a osvětlení. Three.js má velkou a aktivní komunitu vývojářů, kteří vytvořili širokou škálu zdrojů, včetně výukových programů, příkladů a zásuvných modulů. Stejně jako v případě Babylonu, jej lze použít k vytváření 3D webové grafiky pro širokou škálu platforem, včetně stolních počítačů a mobilních zařízení, ale také pro virtuální a rozšířenou realitu. Díky

### 4.2.2 A-Frame

Dalším oblíbeným rámcem pro tvorbu 3D webové grafiky je A-Frame. A-Frame používá deklarativní syntaxi HTML, která usnadňuje vytváření 3D scén bez nutnosti psát velké množství kódu. Je postaven nad frameworkem Three.js, díky čemuž zdědil mnoho jeho vlastností a silných stránek. A-Frame je speciálně navržen tak, aby usnadnil vytváření zážitků virtuální a rozšířené reality pro web, a má řadu funkcí pro práci s VR a AR.

### 4.2.3 Unity WebGL

Unity je populární platforma pro vývoj her, která podporuje export projektů na web pomocí WebGL. Unity nabízí řadu funkcí pro tvorbu 3D her a simulací,

včetně fyzikálních enginů, animačních nástrojů a podpory skriptování v jazyce C#. Unity je dobrou volbou pro vývojáře, kteří již znají platformu Unity a chtějí

## 5 Byblyon.js Rendering engine

Babylon.js renderovací engine je postaven na WebGL, nízkoúrovňové grafické API, které mu umožňuje přímý přístup ke grafickému procesoru v počítači uživatele. To umožňuje vysoce výkonné 3D vykreslování ve webovém prohlížeči bez nutnosti používat proprietárních zásuvných modulů nebo softwaru. Skládá se z několika komponent, které společně zajišťují vykreslování 3D scén ve webovém prohlížeči a jsou nezbytné pro funkčnost enginu.

### 5.1 Vykreslovací jádro

Vykreslovací jádro je základní komponentou frameworku, která je zodpovědná za vykreslování 3D scén ve webovém prohlížeči. Poskytuje výkonnou a flexibilní platformu pro vytváření vysoce výkonných interaktivních 3D aplikací, které běží v reálném čase.

### 5.2 Graf sceny

Babylon.js používá hierarchický graf scény k uspořádání 3D objektů a prostředků ve scéně. To umožňuje vývojářům vytvářet složité scény a snadno v nich manipulovat s objekty a animovat je.

### 5.3 Materiály a textury

Engine obsahuje výkonný materiálový systém, který vývojářům umožňuje vytvářet realistické povrchové materiály pro objekty ve scénách. Podporuje řadu materiálových vlastností, jako je difúzní barva, spekulární barva, lesk a kovo-

vost. Obsahuje také podporu mapování textur, normálové mapy a další pokročilé efekty textur.

## 5.4 Animační systém

Dále je součástí výkonný animační systém, který vývojářům umožňuje animovat objekty ve scéně pomocí řady technik, jako je animace klíčových snímků, skeletální animace a morfování cílů.

## 5.5 Fyzikální engine

Integrovaný fyzikální engine, který vývojářům umožňuje přidat do jejich 3D aplikací realistickou simulaci fyziky. Engine podporuje řadu fyzikálních vlastností, jako je gravitace, hmotnost, tření a restituce.

## 5.6 Input systém

Flexibilní vstupní systém, který zprostředkovává a zpracovává uživatelské vstupy, jako jsou vstupy z klávesnice, myši, dotyku a gamepadu.

## 5.7 Zvukový systém

Dovoluje implementovat do scén prostorové 3D zvukové efekty.

# 6 Architektura frameworku

Projekt je strukturován jako modulární knihovna, která se skládá z jednotlivých funkčních modulů. Tato modulární konstrukce systému Babylon.js umožňuje vývojářům vybrat si, které komponenty chtějí do svých projektů zahrnout. To pomáhá udržet projekt lehký a zaměřený na konkrétní potřeby, a přitom v případě potřeby poskytnout přístup k výkonným nástrojům.

## 6.1 Základní moduly

### 1. Babylon.js core

**Úloha:** Tento modul poskytuje jádro vykreslovacího enginu a grafu scény, jakož i základní podporu pro materiály, textury, animace a fyziku.

### 2. Babylon.js loader

**Úloha:** Poskytuje podporu pro nahrávání 3D zdrojů v různých formátech, jako jsou glTF, OBJ a FBX.

### 3. Babylon.js materials

**Úloha:** Poskytuje dodatečnou podporu pro pokročilé materiály, jako je PBR (Physically Based Rendering), podpovrchový rozptyl a procedurální materiály.

