

**POLITECHNIKA BYDGOSKA**  
im. Jana i Jędrzeja Śniadeckich

**WYDZIAŁ TELEKOMUNIKACJI, INFORMATYKI I  
ELEKTROTECHNIKI**



**PRACA DYPLOMOWA INŻYNIERSKA**  
na kierunku Informatyka Stosowana

**Interaktywny konfigurator wnętrz pomieszczeń z  
dekoracyjnymi panelami 3D**

Pracę wykonał:  
Antoni Malak

Nr albumu:  
108650

Kierujący pracą:  
dr inż. Mariusz Sulima

Bydgoszcz,

Tutaj wstawić kartę pracy

## **Streszczenie**

Celem pracy jest zaprojektowanie oraz stworzenie interaktywnego konfiguratora elementów wnętrza, a dokładniej ściennych paneli dekoracyjnych. Najpierw opisane będą istniejące już rozwiązania w tym zakresie, a potem zostanie określony zbiór funkcjonalności oraz zakres interaktywności. Wymagane będzie zdefiniowanie i wymodelowanie elementów wnętrza, a następnie zaprojektowanie logiki programu umożliwiającej edycję parametrów tychże elementów.

**Słowa kluczowe:**

## **Summary**

**Keywords:**

# Contents

<b>1</b>	<b>Wstęp</b>	<b>5</b>
1.1	Cel pracy . . . . .	5
<b>2</b>	<b>Wstęp teoretyczny</b>	<b>5</b>
2.1	Wizualizacja . . . . .	5
2.2	Interaktywność . . . . .	6
2.3	Grafika komputerowa . . . . .	6
2.3.1	Silniki gier komputerowych . . . . .	7
2.3.2	Znane silniki gier komputerowych . . . . .	7
2.4	Konfigurator . . . . .	10
2.4.1	Typy konfiguratorów . . . . .	10
2.5	Architektura wnętrza . . . . .	10
2.5.1	Istniejące już rozwiązania . . . . .	11
2.5.2	Interaktywny widok trójwymiarowy . . . . .	11
2.5.3	Widok projektowania . . . . .	11
2.5.4	Katalog produktów i jego opcji . . . . .	11
2.5.5	Ekran podsumowania . . . . .	11
2.6	Panele ściennie . . . . .	11
2.6.1	Tapety . . . . .	12
2.6.2	Wygłuszające . . . . .	12
2.6.3	Dekoracyjne . . . . .	12
<b>3</b>	<b>Analiza i założenia projektu</b>	<b>13</b>
3.1	Wymagania sprzętowe . . . . .	13
3.2	Zakres funkcjonalności . . . . .	13
3.2.1	Katalog asortymentu . . . . .	13
3.2.2	Widok produktu w scenie . . . . .	13
3.2.3	Prezentacja oferty i możliwości konfiguracji . . . . .	13
3.2.4	Edycja pomieszczenia . . . . .	13
3.2.5	Poruszanie kamerą . . . . .	13
3.2.6	Zmiana oświetlenia . . . . .	13
3.3	Określenie stopnia interaktywności . . . . .	13
3.3.1	Zbiór elementów interaktywnych i stopień ich interaktywności . . . . .	13
3.4	Strona internetowa . . . . .	13
3.4.1	Znaczenie osiągalności informacji o produkcie . . . . .	13
3.4.2	Konfigurator jako katalog produktów . . . . .	13
<b>4</b>	<b>Realizacja projektu</b>	<b>13</b>
4.1	Modelowanie elementów 3d . . . . .	13
4.1.1	pomieszczenia . . . . .	13
4.2	Części projektu . . . . .	13
4.2.1	Logika i schemat działania . . . . .	13
4.2.2	Interfejs użytkownika . . . . .	13
4.2.3	Widok pomieszczenia . . . . .	13
4.2.4	Hierarchia i diagramy klas . . . . .	13
4.2.5	Osadzenie programu wynikowego na stronie internetowej . . . . .	13
<b>5</b>	<b>Wnioski</b>	<b>13</b>
<b>6</b>	<b>Literatura</b>	<b>14</b>
<b>7</b>	<b>Załączniki</b>	<b>15</b>

# 1 Wstęp

Szybki rozwój oraz powszechność internetu pozwolił na łatwiejsze dotarcie do potencjalnego klienta. Podając jedynie adres strony internetowej, zaprosić można kogoś na stronę internetową, której budowa i funkcje mogą być najróżniejsze. Może to być prosta statyczna strona wizytówka, której celem jest reklamowanie kogoś lub czegoś, lub skomplikowany profesjonalny program, który jest dostępny dla każdego, albo coś pomiędzy...

Dawniejszą formą reklamy były bannery reklamowe i ulotki. Bannery zwracały uwagę masy osób, a ulotki pełniły rolę informacyjną, podsuwając klientowi bardziej szczegółowe informacje, dotyczące produktu. Niestety pewne rzeczy lepiej jest przedstawić w sposób graficzny, dlatego powstały katalogi, w których widać produkt w jego 'naturalnym środowisku'. Dzięki technologii możliwe jest dostarczenie podobnej reklamy poprzez internet, nie marnując przy tym papieru, oraz dając sobie jednocześnie więcej możliwości w postaci interaktywności.

