

## Projektowanie gier i mediów – lista zadań laboratoryjnych (Data kompilacji: 4 listopada 2024, t: 22:30)

### Spis treści

<b>1 Wprowadzenie</b>	<b>1</b>	<b>8 Laboratorium nr 7 – Dynamiczne ładowanie terenu</b>	<b>5</b>
<b>2 Laboratorium nr 1 – „Witaj Świecie!!!”</b>	<b>3</b>	8.1 Uwagi związane z zadaniem . . . . .	6
2.1 Różne uwagi związane z ćwiczeniem . . . . .	3	<b>9 Laboratorium nr 8 – System oświetlenia</b>	<b>6</b>
<b>3 Laboratorium nr 2 – „Tocząca się kulka”</b>	<b>3</b>	9.1 Uwagi związane z zadaniem . . . . .	6
3.1 Różne uwagi związane z ćwiczeniem . . . . .	3	<b>10 Laboratorium nr 9 – Rzeczywistość rozszerzona</b>	<b>6</b>
<b>4 Laboratorium nr 3 – Wstęp do trybu 2D – gra typu „Arkanoid”</b>	<b>3</b>	10.1 Uwagi związane z zadaniem . . . . .	7
4.1 Uwagi związane z ćwiczeniem . . . . .	4	<b>11 Laboratorium nr 10 – Rzeczywistość wirtualna</b>	<b>7</b>
<b>5 Laboratorium nr 4 – Odtwarzacz plików w formacie mp3, ogg i modułów w Unity</b>	<b>4</b>	11.1 Uwagi związane z zadaniem . . . . .	7
5.1 Uwagi związane z zadaniem . . . . .	4	<b>12 Laboratorium nr 11 – Tworzenie proceduralne obiektów w systemie węzłów aplikacji Blender</b>	<b>7</b>
<b>6 Laboratorium nr 5 – Fotogrametria i skanowanie obiektów w praktyce</b>	<b>4</b>	12.1 Uwagi związane z zadaniem . . . . .	8
6.1 Różne uwagi związane z ćwiczeniem . . . . .	5	<b>13 Laboratorium nr 12 – Tworzenie grafiki z wykorzystaniem sieci generatywnych i programu Krita</b>	<b>10</b>
<b>7 Laboratorium nr 6 – Napisy w stylu „Star Wars”</b>	<b>5</b>	13.1 Uwagi związane z zadaniem . . . . .	10
7.1 Uwagi związane z zadaniem . . . . .	5	<b>14 Laboratorium nr 13 – gra typu „Beach Head” – zadanie opcjonalne</b>	<b>10</b>
		14.1 Uwagi związane z zadaniem . . . . .	10
		<b>15 Dalsze informacje</b>	<b>11</b>

## 1 Wprowadzenie

Unity (<http://www.unity.com>), to środowisko pracy do rozwoju interaktywnych aplikacji z grafiką 3D oraz 2D. Istotną cechą środowiska Unity jest multi-platformowość. Oprócz dwóch wersji środowiska Unity dla systemów Windows oraz macOS (dostępna jest wersja dla systemu Linux), Unity pozwala na utworzenie postaci finalnej projektowanej aplikacji dla kilku różnych platform sprzętowo-programowych min. Windows, Linux, Android, IOS.

Należy dodać, iż istnieją inne darmowe systemy, lub przynajmniej oferujące możliwość darmowego tworzenia/rozwijania gier lub aplikacji 3D:

- (a) Shiva Engine, <https://shiva-engine.com/>, **nieaktywny**,
- (b) CryEngine, <http://www.crydev.net/>,
- (c) Project Anarchy, <http://www.projectanarchy.com/>, **nieaktywny**,
- (d) Torque3D, <https://torque3d.org/>,
- (e) Neoaxis, <http://www.neoaxis.com/>,
- (f) Evergine, <https://evergine.com/>,
- (g) Unigine, <https://unigine.com/>,
- (h) Unreal Engine, <https://www.unrealengine.com/>,
- (i) Stride Engine, <https://www.stride3d.net/>,
- (j) Open 3D Engine, <https://o3de.org/>,

- (k) C4 Engine, <http://c4engine.com/>,
- (l) Flax Engine, <https://flaxengine.com/>.

W przypadku Unity oprócz informacji oraz dokumentacji na stronach producenta:

- (a) podstawowe informacje o użytkowaniu środowiska Unity – <http://docs.unity3d.com/Documentation/Manual/index.html>,
- (b) opis komponentów, z których tworzone są aplikacje w Unity – <http://docs.unity3d.com/Documentation/Components/index.html>,
- (c) informacje o API i programowaniu w Unity – <http://docs.unity3d.com/Documentation/ScriptReference/index.html>.

Należy też wskazać nieco starsze strony WWW tj. Unity Community – jest to strona społeczności m.in. z dodatkowymi skryptami: <https://forum.unity.com/>. Przydatne informacje znajdują się także na stronie: <http://www.unity3dstudent.com>. Jednakże wymienione strony nie odnoszą się do najnowszych wersji środowiska Unity.