### 4. Babylon.js post-processes

**Úloha:** Poskytuje podporu pro postprocesingové efekty, jako je bloom, hloubka ostrosti a odlesky objektivu.

### 5. Babylon.js GUI

**Úloha:** Tento modul poskytuje podporu pro vytváření uživatelských rozhraní pomocí systému Babylon.js GUI.

### 6. Babylon.js procedural textures

**Úloha:** Poskytuje podporu pro procedurální generování textur pomocí algoritmů šumu, vzorů a fraktálů.

### 7. Babylon.js debugging

**Úloha:** Umožňuje debugování a profilování aplikací Babylon.js a zahrnuje funkce, jako je nástroj Inspector a nástroje pro logování.

## 6.2 Doplnkové moduly

### 1. Babylon.js audio

**Úloha:** Poskytuje podporu prostorového 3D zvuku.

### 2. Babylon.js physics

**Úloha:** Poskytuje další podporu pro pokročilou simulaci fyziky.

### 3. Babylon.js lens flares

**Úloha:** Zprostředkovává podporu pro vytváření lens flares a dalších efektů objektivů.

### 4. Babylon.js sprites

**Úloha:** Poskytuje podporu pro vytváření 2D animací a efektů spritů.

### 5. Babylon.js voxel

**Úloha:** Poskytuje podporu vykreslování voxelů a procedurálních voxelových světů.

### 6. Babylon.js terrain:

**Úloha:** Poskytuje podporu pro generování a vykreslování realistických 3D terénů.

### 7. Babylon.js animations

**Úloha:** Umožňuje další dodatečné animace, jako je prolínání, retargeting a stavové stroje.

### 8. Babylon.js procedural textures library

**Úloha:** Poskytuje další procedurální textury, například mramor, dřevo a mraky.



Samotné jádro Babylon.js je páteří celého frameworku a poskytuje základní infrastrukturu, která umožňuje bezproblémovou spolupráci ostatních modulů. Ostatní základní moduly poskytují další doplňující funkce a jejich použití je čistě vázané na potřebu v rámci funkcionalit projektu.

### 6.3

Základní modul Babylon.js je pro funkčnost frameworku nezbytně nutný, a proto je jeho použití nutné a povinné. Poskytuje jádro vykreslovacího jádra, graf scény a základní podporu pro materiály, textury, animace a fyziku. Bez modulu Babylon.js core by framework nemohl správně fungovat.

Některé z volitelných modulů však nemusí být pro každý projekt potřeba, záleží na konkrétních požadavcích vyvíjené aplikace. Pokud například aplikace nevyžaduje prostorový 3D zvuk, nemusí být modul Babylon.js audio nutný. V takových případech lze tento modul z projektu vyloučit, ale jádro Babylon.js musí být obsaženo vždy.

## 7 Online nástroje

Babylon.js nabízí ve svém jádru výkonný vykreslovací engine, ale jeho skutečný potenciál spočívá také mimojiné v komplexní sadě online nástrojů, které poskytuje nad tímto rámcem. Ty se starají o různé aspekty vývoje 3D webových stránek a svými funkcemi zjednodušují proces vývoje a to vše v rámci jednoho uceleného a propojeného ekosystému. Nabídkou těchto nástrojů nad základním frameworkem se Babylon.js zařadil mezi přední volby pro vývojáře, kteří chtějí vytvářet špičkové 3D aplikace pro web.