Dzięki zaimplementowaniu funkcji interaktywnych w konfiguratorze, możliwe będzie rozszerzenie zestawu funkcji ze statycznego obrazu do namiatki wizyty w salonie sprzedającym ten produkt. Ważne jest odpowiednie podejście do wizualizacji. Możliwe będzie obejrzenie pojedynczego panelu, ale bardziej pomocny dla klienta będzie widok przykładowego wnętrza, z ich użyciem. Możliwa będzie częściowa edycja panelu, oraz pomieszczenia w którym będzie on przedstawiany.

## 1.1 Cel pracy

Celem tej pracy dyplomowej jest stworzenie interaktywnej wizualizacji wnętrza, przedstawiającej panele ścienne potencjalnemu klientowi. Możliwy będzie wybór panelu z katalogu, zmiana jego koloru, oraz definiowanie sposobu ich ułożenia. Możliwa będzie również konfiguracja wnętrza, aby klient mógł obejrzeć produkt, przedstawiony we wnętrzu, które przypomina jego własny dom. Do wyboru będą z góry zdefiniowane pomieszczenia, pozwalające na częściowe dostosowywanie ich w kwestii kolorystyki wnętrza, ilości i rozmiaru okien oraz konfiguracji mebli. Ważnym aspektem będzie możliwość osadzenia programu na stronie internetowej, co zwiększy rzeszę potencjalnych klientów.

Przed napisaniem programu, wymagana będzie analiza podstawowych pojęć związanych z tematem, w celu lepszego zrozumienia tematu, oraz precyzyjniejszego nakreślenia zakresu funkcji, które powinien pełnić program. Trzeba będzie przeanalizować i porównać istniejące już rozwiązania, aby nowo powstały program nie powielał błędów swoich poprzedników i starał się inspirować ich mocnymi stronami. Warto byłoby rozłożyć ideę na czynniki pierwsze i zastanowić się nad różnymi ich aspektami, dzieląc te już istniejące na kategorie i porównując je ze sobą.

Dopiero potem możliwe będzie poprawne zdefiniowanie założeń i zakresu funkcji programu. Wyciągając wnioski z podobnych rozwiązań, będzie można zaprojektować rozwiązanie idealnie pasujące do postawionego problemu. Zdefiniowany zostanie zakres funkcjonalności, które powinna pełnić taka aplikacja. Ważne również będzie uprzednie zdefiniowanie stopnia interaktywności, gdyż implementacja pewnych funkcji nie może być możliwa w późniejszym stadium programowania. Warto również zastanowić się nad funkcjami jakie powinna pełnić internetowa aplikacja, która pełni również rolę reklamy.

Po fazach analizy i projektowania, możliwe będzie przejście do właściwej implementacji projektu, zaczynając od zaprogramowania logiki programu, poprzez modelowanie obiektów trójwymiarowych, aż do stworzenia wizualizacji, którą będzie można osadzić na stronie internetowej.

# 2 Wstęp teoretyczny

## 2.1 Wizualizacja

Wizualizacja jest obrazową reprezentacją przedmiotu, procesu lub stanu. Dawniej odnosiło się to do obrazu, albo procesu jego tworzenia, w naszej ludzkiej wyobraźni. Obraz ten istniał jedynie w naszej głowie, by przedstawić go komuś innemu wymagane było nakreślenie jego na jakimś medium, co wymagało dużych umiejętności i środków. Całą gałąź sztuki polega właśnie na doskonaleniu tego wymagającego procesu.

Wizualizacja komputerowa zajmuje się czymś podobnym. Stara się przedstawić coś nie fizycznego za pomocą ilustracji. Przedstawiany może być przedmiot, proces albo inny bardziej abstrakcyjny koncept. Dzięki oprogramowaniu graficznemu możliwe jest zautomatyzowanie tego procesu dla, z góry określonych, scenariuszy. Zadaniem wizualizacji komputerowej jest zebranie ogromu danych, odpowiednie ich przetworzenie, oraz przedstawienie ich w prosty i zrozumiały sposób użytkownikowi końcowemu.

Medium komputerowe pozwala na wiele możliwości w kwestii edycji prezentowanej grafiki. Parametry mogą zostać łatwo zmienione, a ich zmiana może zostać odzwierciedlona w warstwie graficznej. Mogą się one dotyczyć działania całego programu i jego algorytmów, albo części renderującej grafikę. Peryferia komputerowe pozwalają na łatwe wprowadzanie i dostrajanie parametrów, a program bierze te zmienione wartości pod uwagę tworząc nowy, zmieniony obraz. Funkcja pozwalająca na wprowadzanie zmian w symulacji, oraz natychmiastowa jej reakcja jest podstawą interaktywności.

## 2.2 Interaktywność

Interakcja jest wzajemnym oddziaływaniem na siebie osób, przedmiotów lub zjawisk.<sup>1</sup> Ważne żeby oddziaływanie te było dwu-kierunkowe, gdyż jednokierunkowym odpowiednikiem interakcji jest przyczynowość. Interaktywność jest miarą interakcji i jej jakości.

Interaktywność jest sytuacją w której każda kolejna akcja bierze poprzednią, lub kilka poprzednich, pod uwagę zmieniając jej kontekst. Wykonanie tej samej akcji w innym kontekście będzie skutkowało wykonaniem innej czynności. Ważne jest aby różne konteksty były łatwo rozpoznawalne i znane użytkownikowi.