Dostępne są także serwisy, zbiory obiektów, tekstur zarówno o licencji komercyjnej, a także bezpłatnych:

- (a) zbiór siatek podstawowych: <https://www.thebasemesh.com/>,
- (b) tekstury na licencji CC0: <https://ambientcg.com/>,
- (c) zestaw tekstur i obiektów 3D: <https://polyhaven.com/>,
- (d) tekstury komercyjne, ze wsparciem dla Unity: <https://quixel.com/>,
- (e) zestaw tekstur, o darmowej licencji ze strony GPUOpen: <https://matlib.gpuopen.com/main/materials/all>,
- (f) obiekty komercyjne, <https://3drt.com/store/>, <https://kitbash3d.com>,
- (g) Mixamo, obiekty 3D wraz z bardzo bogatym zbiorem animacji: <https://www.mixamo.com>,
- (h) Pixel Art 2D: <https://ansimuz.com/site/>,
- (i) zasoby Kenny NL (zasoby darmowe oraz komercyjne): <https://kenney.nl/assets>,
- (j) zasoby ze sklepu itch.io: <https://itch.io/>,
- (k) zasoby dla programu Blender: <https://www.blenderkit.com/>.

Inne systemy programy wspomagające tworzenie gier:

- (a) Kythera AI, <https://kythera.ai>,
- (b) Krikey AI Animation Maker, <https://www.krikey.ai/>,
- (c) Cascadeur – system wspomagający animację, <https://cascadeur.com/>,
- (d) Daz3D – tworzenie postaci, <https://www.daz3d.com/>,
- (e) Poser3D – tworzenie postaci, <https://www.posersoftware.com/>,
- (f) Spriter Pro – tworzenie duszków 2D, <https://brashmonkey.com/spriter-pro/>,
- (g) Pro Motion NG – program do tworzenia grafiki typu PixelArt, <https://www.cosmigo.com/>,
- (h) PaintTool SAI – program do rysowania, <https://www.systemax.jp/en/sai/>,
- (i) Oprogramowanie firmy e-on Software: <https://www.bentley.com/software/e-on-software-free-downloads/>,
- (j) Terragen – tworzenie i rysowanie terenu, krajobrazu, <https://planetside.co.uk/>,
- (k) World Machine, <http://www.world-machine.com/>,
- (l) World Creator, <https://www.world-creator.com/en/index.phtml>,
- (m) SideFX Houdini, <https://www.sidefx.com/>.

## 2 Laboratorium nr 1 – „Witaj Świecie!!!”

Przygotować aplikację w Unity, która będzie odpowiednikiem aplikacji typu "Hello World!". Napis w wersji polskiej lub angielskiej, może zostać wykreowane jako element interfejsu użytkownika wykorzystujący obiekt Canvas oraz system 2D UI. Warto wykorzystać komponent/obiekt o nazwie Text – TextMeshPro.

### 2.1 Różne uwagi związane z ćwiczeniem

Przygotować wersję binarną dla systemów typu Desktop oraz systemu Android.

## 3 Laboratorium nr 2 – „Tocząca się kulka”

Na podstawie materiału „Roll a ball”:

- <https://learn.unity.com/project/roll-a-ball>),

opracować podobny prototyp gry, składający się z trzech poziomów. Po ukończeniu przez gracza poziomu i wyświetleniu się stosownego komunikatu ma zostać uruchomiony kolejny poziom gry.

### 3.1 Różne uwagi związane z ćwiczeniem

Nie jest to konieczne, ale można dodać kolejny poziom oraz wykonać wersję na urządzenia mobilne wykorzystując dostęp do akcelerometru. W tym przypadku sterowanie bilą będzie wykorzystywać dostępny akcelerometr.

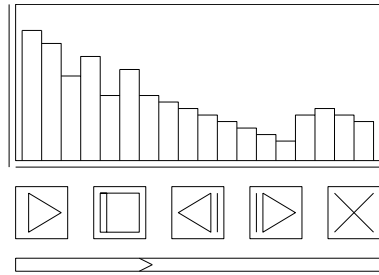
## 4 Laboratorium nr 3 – Wstęp do trybu 2D – gra typu „Arkanoid”

Opracować prototyp gry typu Arkanoid, za wzór można wykorzystać oryginalną grę wydana w roku 1988, ilustracje z gry zaprezentowana na Rys. 1.



Rysunek 1: Gra Arkanoid z 1988 roku wydana na komputer ośmiobitowy ZX Spectrum, przez Taito

Podstawową mechanikę gry jest naturalnie zbijanie klocków, i ten mechanizm rozgrywki musi być zaimplementowany. W oryginalnej grze Arkanoid pojawiają się także dodatkowe elementy jak obiekty pojawiające się po zbitiu klocka np. dodatkowe punkty, dodatkowe tzw. życie a także możliwość strzelenia do klocków, czy też dodatkowe bile jakie pojawiają się po zebraniu dodatkowego obiektu przez paletkę gracza.