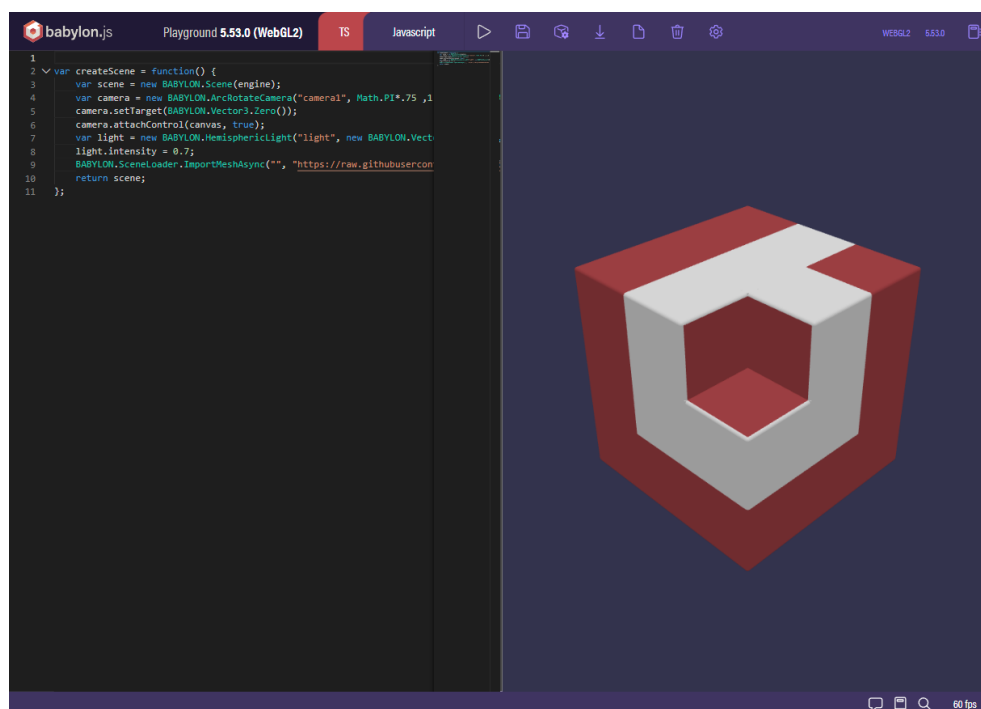
### 7.1 Playground

Babylon.js Playground dostupný na webové adrese <https://playground.babylonjs.com/>, je funkčně bohatý a výkonný online nástroj, který zjednodušuje proces práce s 3D scénami v rámci Babylon.js. Díky integrovanému editoru kódu, náhledu v reálném čase, integraci nástroje Inspector, možnostem sdílení a spolupráce a možnostem exportu a importu je Playground nepostradatelným nástrojem pro vývojáře, kteří se chtějí učit, vytvářet a zdokonalovat své 3D zážitky na webu. Umožňuje rychlé vytváření prototypů a experimentování s Babylon.js přímo z prostředí webového prohlížeče.

#### 7.1.1 Editor kódu

Playground obsahuje vestavěný editor kódu, který vývojářům umožňuje psát a upravovat kód v jazyce JavaScript nebo TypeScript přímo v nástroji. Editor kódu je vybaven funkcemi zvýrazňování syntaxe, automatickým doplňováním a kontrolou chyb, které usnadňují bezproblémové psaní kódu.

Editor musí vždy obsahovat definici funkce s názvem `createScene`, která vrací objekt scény. Na základě definice tohoto objektu je následně vykreslená požadovaná 3D scéna. Při prvním otevření nástroje již editor obsahuje výchozí kód funkce s vytvořením základní scény. Tato scéna kterou funkce navrácí se skládá z kamery, zdroje světla a jednoduché 3D geometrie koule a plochy. Předpřipravený kód slouží jako výchozí bod, se kterým mohou vývojáři experimentovat, upravovat a rozšiřovat jej podle svých potřeb.



Obrázek 1: Prostředí nástroje Babylon.js Playground

### 7.1.2 Náhled 3D scény

Playground nabízí náhled 3D scény v reálném čase, takže vývojáři mohou okamžitě vidět výsledky změn svého kódu. Tato funkce eliminuje nutnost přepínat mezi samostatnými okny kódu a náhledu, čímž zefektivňuje proces vývoje.

Pro zobrazení výsledků změn provedených v kódu je potřeba kliknout na tlačítko "Run" v liště nástrojů nebo stisknout klávesy `Alt + Enter`. Tím se kód spustí a náhled 3D scény se aktualizuje v reálném čase.

### 7.1.3 Ukládání změn

Po provedení požadovaných změn mohou uživatelé svou práci uložit kliknutím na tlačítko "Save" nebo klávesovou zkratku Ctrl + S. Tím se vytvoří nová verze Playgroundu a vygeneruje se adresa URL (příklad: <https://playground.babylonjs.com/#420KZS>) s jedinečným identifikátorem projektu (v tomto případě: #420KZS). Tuto adresu URL pak mohou uživatelé sdílet s ostatními nebo ji přidat do záložek pro budoucí použití.

S každým dalším uložením nedochází k přepsání předchozího uloženého stavu aplikace, jelikož si playground udržuje historii verzí uložených projektů, což uživatelům umožňuje přístup k předchozím iteracím jejich práce. Každá tato verze je pak přístupná přidáním znaménka # a indexu verze za jedinečnou adresu projektu (příklad url adresy s verzí projektu: <https://playground.babylonjs.com/#420KZS#1>)