Interaktywność może zostać zdefiniowana jako ciągła wymiana informacji, pomiędzy minimum dwoma uczestnikami. W tym przypadku jednym uczestnikiem jest wizualizacja, a drugim użytkownik. Użytkownik podaje informacje w postaci parametrów wejściowych, a wizualizacja zwraca użytkownikowi jak wyglądać może scena spełniające podane wymagania i parametry. Jest to proces który zachodzi cały czas w formie pętli, lecz nie każda akcja użytkownika musi równać się natychmiastowej reakcji systemu.

Ważne jest zachowanie przejrzystości po obu stronach. Należy dać użytkownikowi interfejs, który jest zrozumiały i łatwy w obsłudze. Dzięki zastosowaniu się do podstawowych zasad projektowania interfejsów powinien on być wizualnie przejrzysty i jednolity. Sensowne rozplanowanie różnych kontekstów, pozwoli na bardziej intuicyjną nawigację po programie i jego funkcjach.

Interaktywność pozwala użytkownikowi na kontrolowanie parametrów wejściowych symulacji, oraz obserwowanie nowo powstałego obrazu. Ważne jest aby użytkownik czuł, że ma kontrolę nad przebiegiem symulacji, a każda wprowadzona zmiana jest wprowadzana i aktualizowana tak szybko jak szybko się da. Dlatego czas obliczenia kolejnego widoku powinien być odpowiednio krótki. Zauważ to częściowo jak wiele obliczeń możemy przeznaczyć na ten cel, oraz jak wymagająca i realistyczna może być wizualizacja i jej grafika.

## 2.3 Grafika komputerowa

Grafika komputerowa jest dziedziną zajmującą się tworzeniem, przetwarzaniem oraz wyświetlaniem obrazów w medium komputerowym. Obraz może być wygenerowany przez komputer, tworząc obraz generowany komputerowo (tzw. CGI), albo może być odzwierciedleniem prawdziwego fizycznego obiektu. Do stworzenia zdjęcia obiektu używane są matryce światłoczułe, zmieniające docierające do niej fotony, na sygnał cyfrowy. Możliwe jest również stworzenie obrazu, będącego wizualizacją powstałą z pomiarów innych urządzeń i sensorów cyfrowych. Mogą to być zarówno wykresy przedstawiające zmianę wartości mierzonej przez sensor, albo coś bliższego obrazowi obiektu jak trójwymiarowy model powstały z zeskanowania bryły fizycznego obiektu.

Podstawowe biblioteki graficzne, takie jak SDL, OpenGL, pozwalają na rysowanie podstawowych kształtów, obrazów i stosowanie podstawowych przekształceń na tych obrazach. Może to być wystarczające do stworzenia prostych wizualizacji, ale aby osiągnąć oczekiwane wyniki wymagany byłby ogromny wkład pracy. Biblioteki te nie pozwalają na wyświetlanie skomplikowanych modeli trójwymiarowych, ani ich tekstur. Oświetlenie musiałoby być zaimplementowane od podstaw wymagając ogromnej wiedzy oraz dobrej optymalizacji. Na szczęście

---

<sup>1</sup>Definicja z internetowego słownika PWN

istnieje wiele istniejących już rozwiązań, które implementują wszystkie wymagane funkcje, są to silniki gier komputerowych.

### 2.3.1 Silniki gier komputerowych

Silnik gry komputerowej nie obejmuje jedynie warstwy wizualnej, jest to kompletny pakiet obejmujący każdy aspekt interaktywnego programu z naciskiem na generowanie grafiki w czasie rzeczywistym. Ważniejszymi częściami silnika, użytymi w projekcie będzie pobieranie inputu od użytkownika, modyfikacja modeli i sceny w czasie rzeczywistym, oraz wyświetlenie nowo wygenerowanego obrazu.

Sterowanie odbywa się poprzez urządzenia peryferyjne, takie jak klawiatura, myszka czasami ekran dotykowy, albo tablet graficzny. Możliwe jest również użycie niektórych wewnętrznych sensorów urządzenia takich jak żyroskop czy akcelerometr, które pozwalają na detekcję orientacji urządzenia mobilnego. Pozwalają one na dokładne i precyzyjne manipulowanie sceną. Jest to Kolejny mechanizm, który został już zaimplementowany jako część silnika, oraz nie wymaga on implementacji.

Jedną z funkcji silnika jest możliwość łatwego tworzenia interfejsu użytkownika, za pomocą którego użytkownik może sterować zachowaniem programu. Interfejsy mogą składać się z warstw, mogą pełnić rolę informacyjną (wskazywanie stanu zmiennych), albo interaktywną (wirtualne przyciski). Niektóre akcje programu mogą być wykonywane w sposób mniej jawny, ale bardziej intuicyjny (n.p. gesty dotykowe do obracania widoku). Inne natomiast będą wymagały kliknięcia na przycisk, albo inny element interfejsu.

