

Wyższa Szkoła Komunikacji i Zarządzania
w Poznaniu

Mateusz Marchwicki

Spatial video mapper - Program do tworzenia projekcji przestrzennych

Praca dyplomowa inżynierska

Promotor: dr inż. Ewa Idzikowska

Instytut Informatyki

Instytut Informatyki

kierunek: Informatyka

specjalność: Technologie i zastosowania Internetu

Poznań 2011

Spis treści

1	Wstęp	4
2	Analiza rynku dostępnych aplikacji	6
2.1	Modul8	6
2.2	VDMX5	7
3	Projekcja przestrzenna	8
3.1	Definicja	8
4	Założenia projektowe	8
4.1	Funkcje aplikacji	8
4.1.1	Operacje na warstwach	9
4.1.2	Import mediów	9
4.1.3	Operacje na mediach	9
4.2	Wstępna budowa programu	10
4.3	Diagram klas	11
5	Wykorzystane technologie i narzędzia	11
5.1	Język C#	11
5.2	Windows Presentation Foundation	12
5.3	Visual Studio 2010	12
6	Implementacja poszczególnych klas aplikacji	13
6.1	LayerControl	13
6.2	Layer	13
6.3	Media	13
6.4	MediaTexture	13
6.5	MainWindow	13
6.6	RenderWindow	14
6.7	LayerNameChangeDialog i MediaNameChangeDialog	14
7	Interfejs użytkownika	14
7.1	Pasek Menu	14
7.2	Zakładka Transform	15
7.3	Zakładka Media Controller	16
7.4	Lista warstw	17
7.5	Lista mediów	18
7.6	Lista media warstwy	19
7.7	Podgląd projekcji	19
7.8	Okno projekcji	20
8	Przykładowe zastosowanie aplikacji	20
9	Wymagania sprzętowe oraz programowe	21
10	Podsumowanie pracy	21

Spis rysunków

2.1	Wygląd interfejsu użytkownika programu Modul8
2.2	Wygląd interfejsu użytkownika programu VDMX5
3.1	Przykładowa projekcja przestrzenna
4.3	Diagram klas
7.1.1	Pasek Menu Edit
7.1.2	Pasek Menu View
7.2	Zakładka transform
7.3	Zakładka Media Controller
7.4	Lista Layers
7.5	Lista Media
7.6	Lista Layers Media
7.7	Podgląd okna projekcji
8.1	Utworzone warstwy na elementach przestrzennych
8.2	Projekcja przestrzenna przy zastosowaniu aplikacji Spatial Video Mapper

1 Wstęp

Współcześnie tworzenie prezentacji dla klienta, organizowanie multimedialnych pokazów wymaga sporych nakładów sprzętowych takich jak: projektory, oświetlenie i oczywiście zaplecze specjalistów zajmujących się obsługą powyższego sprzętu. To wszystko sprawia iż koszty są bardzo duże. Aktualnie bardzo szybko rozwijającą się techniką jest tak zwany perspective video mapping czyli przestrzenna projekcja stworzona specjalnie pod powierzchnie na której będzie wyświetlana.

Przygotowanie takiej projekcji składa się na parę etapów. Pierwszym z nich jest dokładne sfotografowanie przestrzeni i wykonanie wszystkich pomiarów płaszczyzn na których projekcja będzie się odbywać. Podczas wykonywania fotografii należy zwrócić uwagę na to aby aparat znajdował się w dokładnie tym samym miejscu w którym będzie się znajdować projektor. Także optyka obiektywu aparatu powinna być jak najbardziej zbliżona do tej którą będzie posiadać projektor aby uniknąć późniejszych zniekształceń obrazu. Kolejnym krokiem jest stworzenie w programie do grafiki trójwymiarowej przestrzeni odwzorowanej ze zdjęć oraz wykonanych pomiarów. Następnie stworzenie tekstury z wideo klipu bądź też grafiki. Stworzona tekstura musi zostać naniesiona na odwzorowany w trójwymiarze element, a następnie cała sekwencja musi zostać zobrazowana do jednego klipu wideo. Zazwyczaj zajmuje to od kilkunastu sekund do paru godzin w zależności od rozdzielczości używanych mediów, mocy obliczeniowej komputera oraz złożoności stworzonej przestrzeni trójwymiarowej. Gdy posiadamy już zobrazowany plik wideo sekwencji oraz odpowiednio ustawiony projektor możemy odtworzyć stworzony klip wideo, który jeśli nie zaniedbaliśmy któregoś z powyższych kroków powinien pokryć całą powierzchnie oraz obraz powinien być widoczny bez zniekształceń. W innym wypadku należeć będzie powtórzyć wszystkie powyższe kroki co owocować będzie w kolejne godziny żmudnej pracy w programach do grafiki trójwymiarowej.

Aplikacja „Spatial video mapper” w dalszej części zwana „aplikacją” ma na celu pominięcie wszystkich wymienionych wyżej procesów przygotowywania projekcji. Dzięki temu klipy wideo bądź też grafiki w dalszej części pracy zwane „mediami”, będą mogły być w czasie rzeczywistym wyświetlane za pomocą projektora na dowolnej prostokątnej powierzchni pod dowolnym kątem bez potrzeby czasochłonnego obrazowania. Dzięki wykorzystaniu możliwości silnika graficznego *Windows Presentation Foundation* wykorzystującym technologię *DirectX* możliwe będzie tworzenie elementów przestrzennych oraz nadawanie im tekstur w czasie rzeczywistym. W każdej chwili działania aplikacji każdy parametr projekcji będzie mógł zostać zmieniony. Wszystkie media będą mogły być odwzorowane na stworzonych płaszczyznach. Dzięki wsparciu kart graficznych wszystkie obliczenia odbywać się będą bezpośrednio na karcie graficznej co znacząco odciąża procesor. Intuicyjny interfejs także wykonany przy pomocy silnika *Windows Presentation Foundation* sprawi że każdy bez problemu obsłuży aplikację. Więc czy to na akcji promocyjnej firmy, spektaklu audio-wizualnym, czy też w domu gdzie nie zawsze znajdzie się powierzchnia do projekcji, aplikacja ta znajdzie swoje zastosowanie.

Praca składa się z 9 rozdziałów. Omawiają one kolejne etapy powstawania aplikacji. Pracę zaczyna wstęp, który wprowadza w temat projekcji przestrzennych oraz problemów występujących podczas tworzenia takich projekcji.

Rozdział drugi przedstawia dwie istniejące aplikacje, służące do tworzenia projekcji przestrzennych i przedstawia ich funkcjonalność.

W rozdziale trzecim zdefiniowana zostanie projekcja przestrzenna w sposób ścisły mówiąca czym jest projekcja tego typu.

Rozdział czwarty porusza założenia projektowe uwzględnione podczas projektowania aplikacji, jej funkcjonalność oraz możliwe opcje.

