

RAPORT Z PRAC DEWELOPERSKICH INTERAKTYWNA MATA DO REHABILITACJI DZIECI

Autor innowacji: Paulina Gembara, Maciej Gorzkowski,
Ewelina Smólkowska, Bartłomiej Burlaga

Nazwa Innowacji: Interaktywna mata do rehabilitacji
dzieci

Inkubator: Samorząd Województwa Wielkopolskiego – Regionalny
Ośrodek Polityki Społecznej w Poznaniu – Lider, partnerzy: Miasto
Poznań, Poznańskie Centrum Superkomputerowo-Sieciowe.

**Innowacja społeczna powstała w ramach realizacji projektu
„Przepis na wielkopolską innowację społeczną -usługi opiekuńcze
dla osób zależnych” współfinansowanego ze środków
Europejskiego Funduszu Społecznego - Program Operacyjny
Wiedza Edukacja Rozwój 2014-2020.**



**Fundusze
Europejskie**
Wiedza Edukacja Rozwój



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz Społeczny



WINS

Raport z prac deweloperskich nad projektem Interaktywna Mata

1. Założenia

Przedmiotem prac był projekt o tytule "Interaktywna Mata", który z założenia miał opierać się o zaawansowany i nowoczesny system informatyczny wykorzystujący najnowsze technologie z dziedziny generowania grafiki komputerowej, detekcji użytkownika oraz mapowania wideo.

2. Plan prac

Projekt miał zostać wykonany w czasie około 200 godzin, na które składać się miało:

1. Spotkania projektowe w celu dokładnego ustalenia funkcji programu
2. Pomoc w wyborze sprzętu (komputer, projektory, czujnik Microsoft Kinect v2)
3. Konsultacje zdalne przy użyciu komunikatorów internetowych
4. Implementacja programu
5. Wdrożenie w miejscu docelowym
6. Przygotowanie dokumentacji

3. Wykonana praca

Ostatecznie praca pochłonęła około 260 godzin. Wykaz wykorzystania czasu prezentuje się następująco:

3.1. Spotkania projektowe

Odbyto około 5 spotkań projektowych w Domu Pomocy Społecznej (DPS) oraz na Politechnice. Czas poświęcony to 7 godzin.

3.2. Wybór sprzętu

Opiniowanie oraz wyszukiwanie potrzebnego sprzętu pochłonęło około 3 godziny.

3.3. Konsultacje zdalne

Dyskusje na tematy okołoprojektowe przeplatały się z czasem implementacji i czas ten zostanie podany wspólnie.

3.4. Implementacja programu

Implementacja pochłonęła około 210 godzin, na które złożyły się następujące etapy:

- Analiza dokumentu opisującego proponowaną funkcjonalność pod kątem wykonywalności
- Wybór technologii
- Przygotowanie środowiska dla projektu oraz jego komponentów
- Raportowanie oraz utrzymywanie infrastruktury około projektowej
- Testowanie eksperymentalnych rozwiązań dla innowacyjnego projektu
- Przyrostowa implementacji funkcji programu
- Modyfikacje programu na podstawie informacji zwrotnych od reszty zespołu
- Testowanie i optymalizacja

3.5. Wdrożenie

Odbyto około 10 wizyt wdrożeniowych, które pokrywały się z pracami implementacyjnymi. Każda wizyta zajmowała około 3 godziny. W ramach wizyt wdrożeniowych wykonywano następujące czynności:

- Wgranie najnowszej wersji systemu na komputer docelowy
- Kalibracja projektorów oraz położenia czujnika Kinect
- Testowanie bez użytkownika końcowego
- Testowanie z użytkowaniem końcowym
- Zbieranie opinii kadry DPS
- Dyskusja na temat funkcji programu

3.6. Przygotowanie dokumentacji

Do dokumentacji zalicza się niniejszy raport, instrukcja instalacji i uruchomienia oraz przygotowanie kopii programu na nośnikach DVD.

Czas poświęcony na te czynności wyniósł 10 godzin.

4. Wykorzystane technologie

Technologie w projekcie należy podzielić na kilka kategorii.

4.1. Projekt główny

4.1.1. Platforma

Platformą dla projektu został **Windows 10 Professional**.

4.1.2. Silnik graficzny

Sama aplikacja wykonana została w oparciu o silnik graficzny **Unreal Engine 4.19** wraz z wtyczkami:

- [VictoryPlugin](#) - wsparcie dynamicznego wczytywania obrazu i dźwięku
- [OceanPlugin](#) - symulacja oceanu
- [ClassicRTSFOW](#) - rysowanie po teksturach
- [Kinect4Unreal](#) - integracja z Kinectem
- [SpoutUE4](#) - integracja z oprogramowaniem do przekazywania obrazu przez kartę graficzną Spout
- [MySQLConnectorUE4Plugin](#) - integracja z bazą danych MySQL

Wykorzystano także zasoby z paczek:

- [ReefKit](#)
- [Underwater Life](#)

4.1.3. Baza danych

Zanonimizowane dane użytkowników przechowywane są w bazie danych **MySQL Server Community 5.7.22** zarządzanej przez **MySQL Workbench 6.3.10**.