Dzięki gotowemu silnikowi graficznemu, zamiast implementować silnik graficzny od zera, możliwe będzie zaimportowanie modeli wraz z teksturami. Modele mogą być otekstutowane, oraz mogą być zrobione z różnego rodzaju materiałów, które inaczej wyglądają i wchodzi w interakcję z oświetleniem. Do sceny można dodawać różne rodzaje obiektów, mogą to być proste modele trójwymiarowe, ale również różne rodzaje źródeł światła, albo efekty cząsteczkowe. Wymagane będzie również ręczne ustawienie sceny, oraz wszystkich jej elementów. Sceny mogą posiadać elementy interaktywne, co musi być uwzględnione w kompozycji jak i implementacji.

### 2.3.2 Znane silniki gier komputerowych

Rynek oferuje wiele gotowych rozwiązań, w kwestii silników do gier komputerowych. Mimo że one wszystkie mają ten sam cel, skupiają się one na różnych aspektach oraz posiadają pewne unikalne cechy. Jedne skupiają się na realistycznej grafice oraz streamowaniu ogromnych ilości danych, a inne skupiają się na prostocie użycia oraz multplatformowości. Każdy z nich swoimi możliwościami określa pewien zakres, w którym gra będzie się zawierać. Pełna gra stworzona na danym silniku nie może wyglądać lepiej niż demo technologiczne czy benchmark tego samego silnika, gdyż po prostu zabraknie zasobów na resztę logiki gry.

### Unreal engine

Silnik graficzny stworzony dla gry Unreal przez studio Epic Games. Gra Unreal została wydana w 1998 roku i była jedną z pierwszych trójwymiarowych gier FPS. Była to w pełni gra trójwymiarowa, która nie używała dwuwymiarowych sprite-ów zamiast przeciwników, jak wcześniej, a modeli trójwymiarowych. Były to czasy początków akceleratorów graficznych, dlatego niektóre obliczenia były wykonywane na procesorze, a inne za pomocą akceleratora graficznego.

Zestaw narzędzi był o wiele bardziej podstawowy niż te spotykane obecnie ze względu na ograniczenia graficzne oraz fakt, że gry komputerowe nadal były nowością, a technologia do tworzenia gier komputerowych była bardzo wczesna. Dlatego zostały w nim zaimplementowane funkcje, które można uznać za dość podstawowe, takie jak: detekcja kolizji, filtrowanie tekstur, wsparcie dla kolorowych źródeł światła, oraz edytor poziomów. Ciekawą nową funkcją było dodanie obiektów wchodzących w interakcję ze światłem, źródeł światła, przestrzennej mgły, czy implementacja sklepienia nieba.

Z czasem silnik się rozwijał, oddawane zostały kolejne narzędzia, a większa moc obliczeniowa komputerów pozwalała na stworzenie realistycznej grafiki. Czwarta wersja dodawała nowy sposób na generowanie oświetlenia, zamiast statycznego oświetlenia sceny użyte było dynamiczne generowanie oświetlenia, reagujące na poruszające

się obiekty jak i samego bohatera. W tej samej wersji dodano również narzędzia do szybkiej implementacji logiki gry.

Najnowsza piąta wersja, dodała przełomową funkcję, silnik Nanite. Pozwala on na importowanie modeli w najlepszej możliwej jakości oraz zmniejszanie ich jakości w locie, wedle potrzeby. Jeśli obiekt zajmuje większą powierzchnię ekranu, model rysowany jest z większą ilością wielokątów, jeżeli jest mniej widoczny jest on używany z mocno ograniczoną ich liczbą. Z perspektywy programisty jest to wygodne i nie wymaga ręcznej implementacji LOD, oraz jednocześnie samo dba o optymalizację modeli w całej grze.

Wymagania sprzętowe silnika są wysokie, mogą one być zbyt wysokie dla niektórych komputerów. Celem jest stworzenie wizualizacji, którą można będzie uruchomić na większości urządzeń, ale nie można wymagać od użytkownika posiadania nawet dedykowanej karty graficznej. Silnik byłby idealny jeżeli głównym celem byłaby realistyczna grafika, ale niestety jest on narzędziem o zbyt wysokich wymaganiach sprzętowych, których niektóre urządzenia klientów nie mogłyby sprostać.

## Unity3D

Unity jest silnikiem gier stworzonym na potrzeby tworzenia gier dla systemu Mac OS. Potem został zmodyfikowany do wspierania innych platform takich jak: komputery stacjonarne, konsole oraz platformy mobilne. Dodane zostało również wsparcie dla gier wirtualnej rzeczywistości. Z jego pomocą możliwe jest stworzenie gier zarówno dwuwymiarowych jak i trójwymiarowych. Znany jest ze swojej prostoty i łatwości obsługi.

Silnik nie jest przeznaczony do tworzenia gier AAA, lecz jest nastawiony na szybkie i łatwe tworzenie prostszych gier nastawionych na ciekawą rozgrywkę, albo jej mechanikę. W tego typu grach postać grafika nie jest problemem, dlatego często jest on używany do tworzenia niezależnych gier komputerowych. Często również jest on używany do tworzenia gier mobilnych, gdzie szybkość tworzenia gier jest dość ważna, by móc nadążyć za nieustannie zmieniającymi się trendami.