Rozdział piąty omawia technologie oraz narzędzia użyte podczas tworzenia aplikacji.

W rozdział szósty przedstawia klasy składające się na aplikację.

W rozdziale siódmym przedstawiony jest interfejs użytkownika. Omówione zostaną wszystkie elementy składające się na interfejs użytkownika oraz możliwości jakie daje.

W rozdziale ósmym zostanie przedstawiona przykładowa projekcja przestrzenna stworzona przy użyciu aplikacji „Spatial video mapper”

Rozdział dziewiąty porusza temat wymagań sprzętowych, oraz programowych niezbędnych do poprawnej pracy aplikacji.

W podsumowaniu zawarte są końcowe wnioski, oraz możliwości dalszego rozwoju aplikacji „Spatial video mapper”, opis problemów zaistniałych podczas projektowania i implementacji programu.

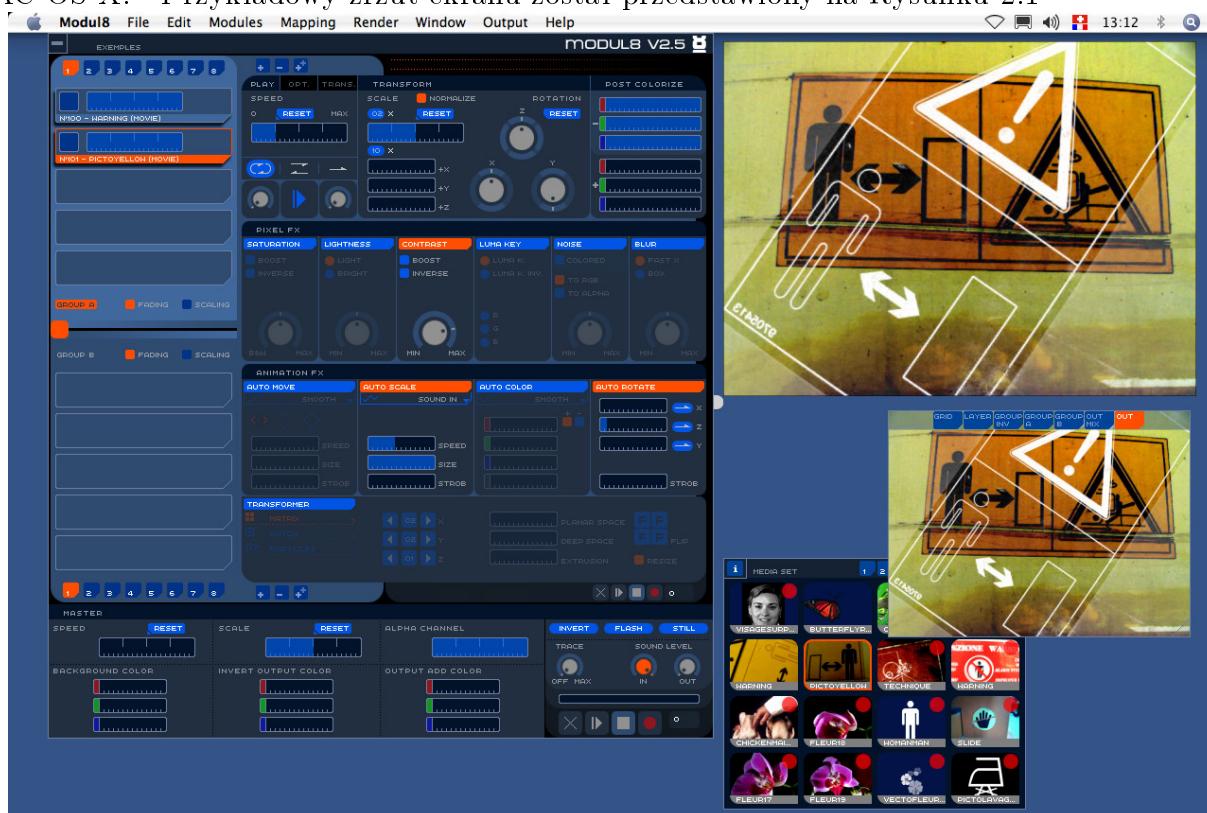
2 Analiza rynku dostępnych aplikacji

Na rynku komputerowym istnieją 2 programy umożliwiające tworzenie projekcji przestrzennych. Oba programy przeznaczone są dla VJ-ów (z ang. „Video Jockey” czyli osoba prezentująca nagrania wideo w dyskotekach lub TV) Podstawowe funkcje powyżej wymienionych aplikacji to:

- mieszanie wielu klipów wideo
- wideo efekty
- zapisywanie i odtwarzanie projektów
- tworzenie projekcji przestrzennych
- sterowanie aplikacją za pomocą zewnętrznych kontrolerów MIDI (ang. “Musical Instrument Digital Interface”)

2.1 Modul8

Jest aplikacją służącą do celów komercyjnych, stworzoną dla systemu operacyjnego MAC OS X. Przykładowy zrzut ekranu został przedstawiony na Rysunku 2.1



Rys 2.1 Wygląd interfejsu użytkownika programu Modul8

Posiada bardzo rozbudowany interfejs użytkownika posiadający:

- wirtualne potencjometry i suwaki
- podgląd projekcji

- liste zaimportowanych mediów
- zakładki efektów
- zakładki parametrów odtwarzania
- tworzenie projekcji przestrzennych

Podstawowa licencja dla dwóch stacji roboczych kosztuje 299 €.

2.2 VDMX5

VDMX5 jest aplikacja także komercyjną dla systemu operacyjnego MAC OS X. Aplikacja ta korzysta z biblioteki graficznej Quartz Composer. Posiada bardzo rozbudowany interfejs użytkownika posiadający wiele opcji pozwalających na manipulacje klipami wideo. Przykładowy zrzut ekranu został przedstawiony na Rys 2.2