### 7.1.4 Export a import

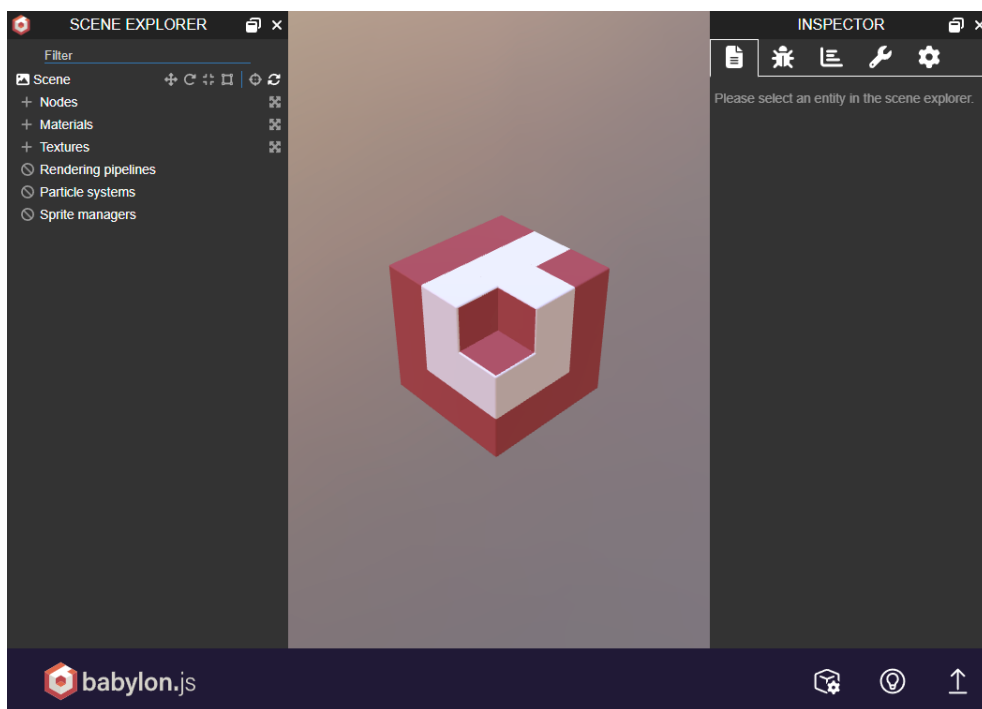
Vývojáři mohou do svých projektů importovat externí prostředky (například 3D modely, textury nebo skripty) nebo exportovat své výtvary v různých formátech, včetně .gltf, .glb a .babylon.

Použití externích prostředků sice může zlepšit proces vývoje, je však nutné zajistit, aby si vytvořený projekt zachoval dostupnost a funkčnost pro budoucí použití. Zjednodušení pomocí základních meshů a existujících textur a modelů může vést k rychlejšímu vývoji a porozumění problému a jejich řešení v prostředí playgroundu. Pokud je však použití externích zdrojů nezbytné, je zapotřebí zajistit přístup k souborům v rámci internetu zkrze URL adresu, například jejich hostováním zkrze služby Githubu, Gitlabu, Dropboxu, Imguru a dalších. Při výběru je třeba mít na paměti, že všechny stránky hostující zdroje musí být kompatibilní s CORS (Cross-Origin Resource Sharing) a používat protokol HTTPS.

## 7.2 Sandbox

Babylon.js Sandbox je důležitý nástroj pro vývojáře pracující s externími modely v rámci frameworku. Sandbox je určen jako intuitivní a uživatelsky přívětivé online prostředí pro rychlé testování, iteraci a ladění 3D modelů a scén díky integraci nástroje inspektoru a všech jeho funkcionalit a to vše izolovaně a nezávisle na zbytku projektu.

Import 3D modelů do nástroje Sandbox je snadnou záležitostí a lze provést zkrze tlačítko pro import v pravém dolním rohu, či pomocí funkcionality drag and drop, která umožňuje jednoduše přetáhnout soubor 3D modelu (například .gltf, .glb nebo .babylon) do okna prohlížeče a ten se automaticky načte a vykreslí ve viewportu. Tím odpadá nutnost psát kód jen pro náhled modelu, což usnadňuje kontrolu a úpravu 3D zdrojů.



Obrázek 2: Importovaný model v prostředí Sandboxu

Jakmile je 3D model načten do Sandboxu, mohou s ním vývojáři pracovat a upravovat v reálném čase a jakmile jsou s úpravami spokojeni, mohou za pomoci funkce inspektoru opětovně exportovat projekt do různých formátů, včetně .gltf, .glb a .babylon.