Rysunek 2: Schemat interfejsu GUI dla odtwarzacza plików audio

#### 4.1 Uwagi związane z ćwiczeniem

Grę typu Arkanoid należy zaimplementować w tzw. trybie 2D, wykorzystując dedykowane obiekty do grafiki 2D, jakie są oferowane przez środowisko Unity.

## 5 Laboratorium nr 4 – Odtwarzacz plików w formacie mp3, ogg i modułów w Unity

Wykorzystując tzw. nowe GUI zbudować odtwarzacz plików audio odtwarzający dostępne formaty plików m.in. pliki w formacie mp3, ogg oraz tzw. moduły. Odtwarzacz w wersji podstawowej powinien odtwarzać pliki zawarte w projekcie. Oznacza to, iż będzie możliwość odważania tylko danych, jakie zostaną wprowadzone na poziomie tworzonego projektu.

Dodatkowym zadaniem jest wprowadzenie możliwości wskazywania dowolnych plików za pomocą systemowego okna dialogowego dostępnego w ramach użytej platformy sprzętowo-programowej. Ogólnie odtwarzacz musi posiadać następujące elementy interfejsu:

- analizator widma dźwięku,
- przycisku sterujące: odtwarzanie pliku/dźwięku, pauza, poprzedni plik, następny plik, zatrzymanie odtwarzania,
- pasek postępu.

Rys. (2) przedstawia ogólny schemat interfejsu dla odtwarzacza plików. Można uzupełnić interfejs o listę plików do wyboru. Naturalnie, układ interfejsu może być dowolny.

#### 5.1 Uwagi związane z zadaniem

Procedury odtwarzające pliki audio powinny być oparte o rozwiązania własne Unity. Uzupełnieniem projektu może być też przygotowanie kilku wersji dla różnych platform sprzętowych. Podstawowa wersja aplikacji powinna być dedykowana dla systemów desktop np. Windows, Mac OS czy Linux, ale warto rozważyć przygotowanie wersji dla systemu Android. Dodatkowo można do projektu dodać możliwość odtwarzania plików video. Podstawowe informacje o odtwarzaniu plików cyfrowego video znajdują się pod adresem:

- <https://docs.unity3d.com/Manual/class-VideoPlayer.html>.

## 6 Laboratorium nr 5 – Fotogrametria i skanowanie obiektów w praktyce

Wykorzystując dowolne oprogramowanie np. VisualSFM, CMSV, MeshLAB, Meshroom, Reality Capture lub skaner 3D wykonać skan wybranego obiektu. Dodatkowa obróbka, aby uzyskać obiekt o niskiej rozdzielności może być ograniczona do podstawowej operacji decymacji siatki trójkątów. Można wykorzystać dostępne zestawy zdjęć podane na wykładzie, aby odtworzyć obiekty omawiane podczas wykładu. Można także wykorzystać narzędzie Meshroom<sup>1</sup> zastępujące konieczność stosowania dwóch oddzielnych programów tj. VisualSFM oraz CMSV:

<sup>1</sup>Fotogrametria stała się dość popularna również na urządzeniach mobilnych np. <https://get.display.land/>.

- <https://alicevision.github.io/#meshroom>.

Dostępny skaner 3D w laboratorium również pozwala na utworzenie obiektu 3D. Obiekt pozyskany poprzez procedurę skanowania (skanerem 3D lub fotogrametrią) powinien posiadać teksturę.

## 6.1 Różne uwagi związane z ćwiczeniem

Dodatkowe (nie jest obowiązkowe) zadanie, polega na porównaniu jakości otrzymanego obiektu przy zastosowaniu podanych programów oraz dedykowanego oprogramowania związanego ze skanerem 3D jakie jest obecnie dostępne do realizacji rekonstrukcji obiektu. Dostępny w pracowni skaner wykonuje również zdjęcia skanowanego obiektu. Pozyskane zdjęcia można wykorzystać do realizacji procedury fotogrametrii, aby sprawdzić, czy możliwe jest wykorzystanie ich do rekonstrukcji obiektu 3D.

Dalsze dodatkowe informacje dotyczące technik fotogrametrii można odszukać m.in. za pomocą poniższych linków:

- <https://github.com/mikeroyal/Photogrammetry-Guide>,
- <https://github.com/quilime/photogrammetry>,
- <https://github.com/awesome-photogrammetry/awesome-photogrammetry>,
- [https://github.com/openMVG/awesome\\_3DReconstruction\\_list](https://github.com/openMVG/awesome_3DReconstruction_list),
- [https://github.com/natowi/photogrammetry\\_datasets](https://github.com/natowi/photogrammetry_datasets),
- <https://github.com/natowi>,
- <https://openheritage3d.org>.