Rys 2.2 Wygląd interfejsu użytkownika programu VDMX5

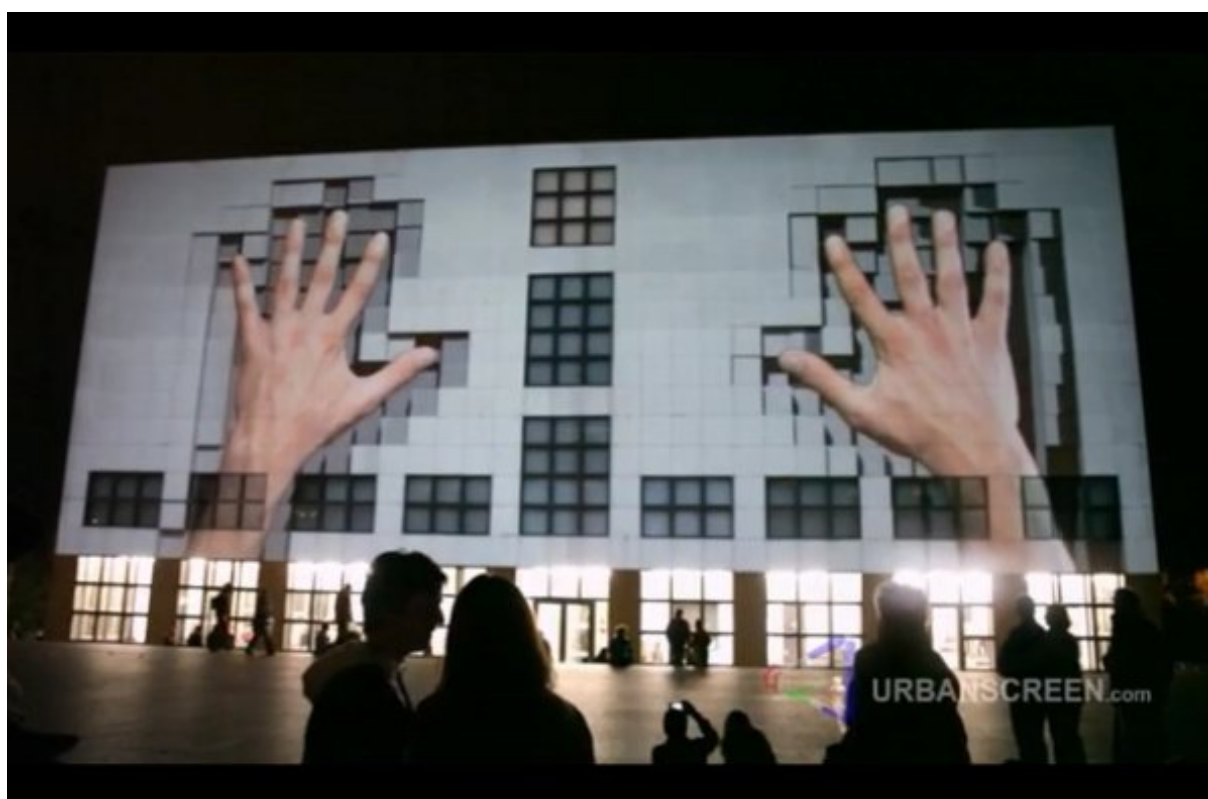
Aplikacja ta posiada możliwość instalacji rozszerzeń, jeden z takich dodatków daje możliwość tworzenia projekcji przestrzennych.

3 Projektcja przestrzenna

Projektcja przestrzenna często stosowana współcześnie w sztuce pozwala nadawać elementom architektury np. budynkom “życia” poprzez światło, kolor i ruch. W tym rozdziale zostanie zdefiniowane czym jest projektcja przestrzenna, oraz pokazane zostanie przykład takiej projektji.

3.1 Definicja

Projektcja przestrzenna jest techniką przetwarzania obrazu w sposób uwzględniający architektoniczne uformowanie płaszczyzny na której odbywa się projektcja. Obraz musi zostać przetworzony w taki sposób, aby rzucana projektja odzwierciedlała cechy ukształtowania architektonicznego płaszczyzny. Przykład takiej projektji przedstawia rysunek 3.1



Rys 3.1 Przykładowa projektja przestrzenna

4 Założenia projektowe

W powyższym rozdziale poruszone zostaną założenia projektowe uwzględnione podczas projektowania aplikacji, jej funkcjonalność oraz opcje.

4.1 Funkcje aplikacji

Głównym celem aplikacji jest umożliwienie użytkownikowi tworzenie projektji przestrzennych na dowolnych prostokątnych płaszczyznach w czasie rzeczywistym.

4.1.1 Operacje na warstwach

Operacje na warstwach powinny użytkownikowi pozwalać na:

- Tworzenie nowych warstw
- Usuwanie warstw
- Zmianę kształtu utworzonej warstwy
- Zmiany nazw warstw
- Wyświetlanie siatki pomocniczej ułatwiające ustawienie
- Wyświetlanie klipów wideo oraz obrazów na danej warstwie

4.1.2 Import mediów

Aplikacja musi umożliwić użytkownikowi importowanie mediów takich jak obrazy lub klipy wideo, które następnie będą mogły służyć do nałożenia obrazu na warstwę. Podczas importowania aplikacja musi zostać sprawdzona zgodność importowanych plików z obsługiwanyymi formatami.

4.1.3 Operacje na mediach

Operacje na mediach muszą pozwalać użytkownikowi na:

- Odtwarzanie
- Pauzowanie
- Zatrzymywanie
- Przewijanie klipu wideo
- Zmianę prędkości odtwarzania
- Ustawienie przeźroczystości

Zmianę opcji takich jak:

- zapętlanie klipu wideo
- wyciszenie fonii
- wyłączenie podglądu puszczanego klipu

4.2 Wstępna budowa programu

Program musi zostać stworzony w taki sposób aby poszczególne elementy aplikacji były od siebie oddzielone. Dzięki temu podejściu w przyszłości wprowadzanie zmian i rozszerzanie funkcjonalności aplikacji odbywać się będzie przy jak najmniejszej ingerencji w istniejący już kod. Aplikacja składa się z następujących elementów:

- Interfejs użytkownika
- Warstwy
- Tekstury
- Kontroler warstw
- Kontroler mediów
- Okno projekcji

Interfejs aplikacji powinien być jak najbardziej wydajny i prosty w użyciu. Ponieważ aplikacja może być uruchamiana na wielu różnych konfiguracjach sprzętowych aplikacja posiada możliwość wyłączenia podglądów w głównym oknie aplikacji, co znacząco przyspiesza działanie aplikacji. Tekstury będą mogły być transparentne poprzez zmianę parametru przeźroczystości. Także będą ponumerowane pod względem kolejności występowania na warstwie. Dzięki temu w trakcie działania programu ich kolejność będzie mogła być zmieniana na słabszych konfiguracjach sprzętowych.

Warstwy reprezentować będą prostokątne geometrie, których kształt będzie mógł być zmieniany poprzez manipulację każdym z czterech rogów prostokąta.