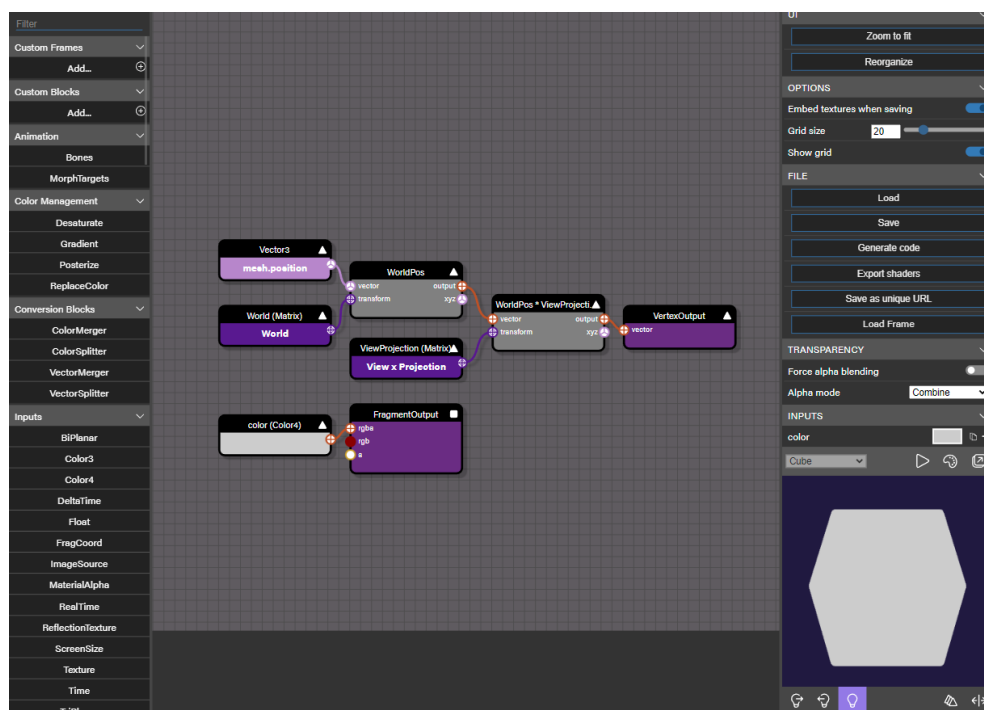
Nástroj Sandbox je dostupný na webové adrese <https://sandbox.babylonjs.com/>.

### 7.3 Node Material Editor

Node Material Editor je navržen pro zjednodušení tvorby a přizpůsobení shaderů pro 3D scény v projektech Babylon.js přizpůsobené specifickým potřebám a požadavkům projektu. Tento editor eliminuje nutnost ručního psaní kódu shaderů a umožňuje vizuálně vytvářet a upravovat složité materiály pomocí vizuálního programovacího rozhraní. Tento nástroj tak umožňuje vytvářet pokročilejší a jedinečné vizuální efekty, které mohou vylepšit celkový vzhled 3D scény.

Toto rozhraní se skládá z plátna, na které lze přidávat, propojovat a uspořádat logické bloky a definovat tak požadovaný materiál. Jednotlivé bloky představují různé operace, vstupy, výstupy a další prvky, které se podílejí na výsledném shaderu. Editor poskytuje rozsáhlou kolekci zabudovaných logických prvků, jako jsou základní matematické operace, vzorkování textur a vstupní uzly reprezentující různá data o scéně, například pozici ve virtuálním světě nebo UV souřadnice.

Při vytváření materiálu lze jednoduše přetahovat uzly z knihovny na plátno a propojovat je do struktury podobné grafu. Spojení mezi bloky určují tok dat a pořadí operací, což v konečném důsledku určuje vzhled materiálu. Editor také umožňuje snadnou změnu uspořádání, přidávání nebo odstraňování bloků, což umožňuje flexibilní proces návrhu. Mimo to poskytuje Node Material Editor náhled shaderu v reálném čase a zobrazuje vzhled materiálu na 3D modelu. Tato funkcionality umožňuje uživatelům rychle pozorovat výsledky svých změn a provést odpovídající úpravy.



Obrázek 3: Node Material Editor

Jakmile je materiál vytvořen a doladěn, mohou jej vývojáři exportovat jako soubor JSON nebo snippet Babylon.js, který lze následně importovat do projektu Babylon.js. Editor také podporuje import existujících materiálů pro další úpravy nebo kontrolu.

## 8 Implementace

Před samotnou implementací Babylon.js v rámci projektu, je třeba zvážit několik faktorů, které je třeba vzít v úvahu při rozhodování se mezi způsoby použitím Babylon.js. Existuje totiž více možností jak framework pro účel vizualizace 3D aktiv na stránkách použít a pro každý způsob pak existují různé způsoby implementace.

### 8.1 Babylon.js Viewer

Prohlížeč Babylon.js Viewer je nenáročná a snadno použitelná komponenta, která umožňuje vkládat a zobrazovat 3D scény a modely na webových stránkách bez nutnosti detailní znalosti frameworku Babylon.js nebo konceptů 3D programování. Pomocí Vieweru lze zjednodušit proces přidávání 3D obsahu do svých projektů a zároveň využívat výkonné vykreslovací schopnosti frameworku Babylon.js.

Viewer slouží jako souhrnný element pro engine Babylon.js a dodržuje stejný systém verzování jako engine samotný. Na základě toho je distribuována ve dvou verzích. První je stabilní verze využívající aktuální stabilní verzi engineu Babylon.js a následně preview verze, která využívá preview verzi engineu.

Viewer výrazně snižuje složitost integrace 3D obsahu do webových stránek, což z něj činí ideální řešení pro vývojáře, kteří vyžadují jednoduchý a rychlý způsob přidávání 3D scén nebo modelů. Viewer také automaticky zajišťuje změnu velikosti a responzivitu, čímž zajišťuje, že 3D obsah vypadá a funguje dobře na různých zařízeních a při různých velikostech obrazovky.