**Komentarz 6.1** *Oprogramowanie komercyjne tj. Agisoft Metashape (wersje Professional, Standard), 3DF Zephyr oraz Reality Capture oferują bezpłatne wersje demo bądź trial. Zatem też można je wykorzystać do realizacji zadań w oparciu o technikę fotogrametrii.*

## 7 Laboratorium nr 6 – Napisy w stylu „Star Wars”

Zadanie polega na utworzeniu animacji przesuwających się napisów otwierającego każdy film z sagi o gwiazdnych wojnach. Podobnie jak w filmie, na początku powinien pojawić się napis tytułowy, a następnie zgodnie z oryginalnymi filmami pojawiają się kolejne akapity tekstu wprowadzającego w fabułę filmu. Należy zaimplementować przynajmniej trzy akapity tekstu, wyrównanego do obu stron. W miarę możliwości zastosować czcionkę podobną do oryginalnej stosowanej w filmach. W tle powinny wzorem filmów znajdować się gwiazdy oraz kolor napisów także powinien być zgodny z oryginałem.

### 7.1 Uwagi związane z zadaniem

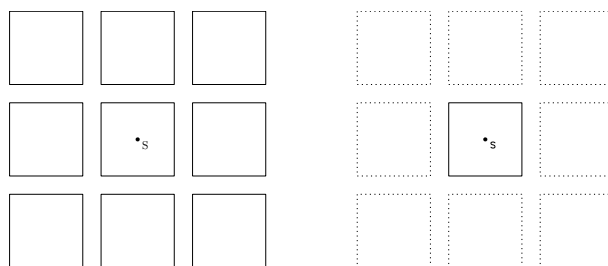
W realizowanym projekcie należy przede wszystkim zwrócić uwagę na płynność animacji przesuwających się napisów. Nie powinien występować efekt szarpania napisów np. ich krawędzi.

Nie jest to wymagane, ale można pokusić się o implementację ścieżki dźwiękowej, czy także początkowej animacji, tj. wylaniającej się planety, co można uprościć do przesuwania kamery w dół tak, aby zobaczyć powierzchnie planety. Sam kształt planety może być teksturą 2D, ale można także umieścić obiekt 3D w postaci sfery z odpowiednią teksturą<sup>2</sup>.

## 8 Laboratorium nr 7 – Dynamiczne ładowanie terenu

Dynamiczne ładowanie terenu (strumieniowanie danych terenu), pozwala na oszczędzanie zasobów pamięciowych w przypadku, gdy obszar jest znaczny lub bardzo dużych rozmiarów. Polega to na podzieleniu terenu na mniejsze bloki, które będą wczytywane w trakcie gry, gdy gracz zbliży się do danego obszaru. Oznacza to także konieczność usunięcia z pamięci obszaru leżącego daleko od aktualnej pozycji gracza.

<sup>2</sup>Przykładowa realizacja jaką można wykorzystać za wzór to: <https://www.youtube.com/watch?v=3UTSFFUhcMk&index=1&list=PL5KbKbJ6Gf99fy9sLMekM5TJQ0Lj-ujHN>.



Rysunek 3: Schemat zarządzania terenem, mamy dziewięć fragmentów terenu. Gracz i jego punkt startowy został oznaczony jako punkt  $S$ . Uwzględniając jego pozycję startową, tylko środkowy teren powinien być używany, pozostałe zostaną załadowane w momencie gry gracz zbliży się do nich na wyznaczoną odległość

Można zastosować dwa główne podejścia do dynamicznego ładowania terenu. Pierwsze polega na wykorzystaniu możliwości asynchronicznego ładowania dodatkowych scen. Pozwala to na posiadanie jednej sceny, która będzie odpowiadać za dołączanie pozostałych scen na których znajdować się będą poszczególne fragmenty terenu.

Drugim rozwiązaniem jest posiadanie tylko jednej sceny, i zarządzanie dynamicznym tworzeniem terenu. Nowo utworzone obiekty naturalnie będą reprezentować poszczególne obszary terenu. Nie są one naturalnie tworzone w sposób proceduralny ale instancjonowane ze wcześniej utworzonych obiektów typu prefab lub tworzone poprzez wskazywanie poszczególnych tekstur elewacji oraz terenu.

### 8.1 Uwagi związane z zadaniem

W zadaniu należy także zaimplementować punkt startowy, zapis pozycji gracza (również wczytanie), a także funkcjonalność teleportów pomiędzy poszczególnymi fragmentami terenu. Teleport powinien pozwalać wybrać miejsce do którego chcemy się przenieść, można to zrealizować np. za pomocą oddzielnego menu. Minimalna ilość fragmentów terenu to dziewięć. Ich rozmieszczenie powinno być symetryczne np. tak jak na Rys. 3.

## 9 Laboratorium nr 8 – System oświetlenia

Wykonać scenę przedstawiającą tzw. pudełko Cornella. Scena powinna powstać z obiektów z czterech głównych obiektów, pudełka, dwóch prostopadłościanów oraz prostopadłościanu reprezentującego źródło białego światła. Przykładowa wizualizacja pudełka Cornella została przedstawiona na Rys. 4.

Dodatkowe informacje o pudełku Cornella (m.in. wymiary poszczególnych ścian, informacje o kolorach) można odszukać na stronie Uniwersytetu Cornella:

- <http://www.graphics.cornell.edu/online/box/>.