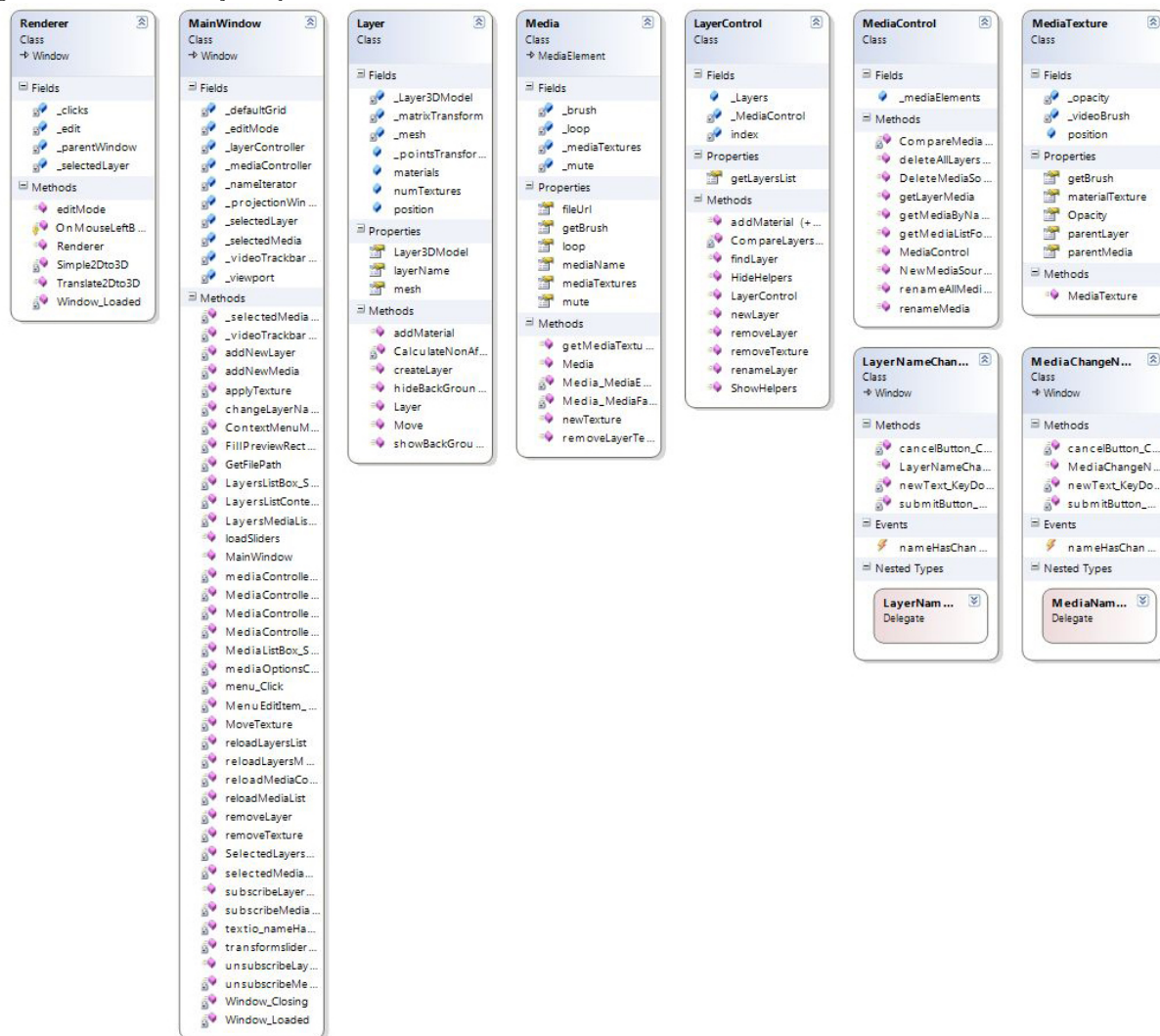
Media będą przechowywać ścieżki źródłowe do plików, nazwy mediów, oraz agregować będą tekstury dla konkretnej warstwy. Podejście takie zostało zastosowane ponieważ wiele warstw może posiadać jeden klip wideo jako swoją teksturę, której przeźroczystość będzie mogła być następnie niezależnie regulowana od pozostałych. Dzięki temu nie będzie potrzeby wczytywania do pamięci tego samego klipu wideo dla każdej nowej warstwy na której ma być odtworzone.

Tekstury będą przechowywać nazwy mediów przez które zostały utworzone, jak i warstw z którymi są skojarzone. Tekstury będą mogły być transparentne poprzez zmianę parametru przeźroczystości. Numeracja pod względem kolejności występowania na warstwie pozwoli na zmianę kolejności ich wyświetlania w trakcie działania programu.

Kontrolery warstw i mediów pełnić będą funkcję agregującą utworzonych warstw oraz mediów. Główną ich rolą będzie wyszukiwanie obiektów tych klas na podstawie ich nazw, innych obiektów skorelowanych z nimi, czy też instancji klas które je tworzą.

4.3 Diagram klas

Diagram klas ilustruje Rys. 4.3



Rys 4.3 Diagram klas

5 Wykorzystane technologie i narzędzia

W tym rozdziale omówione zostaną technologie i narzędzie wykorzystane podczas tworzenia aplikacji.

5.1 Język C#

Jest interpretowanym językiem zorientowanym obiektowo stworzonym dla systemu *Microsoft Windows* o bardzo dużych możliwościach z bardzo dużą liczbą bibliotek wbudowanych w platformę programistyczną *.NET* która jest integralną częścią tego języka.

Wielką zaletą tego języka jest wbudowany komponent o nazwie *Garbage Collector* (z ang. odśmiecacz) którego rolą jest czyszczeniem pamięci z niepotrzebnych obiektów. Dzięki temu nie ma potrzeby ręcznego zwalniania pamięci, ponieważ mechanizm ten automatycznie niszczy obiekty i zwalnia pamięć, gdy w programie istniejący obiekt przestaje być potrzebny do poprawnej pracy aplikacji.

Program napisany w języku kompilowany jest do kodu pośredniego *Common Intermediate Language* (CIL) który jest wykonywany następnie w środowisku uruchomieniowym *.NET*.

Dzięki implementacją środowiska uruchomieniowego *.NET* takich jak *MONO* istnieje możliwość uruchamiania aplikacji w systemach takich jak *Linux*, oraz *MAC OS*.

5.2 Windows Presentation Foundation

Jest silnikiem graficznym wchodzącym w skład platformy programistycznej *.NET* wersji 3.0 bazującym na bibliotece graficznej *DirectX*.

W skład możliwości powyższego silnika graficznego wchodzi:

- Tworzenie bogatych graficznie interfejsów użytkownika
- Obsługę wideo, obrazów oraz dźwięków
- Tworzenie grafiki 2D i 3D
- Tworzenie animacji
- Operacje na dokumentach

Wszystkie elementy języka mogą być deklarowane poprzez język znaczników *XAML* (*Extensible Application Markup Language*) bazujący na technologii *XML* (*Extensible Markup Language*) bądź też dynamicznie w kodzie aplikacji.

Graficzna część silnika wykorzystuje grafikę wektorową, oraz wspiera akcelerację sprzętową, co znacząco odciąża procesor.