W obecnej wersji silnika, gry programowane mogą być w języku C#. Pozwala to na lepsze użycie obiektowości oraz unikalnych funkcji języka C# nastawionych na programownie obiektowe. Środowisko programistyczne do tworzenia gier jest dostępne, dla systemów operacyjnych Windows, GNU/Linux oraz Mac OS, a tworzone projekty mogą być przeznaczone na każdą możliwą wspieraną platformę.

Możliwe jest stworzenie gry wieloosobowej. Dzięki zastosowaniu wielu wirtualnych kamer możliwe jest zaimplementowanie gry wieloosobowej na podzielonym ekranie. Silnik posiada warstwę sieciową, dzięki której można tworzyć sieciowe gry wieloosobowe, pozwalające na grę dwóch lub więcej graczy przez internet. Pozwalają one również na tworzenie serwerów gry wieloosobowej.

Liczba edytorów i narzędzi jest mniejsza i mniej wyspecjalizowana dlatego ten sam zestaw narzędzi można wykorzystać w większości projektów, niezależnie od gatunku tworzonej gry. Można tworzyć animacje dwuwymiarowe oraz trójwymiarowe, a wbudowany edytor animacji, pozwala na płynne przełączanie się pomiędzy różnymi animacjami.

Wbudowany edytor interfejsu użytkownika pozwala na tworzenie interaktywnych ekranów, z różnego rodzaju przyciskami, suwakami oraz innymi wirtualnymi metodami wprowadzania i wyświetlania danych. Pozwalają one na stworzenie zarówno interfejsu użytkownika widocznego podczas gry, jak i bardziej złożonych ekranów wyboru, czy ekranów sterowania ustawieniami gry.

Silnik fizyczny posiada podstawowe funkcjonalności, pozwalające na tworzenie gier opartych o fizykę. Obiekty gry mogą posiadać wagę, wyporność, oraz implementować cechy fizycznych materiałów z których są zrobione. Fizyczne obiekty mogą być połączone ze sobą wiązaniami, imitującymi połączenia przedmiotów w prawdziwym świecie, albo te zupełnie wirtualne. Możliwe jest emulowanie działania zawiasu, lin, szyny czy sprężyn. Wszystkie z nich są sparametryzowane co pozwala na ich dostosowanie, a nawet tworzenie własnych połączeń fizycznych.

Zintegrowanym środowiskiem programistycznym na platformie Windows i Mac OS, jest Visual Studio, nie jest ono dostępne na systemy operacyjne GNU/Linux. Istnieje wsparcie w postaci dodatków i wtyczek dla Visual Studio Code, które pozwalają na wygodne pisanie kodu i skryptowania na wszystkich wymienionych platformach.



## Godot

Godot jest silnikiem na licencji otwartego oprogramowania, co czyni go w pełni darmowym. Został stworzony przez argentyńskich programistów na potrzeby gier komputerowych w Ameryce Łacińskiej. Wspiera on większość znanych systemów operacyjnych: Windows, macOS, Linux oraz różne wersje BSD. Dodatkowo wspiera on platformy mobilne, oraz uruchamianie gier w przeglądarce. Jest przeznaczony do tworzenia małych i średnich gier, oraz grafika przez niego generowana nie jest w stanie dorównać tej, reprezentowanej przez obecne gry AAA.

Gry można programować w wielu językach, wspierany jest unikalny język skryptowy GDScript, napisany na potrzeby silnika, oraz język c++ i c#. Dodatkowo dzięki wparciu dla wiązań językowych możliwe jest pisanie kodu w językach takich jak: Rust, Nim, JavaScript, Haskell, Coljure, Swift. Dodatkowo możliwe jest programowanie wizualne, pozwalające na składanie logiki programu, bez znajomości języków programowania. Zintegrowany edytor tekstu, wspiera podstawowe funkcje, środowiska programistyczne takie jak: formatowanie kodu, podświetlanie pisowni, uzupełnianie kodu, wbudowany jest również debugger pozwalający na tworzenie punktów wstrzymania i uruchamianie programu, linkują po linijce.

Wspierane jest tworzenie gier dwuwymiarowych i trójwymiarowych, gdzie oba tryby mogą pracować jednocześnie, oddzielnie od siebie nawzajem. Stworzony został system animacji, dla obrazów dwuwymiarowych i modeli trójwymiarowych. Animacja szkieletowa jest wspierana w obu przypadkach, animacje obrazu posiadają dodatkowe unikalne dla obrazów animacje. Obrazy można skalować, obracać i edytować ich kolorystkę w czasie trwania animacji.

Kolejnymi wspieranymi elementami jest tworzenie graficznego interfejsu użytkownika. Czasami ten sam silnik używany jest właśnie, do tworzenia programów użytkowych, zamiast gier komputerowych. Wspierana jest również, wielowątkowość, statyczne oświetlenie sceny, oraz efekty cząsteczkowe.

Godot przeznaczony jest to tworzenia małych i średnich projektów, oraz daje użytkownikowi wiele możliwości. Jest rozwiązaniem na licencji otartego oprogramowania, dzięki czemu posiada duże wsparcie wolontariuszy. Jest to dobry wybór do stworzenia projektu wizualizacji architektonicznej.