Babylon.js Viewer je jednoduchý za cenu omezené kontroly nad 3D scénou. Ačkoliv viewer do jisté míry úpravy zobrazení umožňuje pro pokročilejší úpravy a přesnější ovládání chování scény může být zapotřebí použití přímo frameworku Babylon.js. Přestože je Viewer optimalizován z hlediska výkonnosti, nemůže nabídnout takové možnosti optimalizace pro konkrétní případy použití nebo konfigurace hardwaru jako samotný vývoj v rámci samotného



frameworku.

### 8.1.1 Přidání Vieweru do projektu

Pro přidání Vieweru Babylon.js do projektu je nejdříve zapotřebí přidání script tagu v hlavičce HTML souboru, v rámci něhož se získá reference na hostovanou verzi knihovny Vieweru sdílenou na CDN serverech.

```
1 <script src="https://cdn.babylonjs.com/viewer/babylon.
  viewer.js"></script>
```

Příklad 1: Script tag s referencí stabilní verze Vieweru

Pak již stačí přidat do těla souboru HTML element `babylon` s požadovanými atributy, například atributem `model` s hodnotou adresy URL k souboru 3D modelu.

```
1 <babylon model="model.glTF"></babylon>
```

Příklad 2: Script tag s referencí stabilní verze Vieweru

## 8.2 Export HTML souboru z nástroje Playground

Další způsobem přidání 3D prvků do svého projektu je možnost vyžití nástroje Playgroundu pro export plně funkčního HTML souboru, obsahující canvas již s plně implementovaným enginem Babylon.js a veškerou logikou daného vyvíjeného projektu Playgroundu. Tento přístup umožňuje rychlou tvorbu prototypů 3D scén, což uživatelům umožňuje experimentovat s různými nastaveními a prohlížet si výsledky v reálném čase před jejich začleněním do svých projektů. Navíc díky exportu souboru HTML odpadá proces nastavení rámce Babylon.js v rámci projektu, což šetří čas a úsilí, zejména u menších projektů nebo projektů s omezeným 3D obsahem.

Další výhodou používání exportovaných souborů HTML je snadné sdílení s ostatními. Tato pohodlná metoda usnadňuje spolupráci na 3D scénách nebo modelech, aniž by příjemce musel konfigurovat rámec Babylon.js. Exportovaný

soubor HTML navíc obsahuje veškerý potřebný kód a prostředky Babylon.js, což umožňuje nezávislé spuštění 3D scény nebo modelu bez závislosti na externích závislostech.

Exportování souboru HTML z Babylon.js Playground má také některé nevýhody, například omezenou flexibilitu, problémy s údržbou a výkonnostní problémy. Při použití exportovaného souboru HTML nemusí být k dispozici stejná úroveň kontroly a flexibility jako při přímém použití frameworku Babylon.js, což může představovat problém pro pokročilé případy použití nebo v případech, kdy je zapotřebí přesná kontrola nad scénou.

Navíc s rozšiřováním projektu může být správa více exportovaných souborů HTML stále složitější a obtížnější, což ztěžuje sledování změn a údržbu zdrojového kódu. To může vést k potenciálním problémům při aktualizaci nebo optimalizaci 3D obsahu. Z hlediska výkonu nemusí být exportovaný soubor HTML tak optimalizovaný jako projekt vytvořený přímo pomocí rámce Babylon.js. U složitých scén nebo při cílení na specifické hardwarové konfigurace se může ukázat jako vhodnější volba přímé použití frameworku.

### 8.2.1 Export z nástroje Playground

Nejprve je třeba vytvořit požadovanou 3D scénu nebo model v online nástroji Babylon.js Playground a provést všechny potřebné úpravy, aby bylo dosaženo požadovaného výsledku. Jakmile je scéna vyhovující, stačí kliknout na tlačítko "Download" na panelu nástrojů či použít klávesovou zkratku CTRL+SHIFT+S a uložit do lokálního uložení komprimovaný balíček (.zip) souborů, obsahující soubory textur, modelů a jiných spolu se zcela funkčním index.html souborem.



Obrázek 4: Zvýrazněné tlačítko z lišty nástrojů pro stažení souboru obsahující HTML soubor se scénou.

### 8.2.2 Přidání do projektu

Integrace exportovaného souboru HTML z Babylon.js Playground do projektu může být provedena několika způsoby v závislosti na struktuře a požadavcích projektu. Každý přístup má své výhody a nevýhody, které je třeba zvážit, aby byla zajištěna bezproblémová integrace se stávajícím projektem.

Jednou z možných metod je přímé vložení obsahu exportovaného souboru HTML do stávající struktury HTML. Zkopírováním kódu Babylon.js, včetně elementu canvas a všech potřebných skriptů, a jejich vložení do souboru HTML projektu lze 3D scénu nebo model umístit na požadované místo. Tato metoda umožňuje snadnou integraci 3D obsahu do stávajícího rozvržení a stylování, což z ní činí vhodnou volbu pro mnoho projektů.