Należy wykorzystać wbudowany system realizacji oświetlenia statycznego, aby uzyskać bardzo wysoką jakość obrazu.

### 9.1 Uwagi związane z zadaniem

Informacje uzupełniające znajdują się także na stronach Wikipedii:

- [https://en.wikipedia.org/wiki/Cornell\\_box](https://en.wikipedia.org/wiki/Cornell_box).

## 10 Laboratorium nr 9 – Rzeczywistość rozszerzona

Pakiet Unity oferuje także możliwość pracy z technologią rozszerzonej rzeczywistości (ang. augmented reality – AR). Wykorzystać można dwie biblioteki AR Foundation<sup>3</sup> oraz Vuforia. Zadaniem laboratorium jest realizacja podstawowej aplikacji AR, prezentującej wirtualny obiekt 3D za pomocą materialnego znacznika. W aplikacji,

<sup>3</sup>Starsza wersja tego pakietu pt. ARCore nie jest już wspierana bezpośrednio przez środowisko Unity, i nie jest zalecane jej stosowanie.



Rysunek 4: Cornell Box (pudełko Cornella) narysowane za pomocą programu POV-Ray. Autorem skryptu jest Kari Kivisalo

należy również wykorzystać pojęcie wirtualnych przycisków np. w celu zmiany, tekstury lub koloru prezentowanego obiektu 3D.

### 10.1 Uwagi związane z zadaniem

Aplikację najwygodniej jest zrealizować za pomocą biblioteki Vuforia. Pakiet Vuforia obsługuje większą gamę urządzeń przenośnych telefonów oraz dedykowanych urządzeń tj. okularów. Pakiet AR Foundation ma większe wymagania co do urządzenia z jakim współpracuje, choć jego zaletą jest tryb pracy, gdzie nie trzeba stosować dodatkowych znaczników.

## 11 Laboratorium nr 10 – Rzeczywistość wirtualna

Pakiet Unity od wersji 2017.2.x oferuje bezpośrednie wsparcie dla technologii wirtualnej rzeczywistości w ramach technologii HTC VIVE, Oculus, Microsoft Hololens, ale także Google VR obejmujące okulary Daydream oraz Cardboard.

Główny temat laboratorium to wykonanie dwóch projektów. Pierwszym jest realizacja mini-gry przedstawionej na wykładzie. Drugim typem aplikacji, jaki jest przewidziany do realizacji w ramach laboratorium, jest wirtualny spacer po zaprojektowanej budowlu z piętrem (aspekty wizualne nie są w tym przypadku najważniejsze). Jednakże w aplikacji ma zostać użyty dźwięk 3D, a także wykorzystany system wskazywanie obiektów np. spojrzenie na klamkę drzwi/okna, powinno spowodować iż otwarcie lub zamknięcie. Podobnie może zostać rozwiązany sposób np. włączenia światła w pomieszczeniu.