5.3 Visual Studio 2010

Jest zestawem narzędzi programistycznych w których skład wchodzi:

- *Microsoft Visual C#*
- *Microsoft Visual C++*
- *Microsoft Visual Basic*
- *Microsoft Visual J#*
- *Microsoft Visual Web Developer*
- *Microsoft Visual F#*

Narzędzie to pozwala na tworzenie aplikacji desktopowych, sieciowych, usług sieciowych oraz serwisów internetowych. Posiada bardzo przejrzysty interfejs, możliwość bezpośredniej integracji z bazami danych *SQL Server*. Rozbudowany debugger oraz system automatycznego uzupełniania pisanego kodu *Intellisense*

6 Implementacja poszczególnych klas aplikacji

W poniższym rozdziale zostaną omówione stworzone klasy na które się składa aplikacja tj. LayerControl, MediaControl, Layer, Media, MediaTexture, MainWindow, RenderWindow, LayerNameChangeDialog, MediaNameChangeDialog.

6.1 LayerControl

LayerControl odpowiada za przechowywanie utworzonych warstw w kolekcjach oraz zarządzanie nimi. Klasa pełni funkcje agregującą oraz przeszukującą kolekcje instancji klasy Layer na podstawie jej nazwy. Ponieważ część operacji na warstwach wymaga operacji na mediach, w konstruktorze przekazywana jest instancja klasy MediaControl umożliwiającą takie operacje.

6.2 Layer

Klasa Layer jest kluczowym elementem aplikacji implementuje operacje na obiektach trójwymiarowych. Instancja klasy Layer zawiera w sobie obiekty takie jak MeshGeometry3D, GeometryModel3D, MatrixTransform3D oraz MaterialGroup które są elementami silnika graficznego WPF. Klasa ta umożliwia transformacje siatki trójwymiarowej poprzez zmianę pozycji każdego z czterech rogów czworokąta oraz dodawaniem materiału tekstury pokrywającego powierzchnię warstwy.

6.3 Media

Klasa Media dziedziczy po klasie MediaElement który jest integralną częścią silnika graficznego WPF. Klasa została rozszerzona o funkcje umożliwiające tworzenie lub usuwanie instancji klasy MediaTexture przechowywanej w kolekcji. Zmianę parametrów odtwarzania takich jak zapętlenie czy wyciszenie fonii. Instancja klasy zajmuje się również przeszukiwaniem kolekcji klasy MediaTexture dla danej warstwy, oraz obsługą zdarzeń występujących gdy klip wideo dobiegnie końca, bądź wystąpi błąd w odczycie mediów.

6.4 MediaTexture

Klasa MediaTexture definiuje parametry tekstury wyświetlanej na warstwie. Posiada właściwości umożliwiające pobranie materiału tekstury który jest używany do wypełniania utworzonych warstw oraz udostępnia funkcje pozwalające na zmianę parametru transparencji tekstury. Instancje klasy przechowują informacje pozwalające ustalić dla jakiej warstwy została utworzona oraz jaki element klasy Media ją utworzył.

6.5 MainWindow

MainWindow jest głównym oknem aplikacji. Dzieli się na kod XAML (z ang. „Extensible Application Markup Language”) który jest językiem opisu interfejsu użytkownika, oraz kodem aplikacji. Do wyświetlania grafiki trójwymiarowej używany jest element o nazwie ViewPort3D należący do platformy programistycznej .NET. Elementy zawarte w klasie MainWindow to głównie obsługa zdarzeń takich jak zmiany stanów kontroltek, kliknięć na listy, oraz elementów menu. W klasie znajdują się także funkcje dotyczące obsługi wszystkich powyżej wymienionych klas, oraz otwierania okna projekcji RenderWindow. Posiada

również podgląd wyświetlanej projekcji, oraz odtwarzanych mediów. W wyświetlanych zakładkach „Transform” oraz „Media Controller” użytkownik może zmieniać parametry projekcji oraz obsługiwać odtwarzane klipy wideo.

6.6 RenderWindow

RenderWindow jest oknem projekcji, okno to jest otwierane na ekranie projektora i wyświetla wszystkie utworzone warstwy. Składa się z kodu XAML oraz kodu aplikacji. Do wyświetlania grafiki trójwymiarowej używany jest element o nazwie ViewPort3D, który jest częścią platformy programistycznej .NET. Podczas tworzenia jej instancji do konstruktora jako parametr przekazywana jest kolekcja utworzonych warstw, oraz instancja głównego okna MainWindow potrzebna do odświeżania stanów kontrolerek w głównym oknie. Dzięki przechwytywaniu zdarzenia kliknięcia myszki na ekranie oraz funkcji umożliwiających translację współrzędnych z 2D do 3D możliwe jest definiowanie współrzędnych warstw poprzez kliknięcia myszką.

6.7 LayerNameChangeDialog i MediaNameChangeDialog

LayerNameChangeDialog i MediaNameChangeDialog są oknami zmiany nazwy warstwy oraz mediów. W konstruktorze przekazywana jest aktualna nazwa, co umożliwia walidację wprowadzanej nazwy.

7 Interfejs użytkownika

W tym rozdziale omówiony zostanie interfejs użytkownika pod kątem wykonywanych operacji oraz prawidłowej obsługi programu. Interfejs użytkownika jest kluczowym elementem każdej aplikacji zarówno internetowej, mobilnej jak i desktopowej. To dzięki intuicyjnemu interfejsowi użytkownik w prosty sposób może stworzyć nową projekcję, zmieniać jej parametry, dodawać i usuwać media. Głównymi elementami interfejsu są dwie zakładki kontrolujące odtwarzane media, suwaki odpowiadające za współrzędne rogów warstw, paski menu oraz trzy listy w których znajdują się zaimportowane media, warstwy, tekstury warstw.

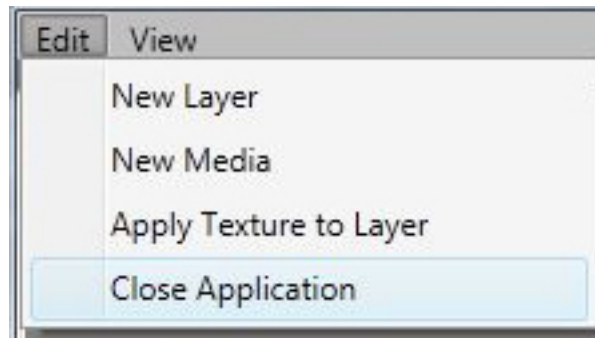
7.1 Pasek Menu

W pasku menu znajdują się 2 elementy Edit oraz View.

W pasku Edit mamy dostęp do funkcji takich jak dodanie nowej warstwy, dodanie nowego klipu wideo bądź grafiki, dodanie tekstury do warstwy oraz zamknięcie aplikacji.