Alternativně lze k vložení exportovaného souboru HTML do projektu použít iframe. Tento přístup zachovává oddělení mezi kódem Babylon.js a hlavním souborem HTML, čímž efektivně odděluje 3D obsah. Za tímto účelem se v souboru HTML projektu vytvoří prvek iframe a atribut "src" se nastaví na cestu k exportovanému souboru HTML. Tato metoda může být obzvláště užitečná pro správu a údržbu 3D obsahu odděleně od zbytku projektu.

Další možnost zahrnuje vytvoření odkazu nebo tlačítka v projektu, které otevře exportovaný soubor HTML v novém okně nebo kartě prohlížeče. Tato metoda udržuje 3D obsah zcela oddělený od hlavního projektu a poskytuje uživatelům vyhrazené okno pro interakci s 3D scénou nebo modelem. Pro implementaci této možnosti se do souboru HTML projektu přidá prvek hypertextového odkazu nebo tlačítka a atribut "href" (u hypertextových odkazů) nebo událost onclick (u tlačítek) se nastaví tak, aby se otevřela adresa URL exportovaného souboru HTML.

## 8.3 Přímá implementace frameworku

Třetím přístupem k integraci 3D obsahu na webové stránky pomocí Babylon.js je přímé použití rámce Babylon.js. Tato metoda poskytuje největší flexibilitu

a kontrolu nad 3D obsahem a je vhodná pro projekty, které vyžadují pokročilé funkce, přizpůsobení a optimalizaci výkonu. Přímé použití frameworku zahrnuje zahrnutí knihovny Babylon.js do projektu a následné napsání vlastního kódu pro vytváření, manipulaci a vykreslování 3D scén.

### 8.3.1 CDN

Oblíbenou a pohodlnou metodou začlenění rámce Babylon.js do projektu je použití sítě pro doručování obsahu (CDN). Síť CDN hostují knihovnu Babylon.js a doručují ji uživatelům s nízkou latencí, čímž poskytují rychlý a efektivní způsob přístupu ke knihovně. Pomocí sítě CDN mohou vývojáři snadno začlenit rámec Babylon.js do svého projektu, aniž by museli sami stahovat nebo spravovat soubory knihovny.

Využití sítě pro doručování obsahu nabízí při začlenění rámce Babylon.js do projektu několik výhod. Jednou z těchto výhod je jednoduchost. Začlenění knihovny vyžaduje pouze script tag v souboru HTML, což je snadná a nenáročná metoda pro vývojáře, kteří chtějí s rámcem rychle začít. Další výhodou jsou automatické aktualizace. Verze rámce hostovaného v síti CDN je obvykle automaticky aktualizována při vydání nové verze. Tím je zajištěno, že projekt vždy používá nejnovější a nejstabilnější verzi knihovny, aniž by vývojář musel provádět ruční aktualizace. Kromě toho síť CDN ukládají soubory knihovny do mezipaměti na více serverech po celém světě, což může zlepšit rychlost načítání 3D obsahu, zejména pro uživatele, kteří se nacházejí daleko od serveru hostující projekt. A konečně, přenesením hostování knihovny Babylon.js na síť CDN se sníží zatížení vašeho vlastního serveru, což může pomoci zlepšit celkový výkon projektu.

Na druhou stranu má používání toto použití i určitá omezení. Jedním z omezení je, že není možné mít plnou kontrolu nad verzí knihovny, což může být problém, pokud projekt závisí na konkrétní verzi nebo pokud potřebujete použít vlastní úpravy knihovny. Další potenciální nevýhodou je riziko výpadku nebo přerušení provozu, což by mohlo ovlivnit dostupnost knihovny Babylon.js.

Ačkoli jsou sítě CDN obecně spolehlivé, vždy existuje možnost, že k těmto problémům dojde.

Každý modul `v` je poskytován ve formě vlastní adresy URL hostované v CDN, což vývojářům umožňuje zahrnout pouze specifické funkce potřebné pro jejich projekty, čímž se snižuje celková velikost a složitost aplikace. Vždy je zapotřebí zahrnout minimálně `babylon.js` core modul. Toho lze docílit přidáním následující značky skriptu do hlavičky souboru HTML:

```
1  <script src="https://cdn.babylonjs.com/babylon.js"></script>
```

Příklad 3: Script tag užívající CDN pro začlenění core modulu frameworku

## 9 Inspektor

Babylon.js Inspector je komplexní a výkonný nástroj určený k zefektivnění procesu vývoje ve frameworku Babylon.js. Díky rozsáhlým funkcím pro průzkum scény, úpravu vlastností, profilování výkonu, nastavení vykreslování v reálném čase, nástrojům pro debugování a úpravy prostředí umožňuje inspektor vývojářům snadno vytvářet, optimalizovat a zdokonalovat své 3D projekty. Tento nástroj je přístupný prostřednictvím online nástrojů jako jsou Sandbox a Playground, které jej integrují.

rozpracované překlady: Inspektor Babylon.js je přístupný především v rámci Playgroundu a Sandboxu, jak jste zmínil. Jedná se o hlavní online nástroje, které integrují Inspektor pro snadný přístup a zefektivnění vývoje.