4.1.4. Wideo mapowanie

Do wyświetlenia obrazu z projektorów wykorzystano system mapowania wideo **Resolume Arena 6**.

4.1.5. Strumieniowanie obrazu

Obraz z silnika graficznego do systemu mapowania przekazywany jest przy użyciu oprogramowania [Spout2](#).

4.1.6. Detekcja użytkownika

Ruch użytkownika śledzi **Microsoft Kinect v2** wykorzystujący oprogramowanie producenta.

4.2. Zarządzanie projektem

4.2.1. Repozytorium

Projekt przechowywany jest w serwisie [GitHub](#) w prywatnym repozytorium.

4.2.2. Narzędzie kontroli wersji

Do kontroli wersji wykorzystano narzędzie **Subversion (SVN)** wraz z nakładką systemową **TortoiseSVN**.

4.2.3. Śledzenie i planowanie zmian

Do śledzenia przebiegu prac oraz raportowania problemów wykorzystano narzędzie [JIRA](#).

4.2.4. Tworzenie dokumentacji

Wszelkie dokumentacje wewnątrz projektowe wykonane zostały przy użyciu zestawu narzędzi **Google Docs**.

4.2.5. Narzędzia komunikacji zespołu

Do komunikacji zespół wykorzystywał oprogramowanie **Slack**.

5. Uwagi

5.1. Przekroczony budżet godzin

Planowany budżet godzin przekroczony został o 30%, które to pokrył programista z własnych środków. Na niedoszacowanie przełożyło się kilka czynników:

- Liczność modułów aplikacji - skala projektu nieznacznie wykraczała poza możliwości budżetowe jego realizacji
- Innowacyjność - innowacyjny charakter aplikacji sprawia, że dokładne oszacowanie czasu jest o wiele trudniejsze niż w przypadku realizacji standardowych projektów
- Złożoność systemu - wykorzystanie wideo mapowania oraz czujnika Kinect utrudnia znacząco testowanie programu, szczególnie w trakcie procesu twórczego w warunkach gdzie sprzęt ten nie jest dostępny
- Specyfika użytkownika końcowego - z powodu utrudnionej komunikacji z użytkownikiem o wysokim poziomie niepełnosprawności analiza scenariuszy użycia jest kłopotliwa

5.2. Brak programu w wersji wykonywalnej

Wtyczka **Kinect4Unreal**, który to służy do obsługi **Kinecta** z poziomu silnika graficznego **Unreal** i jest absolutnie kluczowym w niniejszym systemie posiada politykę nie udostępniania źródeł. Ze względu na specyfikę silnika **Unreal** nie da się zbudować wersji wykonywalnej bez posiadania źródeł wtyczek. Próba kontaktu z twórcami była bezskuteczna. Ewentualne wykupienie płatnej licencji (**1500\$**), bądź wykorzystanie ich pomocy do każdorazowego zbudowania wersji wykonywalnej (**400\$**) wykraczała dalece poza możliwości finansowe projektu, ale należy wliczyć je w przypadku chęci rozwijania projektu.

Aktualnie program wymaga zainstalowanego środowiska **Unreal Engine**, aby mógł się uruchomić. Poza tym nie ma praktycznej różnicy względem wersji wykonywalnej. Ponieważ

źródła są wymagane do uruchomienia aplikacji niemożliwe jest dystrybuowanie jej w wersji zbinaryzowanej.

5.3. Kłopotliwa integracja silnika Unreal i systemu mapowania wideo Resolume Arena 6 przy użyciu narzędzia strumieniowania tekstur Spout

Powyższe rozwiązanie dostępne jest od stosunkowo niedawna po stronie narzędzi do mapowania wideo. Natomiast po stronie **Unreal Engine** nie ma oficjalnego wsparcia dla **Spout**, a wtyczki to umożliwiające tworzone są przez społeczność, natomiast ich jakość pozostawia wiele do życzenia. Autor aplikacji przerobił wtyczkę, aby funkcjonowała zgodnie z wymaganiami, jednak pojawiają się problemy, które kończą się zawieszeniem programu. Naprawienie tych błędów i dalszy rozwój wtyczki wykraczają poza możliwości finansowe projektu. W przypadku rozwijania projektu należy wziąć pod uwagę potrzebę dalszego rozwinięcia wtyczki lub zmianę koncepcji komunikacji silnika graficznego z mapowaniem wideo.