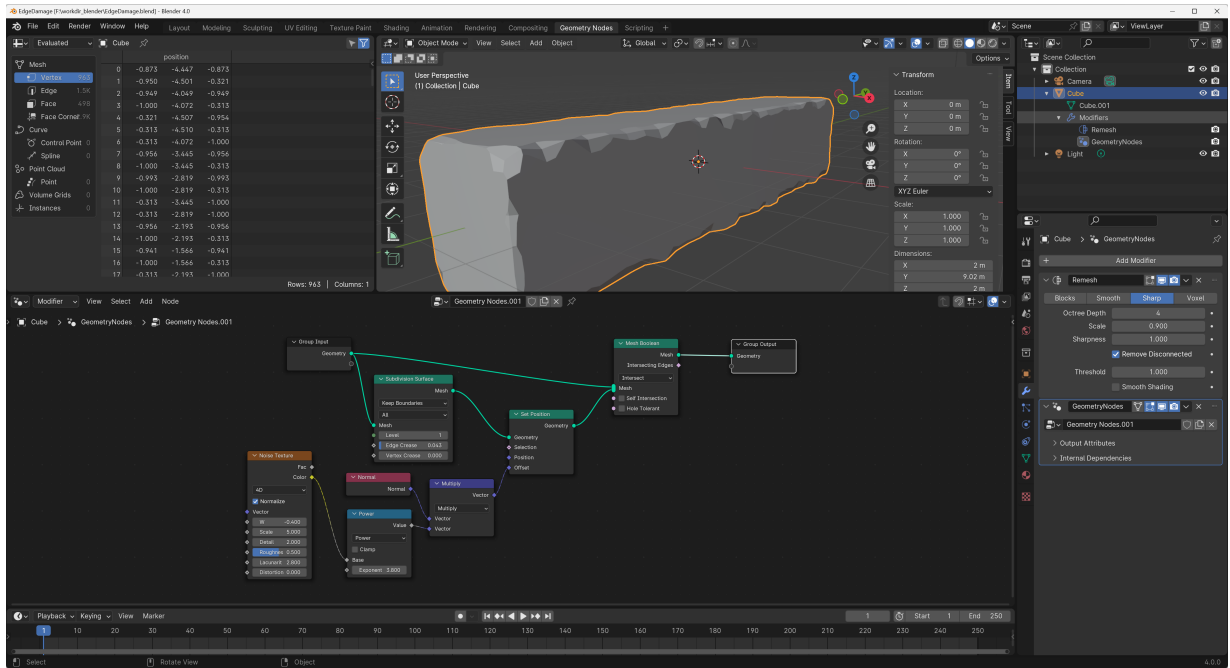
### 11.1 Uwagi związane z zadaniem

Najlepszym rozwiązaniem jest zastosowanie SDK dla hełmu HTC Vive lub OSVR (HTC Vive dostępne w laboratorium 308 A-2, OSVR w 405 A-2). Dopuszczalne jest też zastosowanie SDK dla rozwiązań opartych o telefon, np. Cardboard/Google VR lub Google Daydream View.

## 12 Laboratorium nr 11 – Tworzenie proceduralne obiektów w systemie węzłów aplikacji Blender

Zadanie polega na realizacji proceduralnie tworzonego obiektu za pomocą systemu węzłów w programie Blender. W ramach laboratorium należy wykonać:





Rysunek 5: Schemat systemu wprowadzający uszkodzenia na krawędziach obiektów. System węzłów został zbudowany w programie Blender 4.0. Pod adresem: <https://www.youtube.com/watch?v=1SrG6mYgk5Q>, można odszukać krótki film pokazujący jak realizuje się tego typu system

- (A) system oparty o węzły wprowadzający do modyfikowanego obiektu uszkodzenia krawędzi,
- (B) system generowania pojedynczych skał, pojedynczych kamieni wyłączenie w sposób proceduralny, wraz z materiałem również utworzonym za pomocą systemu węzłów,
- (C) opcjonalnie, można wykonać bardziej skomplikowany model np. kwiat rumianku lub proceduralne drzewo.

## 12.1 Uwagi związane z zadaniem

W przypadku zadania (A) całość systemu uszkadzającego krawędzie sprowadza się do wprowadzania zaburzeń do obiektu poprzez dodatkowy szum wprowadzone do zadanej geometrii. Wykonując operację części wspólnej do obiektów siatkowych można połączyć oryginalną geometrię wraz z jej zmodyfikowaną wersją otrzymując obiekt z pożądanymi uszkodzeniami. Ważnym elementem, iż należy zwiększyć gęstość geometrii na podstawowym obiekcie (modyfikator Remesh), aby system węzłów wprowadził odpowiednie uszkodzenia krawędzi.

Zadanie (B), wymaga definicji systemów węzłów dla generatora skał, Rysunek 6.

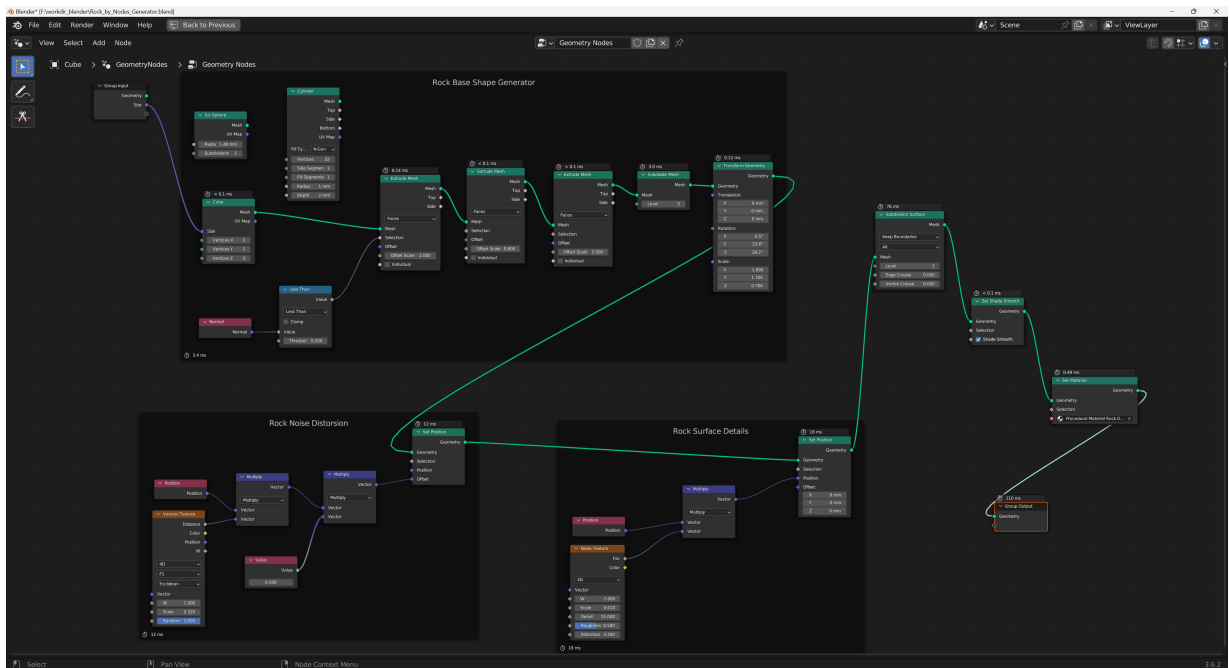
Istotne jest też określenie materiału, który jest renderowany za pomocą systemu Cycles, ważne jest też określenie skyboxa, ponieważ scena nie ma własnego źródła światła, zatem w opisie świata należy zdefiniować obiekt SkyBox wg. parametrów znajdujących się na Rysunku 7.