- Dodanie nowej warstwy
- Dodanie nowego klipu wideo bądź grafiki
- Dodanie tekstury do warstwy
- Zamknięcie aplikacji

Rysunek 7.1.1 pokazuje zrzut ekranu paska Edit

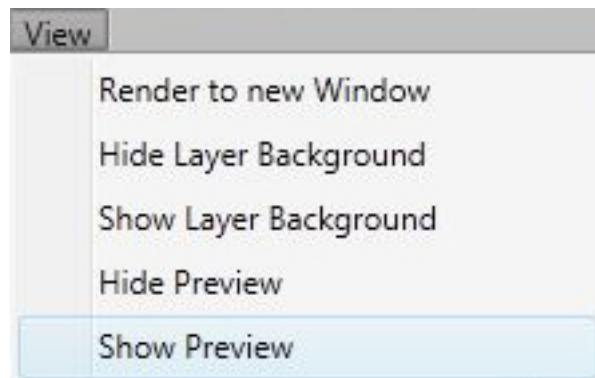


Rys 7.1.1 Pasek Menu Edit

Poprzez pasek View możemy uzyskać dostęp do funkcji pozwalających na zmianę parametrów dotyczących podglądu takich jak:

- Obrazowanie w nowym oknie
- Schowanie pomocniczej siatki
- Pokazanie pomocniczej siatki
- Schowanie podglądu projekcji
- Pokazanie podglądu projekcji

Rysunek 7.1.2 pokazuje zrzut ekranu paska View



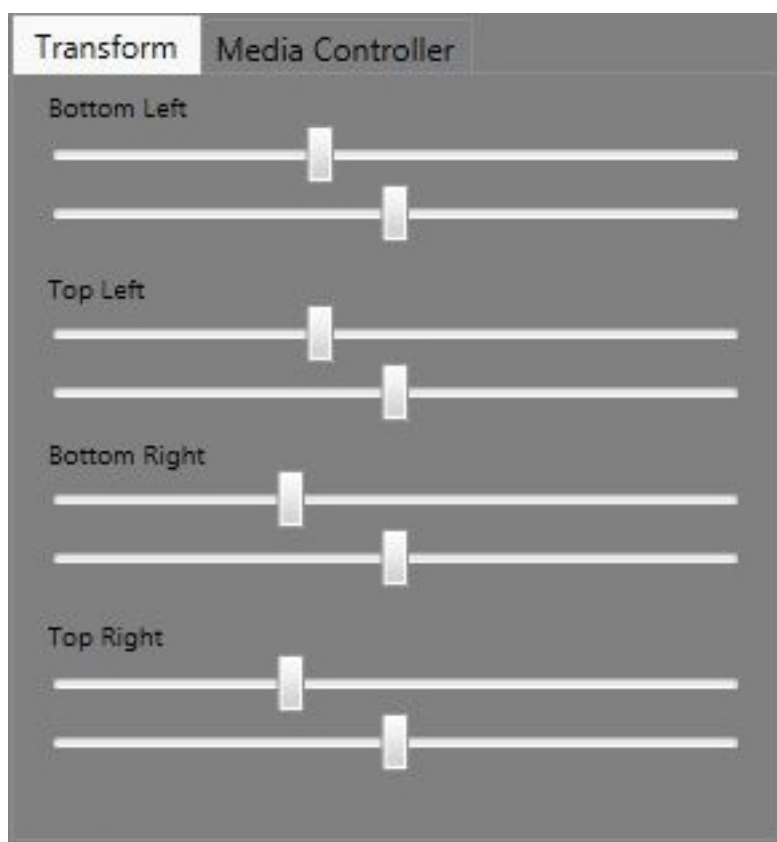
Rys 7.1.2 Pasek Menu View

7.2 Zakładka Transform

W zakładce transform znajduje się 8 suwaków podzielonych na 4 grupy:

- Dolny lewy
- Górny lewy
- Dolny prawy
- Górny prawy

Pierwszy suwak z każdej grupy odpowiada za współrzędną X drugi za współrzędną Y dla danego rogu. Poprzez zmianę położenia każdego z suwaków można zmienić kształt warstwy. Rysunek 7.2 pokazuje zrzut ekranu zakładki



Rys. 7.2 Zakładka transform

7.3 Zakładka Media Controller

Zakładka Media Controller odpowiada za operacje na mediach. Dzieli się na okno podglądu z suwakiem przewijania, przycisk resetowania ustawień oraz 4 sekcje:

- Options

W sekcji Options możemy zaznaczyć opcje zapętlenia, wyciszenia bądź wyłączenia podglądu

- Speed Ratio

W sekcji Speed Ratio za pomocą suwaka możemy zmienić prędkość odtwarzanego klipu wideo.

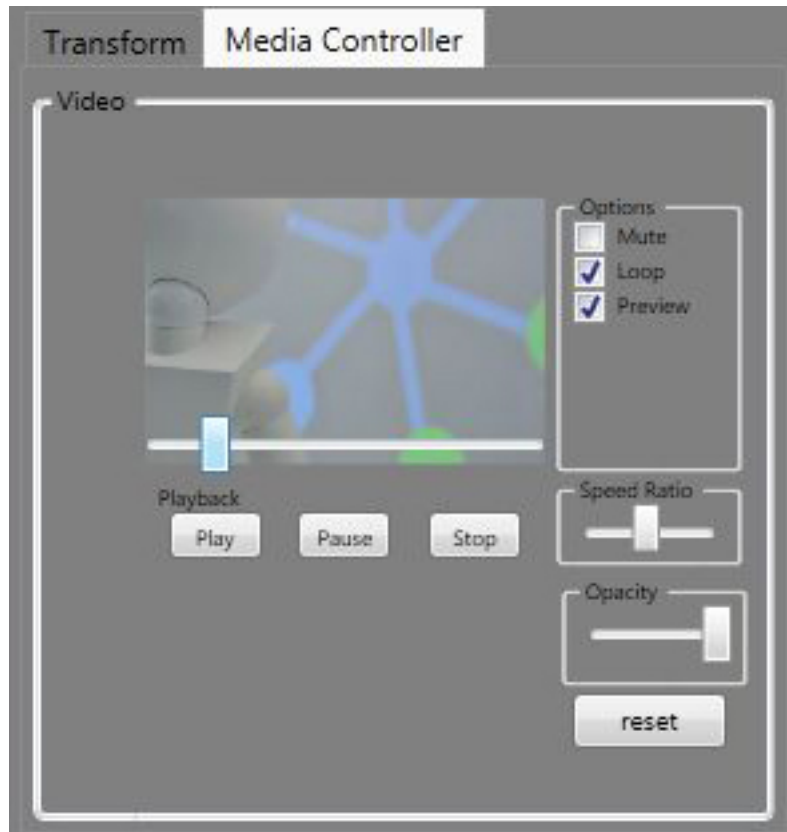
- Opacity

W tej sekcji przy użyciu suwaka możemy zmienić parametr przeźroczystości odtwarzanego klipu wideo.