## CryEngine

Jest silnikiem zaprojektowanym na potrzeby gry Far Cry, przez firmę Crytek. Jego kolejna iteracja została użyta to gry Crysis, która jest znana z bardzo realistycznej grafiki, oraz jest uznawana za jeden z największych przeskoków graficznych i technologicznych w historii gier trójwymiarowych. Silnik ten wyprzedzał swoje czasy oraz pozwalał na generowanie realistycznej grafiki i posiadał fizykę, która pozwalała na niespotykaną dotąd interakcję z otoczeniem.

Silnik pozwala na tworzenie gier AAA, z wymagającą i realistyczną grafiką. Był tworzony z nastawieniem na gry FPS, co odbija się w doborze narzędzi do niego dołączonych. Tworzenie postaci jest proste dzięki narzędziom do ich animacji. Istnieje osobny system do szkieletowych animacji postaci, a animacje te mogą być parametryzowane, co pozwala na dogłębsze dostrajanie ich. Przydatnym dodatkiem jest też osobny edytor animacji twarzy.

W tworzeniu rozgrywki FPS, pomaga generacja kuloodpornych zasłon, oraz częściowo zniszczalne otoczenie. Niektóre obiekty mogą być zniszczalne, a nawet po zniszczeniu nadal być poddawane fizyce gry. Elementami fizycznymi w grze mogą być również wszelkiego rodzaju liny, pozwalające na łączenie obiektów, oraz tworzenie zniszczalnych mostów. Dynamiki rozgrywki dodać może dynamiczne wyszukiwanie ścieżek, pozwalające przeciwnikom na zbliżenie się do postaci gracza.

Wsparcie również dostały pojazdy, których poruszanie się można łatwo zaimplementować, oraz też posiadają dedykowane sobie narzędzia. Edytor pojazdów pozwala na edycję oraz dostrojenie parametrów istniejących pojazdów, albo stworzenie nowego. Systemy pomagające w tworzeniu pojazdów, uzupełnia system budowania dróg na mapie, dzięki któremu w szybki sposób można narysować i wydzielać drogi na mapie świata gry.

Przez lata zostało stworzonych wiele narzędzi, które ułatwiają tworzenie gier w wielu aspektach. Pozwalają na monitorowanie wydajności i parametrów gry podczas pracy, tworzenie materiałów i tekstur, obszerna baza gotowych modeli trójwymiarowych. Istnieje osobna grupa narzędzi pozwalająca na tworzenie i animowanie postaci. Istnieją, też narzędzia przeznaczone do efektów dźwiękowych oraz ich odpowiedniego rozmieszczenia w przestrzeni gry.

## **2.4 Konfigurator**

Konfigurator, dawniej nazywany konfiguratorem produktu, jest narzędziem do tworzenia unikalnej konfiguracji produktu posiadającego wiele opcji konfiguracyjnych. Produkty te posiadają wiele opcji wyboru, niektóre opcjonalne, inne wymagane, a inne wykluczające się nawzajem. Pozwala on na dogłębne dobranie i skonfigurowanie produktu, odpowiadającego klientowi, gdzie efektem końcowym jest gotowa konfiguracja, kompatybilna z systemem. Może on być przeznaczony dla samego klienta, albo być narzędziem wprowadzania dla pracownika jego obsługującego. Ma on na celu oddelegowanie zadania tworzenia konfiguracji z pracownika, na program komputerowy i samego klienta.

Dla klienta może on być interaktywnym katalogiem pokazującym zbiór produktów, oraz wszelkich ich opcji konfiguracyjnych. Czasami klient do końca nie sprecyzował jeszcze swojej wizji i wymagań, ani jak produkt końcowy miałby wyglądać. Poprzez kolejne kroki tworzenia kompozycji, klient jest zapoznawany z zakresem możliwości edycji i opcji dostrojenia produktu końcowego. W ten, koherentny, sposób udziela klientowi informacji, niezbędnych do podjęcia najlepszego dla siebie wyboru. Na końcu informacja ta może być w zautomatyzowany sposób przekazana do realizacji.

Dla pracownika, może być narzędziem pomagającym w pracy i pozwalającym na wprowadzenie gotowej, zwalidowanej konfiguracji do systemu. Może przeprowadzić on pracownika przez kolejne etapy, oraz poinformować go o stanie magazynu, albo innych niedogodnościach, dotyczących łańcucha dostaw. Dzięki temu, pracownik jest odciążony, a swoją uwagę może skupić na zadawaniu bardziej szczegółowych pytań klientowi.

### **2.4.1 Typy konfiguratorów**

Ze względu na zastosowaną technologię i grafikę, wydzielić można typy konfiguratorów produktów. Nie jest to jedyny sposób w jaki można je podzielić.

#### **Prerenderowane wizualizacje**

Są to wizualizacje złożone z wygenerowanych wcześniej obrazów. Zdjęcia oraz prerenderowane obrazy charakteryzują się najlepszą jakością grafiki, oraz są zasobo-oszczędne (z punktu widzenia użytkownika). Zdjęcia jednak nie mogą być mocno zmodyfikowane po ich zrobieniu, zazwyczaj zmieniane są kolory elementów, albo niektóre elementy są zmieniane na inne. Każda wizualna konfiguracja oraz jej możliwości muszą zostać określone z góry, oraz wzięte pod uwagę.