Zadanie generatora skał, można zrealizować na wiele sposobów, jedną z technik jest wykorzystanie operacji wyciągania obiektu i wprowadzania modyfikacji geometrii za pomocą generatora szumu oraz podziału diagramem Voronoi. Pomocne w realizacji tego zadania będą następujące filmy:

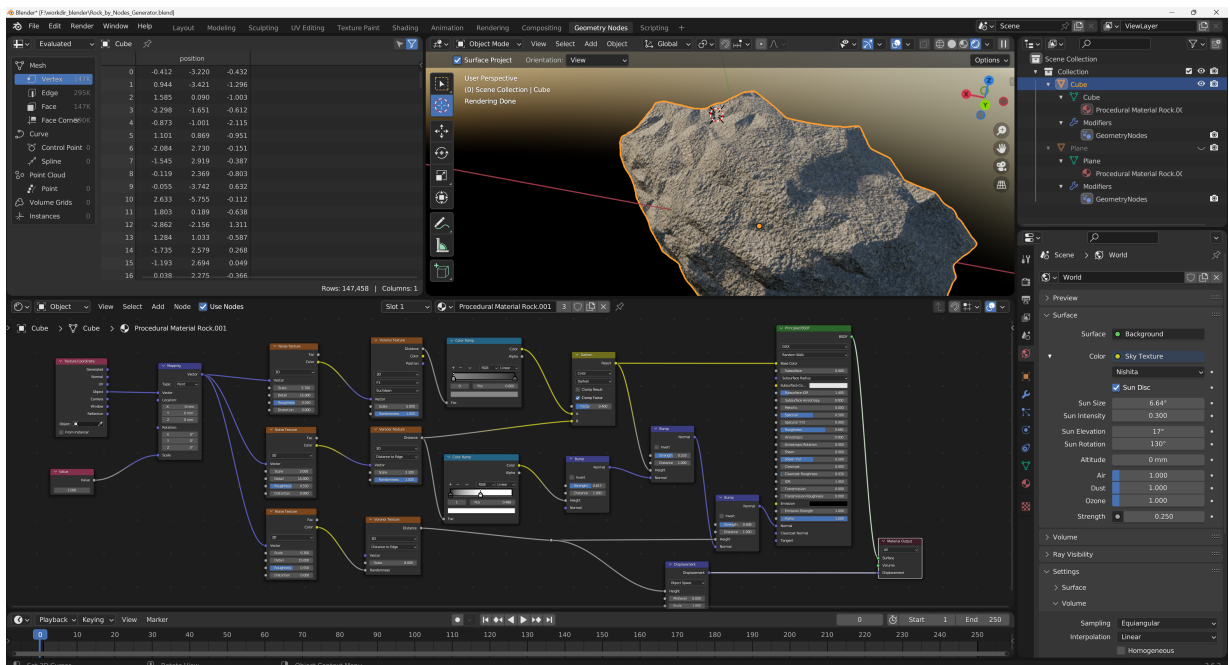
- <https://www.youtube.com/watch?v=y30UFZI4HTI>,
- <https://studio.blender.org/training/geometry-nodes-from-scratch/example-rock-generator/>.

Powstałe obiekty, zazwyczaj posiadają bardzo rozbudowaną geometrię, ich wykorzystanie np. w środowisku Unity, Unreal zazwyczaj wymaga dalszej obróbki, aby geometrię uprościć oraz wygenerować tekstury normalne, aby za pomocą materiału oddać szczegóły powierzchni.