- Playback

Sekcja Playback posiada przyciski odpowiadające za odtwarzanie klipów wideo tj. odtwarzanie, pauzowanie oraz zatrzymywanie.

Rysunek 7.3 jest zrzutem ekranu zakładki Media Controller

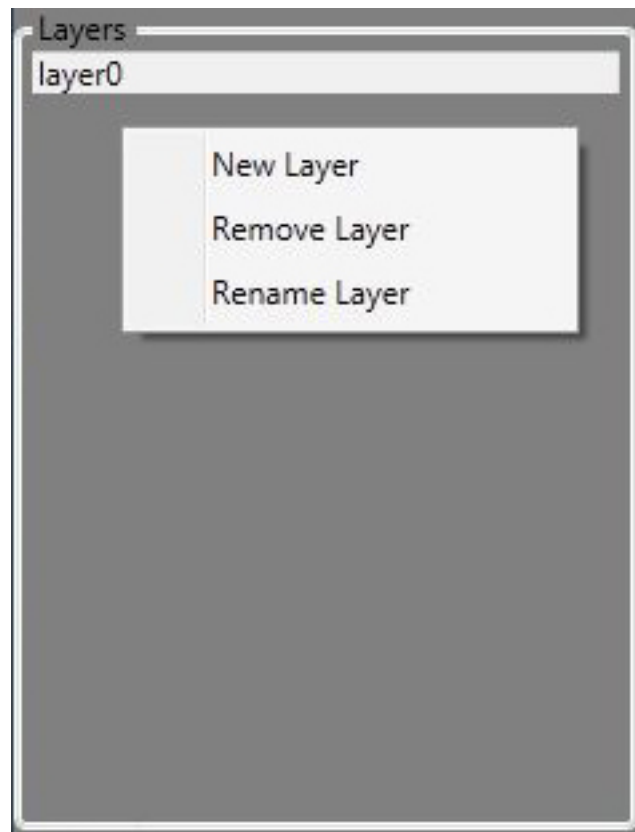


7.3 Zakładka Media Controller

7.4 Lista warstw

W liście Layers przechowywane są nazwy utworzonych warstw poprzez kliknięcie prawym przyciskiem możemy uzyskać dostęp do menu kontekstowego poprzez który uzyskujemy dostęp do funkcji warstw takich jak: Stwórz nową warstwę, usuń warstwę oraz zmień nazwę warstwy.

Rysunek 7.4 jest zrzutem ekranu listy warstw

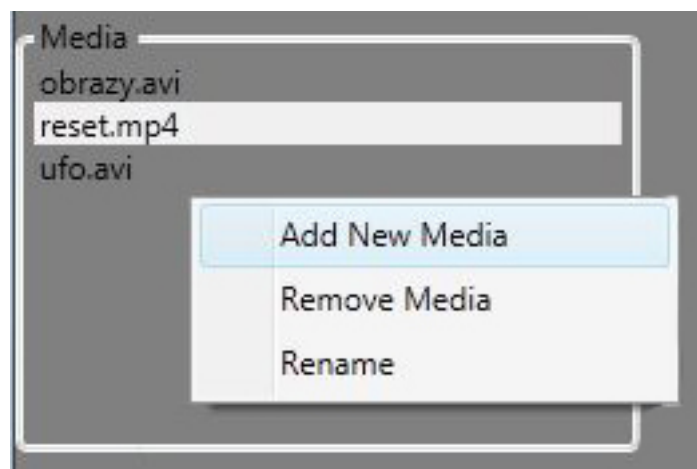


Rys. 7.4 Lista Layers

7.5 Lista mediów

W liście Mediów przechowywane są nazwy zaimportowanych mediów. Lista posiada menu kontekstowe do którego dostęp uzyskać można poprzez kliknięcie prawym przyciskiem myszy. Menu kontekstowe umożliwia importowanie mediów, usunięcie zaznaczonego media oraz zmianę jego nazwy.

Rysunek 7.5 jest zrzutem ekranu listy mediów

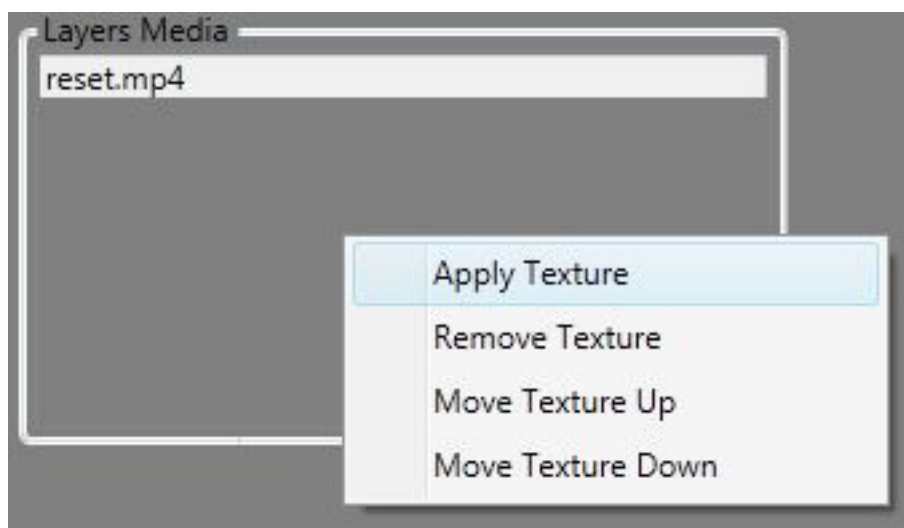


Rys 7.5 Lista Media

7.6 Lista media warstwy

Lista media warstw przechowuje nazwy mediów dodanych do zaznaczonej warstwy. Lista posiada menu kontekstowe dostępne za pomocą prawego kliknięcia myszy, który umożliwia dodanie zaznaczonego media do zaznaczonej warstwy, usunięcie istniejącego media, oraz zmianę kolejności występowania mediów na warstwie poprzez przesuwanie jej w górę bądź w dół.

Rysunek 7.6 jest zrzutem ekranu listy mediów warstw



Rys 7.6 Lista Layer Media

7.7 Podgląd projekcji

Jest podglądem tworzonej projekcji. Wszelkie zmiany dotyczące warstw bądź mediów odtwarzanych na nich będą widoczne w tym podglądzie.