#### **Technologie interaktywne**

Technologie cechującą się większą interaktywnością, pozwalające na obracanie obrazu lub przedmiotu, oraz większe edycje otoczenia. Są one niestety najbardziej wymagające pod względem zasobów, oraz nie zawsze urządzenie klienta może tym wymaganiom sprostać. Do ich stworzenia używane są czasami silniki do gier komputerowych.

## **2.5 Architektura wnętrz**

Architektura wnętrz jest dziedziną zajmującą się projektowaniem oraz wystrojem wnętrz budynków. Obejmuje to zdefiniowanie stylu, i ogólnej kolorystyki wnętrza jak i rozplanowanie rozmieszczenia mebli. Odpowiednie, do jego zastosowania, zagospodarowanie wnętrza sprawia, że wnętrze jest bardziej funkcjonalne, oraz prezentuje się lepiej. W kontekście tej pracy wymagane będzie stworzenie zestawu projektów pomieszczeń, w taki sposób, aby

najlepiej oddawały najróżniejsze kompozycje i style pomieszczeń. Zbiór ten powinien oddawać jak najszerzą gamę stylów, aby klient mógł w nim znaleźć coś co mniej więcej oddaje jego własne wnętrze, aby lepiej do niego dopasować swój własny produkt.

### **2.5.1 Istniejące już rozwiązania**

Trudno znaleźć rozwiązanie, które byłoby używane przez więcej niż jedną firmę. Zazwyczaj każda firma z kolei tworzy własną implementację od podstaw. Przyczyną może być różnorodność produktów, oraz jego otoczenia. Każda osobna implementacja musiałaby posiadać swoje własne tekstury, pomieszczenia, modele. Czasami nawet zakres edycji i rozmieszczenia kamery byłyby zupełnie inne. Ilość włożonej pracy jest niewiele mniejszy niż zaprogramowanie własnego rozwiązania od podstaw. Właśnie dlatego każda branża z kolei decyduje się na tworzenie własnej, skrojonej na miarę, implementacji. Wydzielić można pewne części wspólne i funkcjonalności, które mają niektóre z nich.

### **2.5.2 Interaktywny widok trójwymiarowy**

Widok trójwymiarowy pozwalający na ograniczone obracanie kamery, oraz oglądanie produktu w swoim naturalnym otoczeniu. Pozwala to klientowi na bardziej wizualne podejście do decyzji, oraz skupienie się na innych aspektach i wizualnych cechach produktu, takich jak kolor, rozmiar czy jego kształt. Pokazanie produktu blisko przedmiotów o wiadomych rozmiarach, pozwala na lepszą ocenę rozmiaru produktu, który jeszcze fizycznie nie istnieje.

### **2.5.3 Widok projektowania**

Widok przedstawiający produkt w uproszczonej formie, przypominający rysunek techniczny. Pozwala na dostosowanie wymiarów produktu do wymiarów istniejącego już pokju lub zabudowy. Używany jest kiedy wymiary otoczenia są znane, albo wymiary produktu mogą się mocno różnić zależnie od dobieranego typu lub modelu. Jego zaletą jest przejrzystość, oraz uproszczony, czysty wygląd pozwalający się skupić na jego wymiarach.

### **2.5.4 Katalog produktów i jego opcji**

Częścią konfiguratora może być osobny widok, albo okno pokazujące użytkownikowi inne produkty, albo inne jego wersje. W ten sposób pokazujemy klientowi asortyment i jakie on ma inne możliwości do wyboru. Tego typu widok może pokazywać różne warianty rozmiarowe, stylistyczne czy kolorystyczne produktu. Kolejne wybory można reprezentować poprzez miniaturowy obraz, model trójwymiarowy, albo opis tekstowy, a nawet krótki tytuł. Ważne jest aby kolejne wybory były ustawione w odpowiedniej kolejności, od najogólniejszego do najbardziej szczegółowego.

### **2.5.5 Ekran podsumowania**

Ekran wypisujący wszystkie dokonane zmiany, oraz opcje dodatkowe. Ma na celu pokazanie klientowi ceny produktu końcowego oraz przekazanie innych ważnych dla niego informacji, takich jak czas realizacji, dane kontaktowe oraz numer zamówienia. Po ostatnim kliknięciu, konfiguracja produktu powinna zostać dodana do systemu oraz przetworzona.

## **2.6 Panele ściennie**

Panele są płaskimi fragmentami materiału, o jednokowych kształtach, którymi wykłada się podłogi, sufity oraz ściany budynków. Na początku robiło się to w celu odizolowania termicznego, kamiennych ścian i podłóg. Wraz z czasem, panele zaczęły pełnić rolę dekoracyjną. Wykonywane one były na początku z drewna, wraz z rozwojem technologii i wynalezieniem nowych materiałów zaczęto je tworzyć z tworzyw sztucznych. Dodatkowo wyłożenie powierzchni miększym materiałem, zmniejsza echo w pomieszczeniu. Efekt ten jest użyty w celu całkowitego

wytlumienia echa, za pomocą specjalnych paneli wygłuszających. Paneli można również używać izolacji termicznej budynku, tzw. ocieplania budynku.