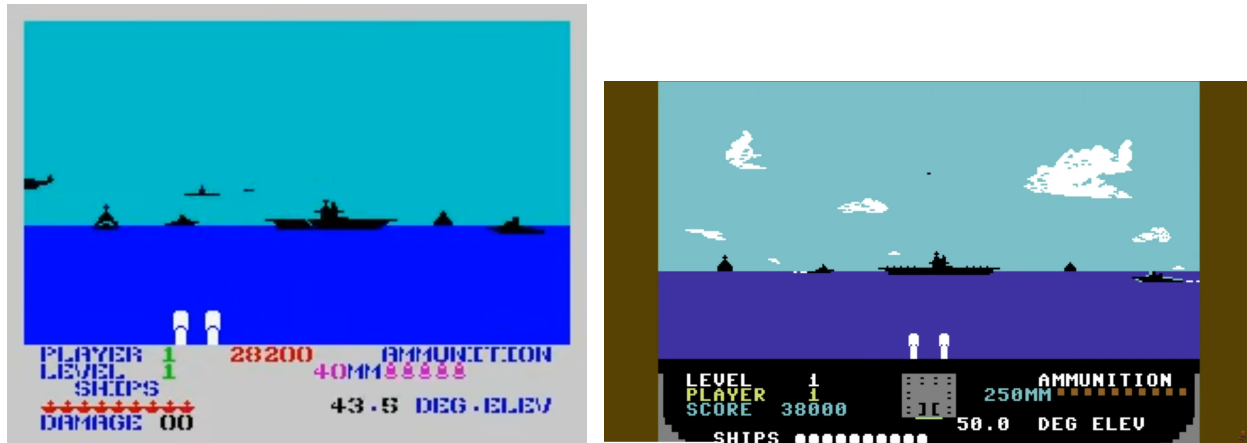




Rysunek 6: System węzłów dla generatora skał, realizowany w Blenderze 3.6.2



Rysunek 7: Definicja materiału oraz parametry skyboxa jaki jest stosowany do oświetlenia sceny, Scena nie zawiera obiektu kamery oraz oświetlenia.



Rysunek 8: Gra Beach Head, ekran lewy prezentuje obronę przed atakami lotniczymi w wersji na komputer ZX Spectrum 48kb, prawy ekran prezentuje pojedynek artyleryjski w wersji na komputer Commodore C64

## 13 Laboratorium nr 12 – Tworzenie grafiki z wykorzystaniem sieci generatywnych i programu Krita

Wykonać dwa rysunki za pomocą programu Krita przy zastosowaniu metod sztucznej inteligencji tworzącej grafikę 2D. Rysunek pierwszy należy wykonać określając tylko główny styl pracy np. poprzez wybór sieci np. Cinematic Photo (XL) oraz podając „prompt” opisujący co ma być treścią rysunku.

Rysunek drugi należy wykonać z wykorzystaniem modułu Live, gdzie systemem na podstawie podanego szkicu podzielonego na warstwy/regiony generuje poszczególne fragmenty ilustracji.

Program Krita domyślnie nie ma funkcji współpracujących z generatywną sztuczną inteligencją, należy zainstalować dodatkową wtyczkę: <https://github.com/Acly/krita-ai-diffusion>. Podstawowe informacje znajdują się na stronie: <https://www.interstice.cloud/>.

### 13.1 Uwagi związane z zadaniem

Do wykonania zadania przydane mogą być następujące materiały z serwisu YouTube:

- <https://www.youtube.com/watch?v=Ly6USRwThe0>,
- <https://www.youtube.com/watch?v=zDfosKZNYMQ>,
- <https://www.youtube.com/watch?v=AF2VyqSApJA>,
- <https://www.youtube.com/watch?v=PPx0E9YH57E>.

## 14 Laboratorium nr 13 – gra typu „Beach Head” – zadanie opcjonalne

Wykonać prototyp gry typu Beach Head z grafiką 3D. Wzorować można na pierwszych przedstawicielach gatunku tj. grach Beach Head z ery komputerów ośmiobitowych. Należy jednak zwrócić uwagę iż gra Beach Head, była grą wieloetapową zawierającą różne mini gry. W przypadku laboratorium prototyp powinien zawierać etap strzelania/obrony przed atakami lotniczymi i/lub morski pojedynek artyleryjski.

### 14.1 Uwagi związane z zadaniem

W roku 2000 na komputery PC została wydana odnowiona i uwspółcześiona wersja gry Beach Head. Realizacja prototypu naturalnie nie wymaga wysokiej jakości grafiki, można stosować obiekty typu low-poly. Warto rozważyć możliwość dodatnia opcji gry kooperacyjnej poprzez sieć Internet.

Informacje o grze oraz zapis rozgrywki dla wersji na komputer ZX Spectrum, można znaleźć np.:

- [https://en.wikipedia.org/wiki/Beach\\_Head\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Beach_Head_(video_game)),
- <https://www.youtube.com/watch?v=JI15QVrFRsg>.

## 15 Dalsze informacje

Przegląd pozycji książkowych dotyczących środowiska Unity w języku angielskim można odszukać na stronie <http://unity3dbooks.com/>. Poniższy spis literatury odnosi się do wszystkich ćwiczeń i zadań z przedmiotu „Projektowanie gier i mediów”.

### Literatura

- [1] Materiały dostępne na stronie głównej Unity, <http://www.unity3d.com>.
- [2] Will Goldstone, Projektowanie gier w środowisku Unity 3.x, Helion, 2012, <http://helion.pl/ksiazki/projektowanie-gier-w-srodowisku-unity-3-x-will-goldstone,prgun3.htm>
- [3] Alan R. Stagner, Unity Multiplayer Games, Pack Pub, 2013, <http://www.packtpub.com/unity-multiplayer-games/book>.