Inspektor však můžete používat i ve vlastních projektech Babylon.js mimo tyto online nástroje. Za tímto účelem musíte do svého projektu zahrnout Inspektor přidáním následující značky skriptu do souboru HTML:

lze jej také použít jako samostatný nástroj ve vlastních projektech Babylon.js, čímž poskytuje všestranný a cenný zdroj informací pro vývojáře pracující s tímto frameworkem.

Ano, je to tak. Babylon.js Inspector je samostatný modul v rámci frameworku Babylon.js. Je navržen tak, aby fungoval dohromady se základní knihovnou Babylon.js a poskytoval další funkce pro ladění, úpravy a optimalizaci 3D scén.

Jak již bylo zmíněno, modul Inspektor můžete do svého projektu zahrnout přidáním značky skriptu pro soubor `babylon.inspector.bundle.js`. Tím bude modul Inspektor dostupný v rámci vaší aplikace Babylon.js, což vám umožní používat jeho funkce vedle jádra frameworku.

Modulární konstrukce systému Babylon.js umožňuje vývojářům vybrat si, které komponenty chtějí do svých projektů zahrnout. To pomáhá udržet projekt lehký a zaměřený na konkrétní potřeby, a přitom v případě potřeby poskytnout přístup k výkonným nástrojům, jako je Inspector.

Lze jej používat jako samostatný nástroj, ale je také integrován do Sandboxu a rozšiřuje jeho možnosti o další funkce pro ladění, analýzu výkonu a úpravy vlastností.

V kontextu diskuse o online nástrojích Babylon.js je důležité poznamenat, že Inspektor nabízí hodnotu jako samostatný nástroj, zejména při použití v rámci vlastní aplikace Babylon.js. Při diskusi o funkcích Sandboxu je však také nezbytné zdůraznit integraci nástroje Inspektor, protože významně rozšiřuje možnosti Sandboxu a zefektivňuje proces vývoje.

Souhrnně lze říci, že inspektor Babylon.js lze považovat jak za samostatný online nástroj, tak za funkci Sandboxu, v závislosti na kontextu, v němž je používán.

```
1   var createScene = function () {  
2       var scene = new BABYLON.Scene(engine);  
3  
4       var camera = new BABYLON.FreeCamera("camera1", new  
           BABYLON.Vector3(0, 5, -10), scene);  
5       camera.setTarget(BABYLON.Vector3.Zero());  
6       camera.attachControl(canvas, true);  
7  
8       var light = new BABYLON.HemisphericLight("light", new  
           BABYLON.Vector3(0, 1, 0), scene);  
9       light.intensity = 0.7;  
10  
11      var sphere = BABYLON.MeshBuilder.CreateSphere("sphere  
           ", {diameter: 2, segments: 32}, scene);  
12      sphere.position.y = 1;  
13  
14      var ground = BABYLON.MeshBuilder.CreateGround("ground  
           ", {width: 6, height: 6}, scene);  
15  
16      return scene;  
17  };
```

Příklad 4: Ukázka předpřipravené JavaScriptové funkce createScene

## 10 Nástin osnovy kapitol práce

### 1. Úvod

- Východiska práce
- Cíle práce
- Metody práce



2. 3D render a grafika na webu
3. Popis nižších vrstev (lower layer) technologií
4. Babylon.js
5. Uživatelské prostředí frameworku
6. Tvorba scény
7. Implementace scény do HTML kódu
8. Tvorba objektů
9. Manipulace s objekty

10. Pokročilé funkce

- Animace
- Uživatelské vstupy
- Fyzikální engine
- Kostry
- Partikly
- Shadery

11. Porovnání s konvenčními 3D editory

12. Import externí 3D grafiky

13. Praktická část

14. Závěr

15. Seznam použité literatury

## 11 Seznam přečtené literatury a zdrojů

### Seznam literatury

- [1] *Home / Babylon.js Documentation. Home / Babylon.js Documentation* [online]. Dostupné z WWW:<<<https://doc.babylonjs.com/>>>
- [2] *WebGL: 2D and 3D graphics for the web - Web APIs / MDN.* [online]. Copyright © 2005 [cit. 04.04.2021]. Dostupné Z WWW: <<<https://developer.mozilla.org/en-US/docs/Web/API/WebGLAPI>>>
- [3] *WebGPU: GitHub Pages* [online]. Copyright © 2021 the Contributors to the WebGPU Specification, [cit. 04.04.2021]. Dostupné z Www: <<<https://gpuweb.github.io/gpuweb/>>>