Rysunek 7.7 jest zrzutem ekranu podglądu projekcji



Rys 7.7 Podgląd projekcji

7.8 Okno projekcji

Okno projekcji służy wyświetlaniu obrazu bezpośrednio na obiektach przestrzennych przy użyciu projektora ustawionego w systemie operacyjnym jako rozszerzenie pulpitu. Inną funkcją którą pełni jest zmiana współrzędnych rogów zaznaczonej warstwy. Zmiana współrzędnych odbywa się poprzez 4 kliknięcia myszki odpowiadające rogom w kolejności:

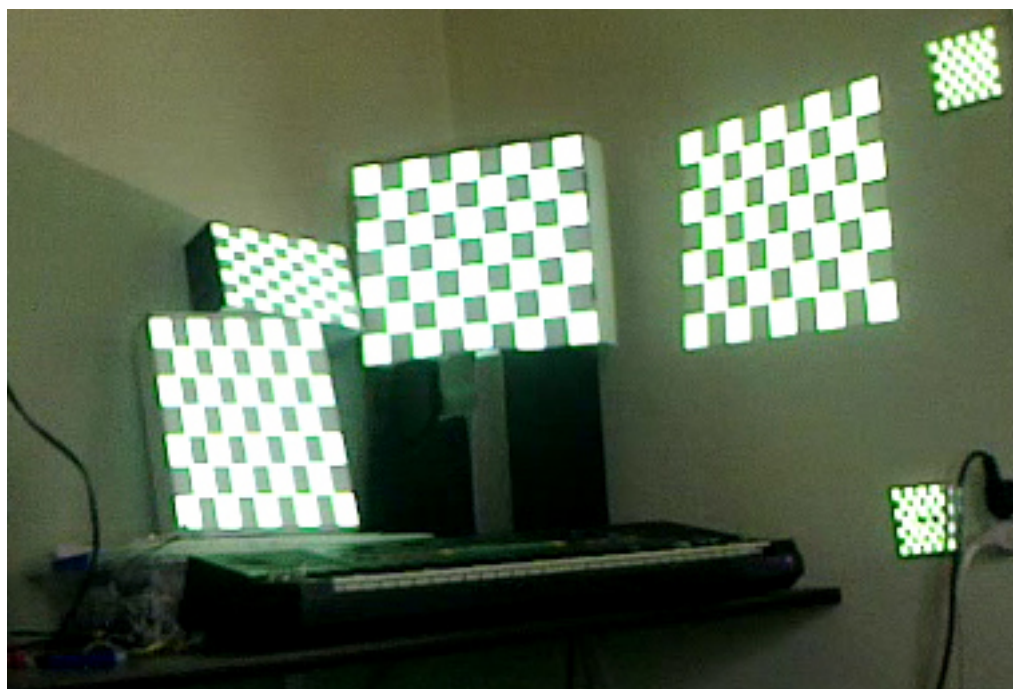
1. Dolny lewy
2. Górny lewy
3. Dolny prawy
4. Górny prawy

Zachowanie kolejności jest istotne, gdyż w wypadku nie zachowania kolejności obraz może być zniekształcony, bądź obrócony o 90 lub 180 stopni.

8 Przykładowe zastosowanie aplikacji

W tym rozdziale przedstawiona jest przykładowa projekcja przestrzenna na 6 płaszczyznach. Do stworzenia tej projekcji został użyty jeden projektor marki Sony o rozdzielczości 800x600 pikseli, oraz lampy o mocy 1300 lumenów. Projektor został podłączony do komputera oraz ustawiony w systemie jako rozszerzenie pulpitu.

Kolejnym krokiem było uruchomienie aplikacji oraz stworzenie 6 warstw, oraz ustawieniem ich w taki sposób aby pokryły miejsca w których miała się odbyć projekcja. Utworzone oraz dostosowane warstwy ilustruje rysunek 8.1



Rys. 8.1 Utworzone warstwy na elementach przestrzennych

Gdy elementy kompozycji zostały pokryte warstwami w taki sposób że wyświetlania na nich siatka nie była zdeformowana do aplikacji zostały zaimportowane 4 klipy wideo. Ostatnim krokiem było dodanie klipów do utworzonych warstw.

Efekt tej projekcji ilustruje rysunek 8.2



Rys. 8.2 Projekcja przestrzenna przy zastosowaniu aplikacji Spatial Video Mapper

9 Wymagania sprzętowe oraz programowe

Aplikacja została stworzona w oparciu o silnik graficzny WPF który jest integralną częścią platformy programistycznej .NET 4.0, więc do poprawnej pracy wymagana jest jej instalacja oraz biblioteki DirectX w wersji 9.0 lub wyższej, wszystkie elementy dostępne są do pobrania ze strony Microsoft'u. Wspierane systemy operacyjne to Windows XP SP2, Windows Vista, Windows 7, Windows Server 2003 oraz Windows Server 2008. Do wykorzystania pełnych możliwości silnika graficznego WPF niezbędna jest karta graficzna posiadająca co najmniej 60MB pamięci wraz ze wsparciem Pixel Shader w wersji co najmniej 2.0. Aby odtwarzać klipy wideo niezbędne jest również posiadanie zestawu odpowiednich kodeków umożliwiających odtwarzanie danych filmów. Niezbędne jest również posiadanie projektora ustawionego w systemie jako rozszerzenie pulpitu, w innym wypadku nie będzie możliwości zobaczenia efektów projekcji przestrzennej.

10 Podsumowanie pracy

W ramach projektu pomyślnie ukończono implementację aplikacji do tworzenia projekcji przestrzennych spełniającej postawione wymagania. Głównym założeniem było stworzenie aplikacji, która umożliwi tworzenie dowolnej ilości płaszczyzn na której będzie mogła odbywać się projekcja. Kolejnym założeniem była możliwość operacji na mediach takich

jak zmiana przeźroczystości czy też parametrów odtwarzania. Interfejs użytkownika miał być prosty w użyciu i jak najbardziej intuicyjny co również udało się osiągnąć.

Głównym problemem zaistniałym podczas implementacji programu była edycja warstw. Użytkownik miał mieć możliwość edycji warstw poprzez zmianę położenia każdego rogu prostokąta. Ponieważ warstwa jest geometrią trójwymiarową składającą się z 4 poligonów tworzących razem prostokątną geometrię zmiana położenia jednego rogu powodowała zmiany tylko na edytowanym poligonie. Efektem tego był obraz przekształcony tylko na jego powierzchni. Rozwiązaniem było przekształcenie całej geometrii przez macierz uwzględniającą to przekształcenie. Dzięki zastosowaniu tej techniki zmiana położenia jednego rogu wpływała na całą geometrię oraz wyświetlany na niej obraz.

Aplikacja zostanie w przyszłości rozwinięta o funkcje umożliwiające obsługę wielu projektorów, dodawania efektów wideo, oraz obsługę aplikacji z zewnętrznymi kontrolerów MIDI.

Bibliografia

- [1] Perry C. S., Core C# i .NET, Wydawnictwo Helion, Gliwice 2006
- [2] Adam Nathan, WPF 4 Unleashed, Wydawnictwo Sams, Indianapolis 2010
- [3] Tom Miller, Managed DirectX 9 Graphics and Game Programming, Wydawnictwo Sams, Indianapolis 2003