### **2.6.1 Tapety**

Tapety są cienką okładziną naklejaną na wnętrza budynków w celach dekoracyjnych, wykonane głównie z papieru, ale również tkanin, skóry i tworzyw sztucznych. Dzięki swojej elastyczności i cienkości mogą być zwijane w rolki na czas transportu, co znacznie ułatwia ich transport. Kolejnym czynnikiem jest ich bardzo mała waga w przeciwieństwie do płytek i paneli nie wymagają mocnego kleju do instalacji. Przykleja się je do ściany za pomocą specjalnego kleju do tapet.

Tapety mogą mieć różne kolory i wzory, dzięki ich różnorodności mogą w duży stopniu zmienić sposób w jaki postrzegane jest pomieszczenie. Jaśniejsze kolory wizualnie powiększają pomieszczenie, a ciemniejsze je zmniejszają. Pionowe wzory sprawiają, że pomieszczenie wydaje się być wyższe, a poziome poszerzają je. Dodatkowo mogą one posiadać wytłoczenia, oraz własną teksturę, w ten sposób podkreślając swój wzór, albo używając jednolitego koloru, oraz swojego własnego wytłoczenia jako subtelnego wzoru.

### **2.6.2 Wygłuszające**

Wytłumianie pomieszczenia polega na zmniejszaniu echa w pomieszczeniu poprzez pokrywanie jego ścian izolatorami akustycznymi. Jest to materiał, który dzięki swoim właściwościom, zapobiega niepożądanemu przenikaniu dźwięków oraz odbijaniu ich tworząc echo. Zależnie od zastosowania materiały te mogą być umieszczone wewnątrz ścian budynku, albo na wewnętrznej stronie ścian pomieszczenia. Umieszczenie ich wewnątrz ograniczy przedostawanie się dźwięków z wnętrza pomieszczenia do zewnątrz i na odwrót, a umieszczenie ich na wewnętrznej stronie ścian, efektywniej zmniejszy echo wewnątrz pomieszczenia.

Izolatory akustyczne mogą występować w różnych formach. Jeżeli jego grubość na to pozwala, może to być giętki materiał sprzedawany w rolkach, który można przyklejać do ścian jak tapetę. Wariant o większej grubości jest sprzedawany w formie jednakowych paneli na ścianę, które posiadają falowane wyżłobienia polepszające ich właściwości tłumiące. Są one czasami sprzedawane w różnych wariantach kolorystycznych oraz pozwalają na tworzenie kolorowych wzorów na ścianę. Ostatnim typem są struktury tłumiące takie jak pułapki dźwiękowe, są to zazwyczaj pudła o różnych wymiarach, zazwyczaj wykonane z drewna z odpowiednio wywierconymi dziurami i szczelinami. Pełnią one rolę odwrotną do pudła rezonansowego, łapiąc i wytłumiając całkowicie dźwięki.

### **2.6.3 Dekoracyjne**

Inną funkcją jaką mogą pełnić panele ściennie jest dekoracja wnętrza, ich zadaniem jest przełamanie monotonii dużych jednolitych ścian w pomieszczeniu. Dodatkowo, są one mniej inwazyjne dla ściany niż tapeta lub glazura, której tak łatwo nie można wymienić na nową. Rozwój technologii i materiałoznawstwa pozwala obecnie na tworzenie klejów i mocowań wytarczająco mocnych na przytwierdzanie paneli do ściany, bez wiercenia dziur w ścianie, ani zostawiania śladów kleju na niej.

Ważne aby panele miały specjalny kształt, pozwalający na szczelne pokrycie powierzchni nazwaną parkietażem. Parkietaż formeny składa się z identycznych wielokątów foremnych, w którym jednakowa liczba figur, schodzi się w wierzchołku. W praktyce oznacza to, że wyprodukować będzie jedynie panele o w kształcie jednej figury geometrycznej. Trzema znanymi figurami geometrycznymi tworzącymi parkietaż foremny są kwadrat, trójkąt równoboczny oraz sześciokąt foremny. Nie trzeba się ograniczać parkietażu foremnego, dlatego możliwe jest użycie innych kształtów, które nie są podstawowymi foremnymi wielokątami.

### **3 Analiza i założenia projektu**

#### **3.1 Wymagania sprzętowe**

#### **3.2 Zakres funkcjonalności**

##### **3.2.1 Katalog asortymentu**

##### **3.2.2 Widok produktu w scenie**

##### **3.2.3 Prezentacja oferty i możliwości konfiguracji**

##### **3.2.4 Edycja pomieszczenia**

##### **3.2.5 Poruszanie kamerą**

##### **3.2.6 Zmiana oświetlenia**

#### **3.3 Określenie stopnia interaktywności**

##### **3.3.1 Zbiór elementów interaktywnych i stopień ich interaktywności**

#### **3.4 Strona internetowa**

##### **3.4.1 Znaczenie osiągalności informacji o produkcie**

##### **3.4.2 Konfigurator jako katalog produktów**

### **4 Realizacja projektu**

#### **4.1 Modelowanie elementów 3d**

##### **4.1.1 pomieszczenia**

#### **4.2 Części projektu**

##### **4.2.1 Logika i schemat działania**

##### **4.2.2 Interfejs użytkownika**

##### **4.2.3 Widok pomieszczenia**

##### **4.2.4 Hierarchia i diagramy klas**

##### **4.2.5 Osadzenie programu wynikowego na stronie internetowej**

### **5 Wnioski**

## **6 Literatura**

## **7 Załączniki**