Oprogramowanie

1. Informacje ogólne

Kluczowym punktem pracy było stworzenie oprogramowania umożliwiającego odtwarzanie nieskompresowanych sekwencji wideo oraz przeprowadzenie testów subiektywnych z użyciem autorskiego odtwarzacza.

1. Wybór narzędzi

Wybór narzędzi wiązał się z koniecznością przeglądu dostępnych zasobów ludzkich pod kątem umiejętności tworzenia oprogramowania w danym języku oraz analizą istniejących rozwiązań w celu znalezienia jak najlepszego narzędzia do rozwiązania problemu.

Odtwarzacz nieskompresowanych sekwencji wideo musi charakteryzować się jak największą wydajnością. Poprzez wydajność rozumiane jest jak najlepsze zarządzanie zasobami w taki sposób, aby na dostępnym sprzęcie komputerowym uzyskać jak najlepsze parametry wyświetlania kolejnych klatek. Zdecydowano się zrezygnować z języków korzystających z maszyn wirtualnych na korzyść takich, które pozwalają na dużą swobodę w zarządzaniu pamięcią. Wybór padł na język C++ ze względu na doświadczenie z tym językiem, zarówno zawodowe jak i nabyte podczas studiów.

Wybranym systemem operacyjnym został Linux, dystrybucja Xubuntu 16.04.2 posiadająca jądro w wersji 4.4.0-72-generic. Decydującymi czynnikami były łatwość instalowania kolejnych pakietów bibliotek, dostęp do narzędzi konwertujących parametry filmów oraz niskie zużycie zasobów sprzętowych przez biernie działający system.

[https://xubuntu.org/]

Zdecydowano się użyć środowiska programistycznego CLion dostarczanym przez firmę JetBrains, korzystając z licencji studenckiej, która pozwala na użycie IDE (ang. *Integrated Development Environment*) w celach edukacyjnych.

[https://www.jetbrains.com/]

Analiza istniejących odtwarzaczy wideo pozwoliła wyselekcjonować bibliotekę, która dała by możliwość darmowego użycia i jak największej możliwości modyfikacji, czyli posiadała licencję *open source*. Ze względu na popularność odtwarzacza VLC w systemach operacyjnych Linux, podjęto decyzję o wykorzystaniu biblioteki z użyciem której powyższy odtwarzacz został stworzony – libVLC.

[http://www.videolan.org/vlc/libvlc.html]

FFmpeg, czyli narzędzie pozwalające na edycje parametrów wzorcowego wideo zostało wybrane ze względu na łatwość użycia, swobodę w wyborze zmienianych aspektów filmu oraz licencję *open source.* Wszystkie użyte w pracy sekwencje wideo zostały wygenerowane przy pomocy FFmpeg z filmów wzorcowych.

[https://ffmpeg.org/]

1. Proces tworzenia

Oprogramowanie przedstawione w pracy tworzone było w zespole dwuosobowym. Organizacja pracy w grupie jest zależna od ilości jej członków, natury rozwiązywanego problemu oraz wielu innych czynników np. ograniczeń czasowych lub sposobu prezentacji postępów.

Obecnie w informatyce dąży się do wdrożenia tak zwanych metod zwinnych programowania, pozwalają one zminimalizować straty w przypadku, gdy część wymogów klienta ulegnie zmianie. Wynika to z faktu, że kod źródłowy programu dostarczany jest iteracyjnie w jak najmniejszych częściach. Podejście to pozwala również zmaksymalizować czas faktycznej pracy każdego z pracowników, ponieważ eliminowane są sytuacje, w których pojawia się konieczność oczekiwania na wartość dostarczaną przez innych pracowników.

Jednym z pierwszych kroków podczas realizacji pracy magisterskiej było zdefiniowanie zasad współpracy między opiekunem, uczelnią a osobami odpowiedzialnymi za nią samą. W związku z koniecznością dotrzymania terminów odgórnie narzuconych przez uczelnie zdecydowano się wyznaczyć kilka terminów wraz z zakresem funkcjonalności, które na dany termin miały by być dostarczone. Pozwoliło to zsynchronizować wiedzę o projekcie między studentami, a prowadzącym oraz skupienie się na wyznaczonych celach. Dopiero po akceptacji postępów przez opiekuna następowało przejście do następnego etapu. Na każdym ze spotkań projektowych przeprowadzano pokaz aktualnej wersji, na którym omawiano aktualny sposób działania, najnowsze zmiany w funkcjonowaniu programu oraz znajdywano ewentualne niedociągnięcia.

W czasie realizacji pracy magisterskiej obaj członkowie zespołu pracowali zawodowo oraz uczęszczali na zajęcia prowadzone na uczelni. W związku z tym proces tworzenia pracy magisterskiej musiał być dostosowany do ich obłożenia czasowego. Naturalnym wyborem była więc praca w metodyce *Kanban* – skupiającej się na produkcie, bez dodatkowych, niepotrzebnych nakładów zasobów oraz minimalizacji bezczynności. Stworzono prostą *tablicę kanbanową* czyli złożenie danych o postępie prac nad projektem przy podziale na poszczególne jego zadania. Zaletą takiego rozwiązania jest wiedza o przewidywanym terminie zakończenia prac oraz dowolność w wyborze czasu, w którym są one realizowane (oprócz terminów końcowych). Takie podejście pozwoliło powiązać pracę zawodową z rozwojem naukowym.

System kontroli wersji zastosowany w projekcie pozwolił na równoczesną pracę nad wieloma funkcjonalnościami programu, wprowadził historię zmian oraz pozwolił na bezproblemową ich synchronizację między poszczególnymi członkami zespołu. Wykorzystano popularną platformę GitHub, która dostarcza gotowego rozwiązania wraz z serwerem do przechowywania danych.

[https://github.com/Kondix/UHDPlayer]

1. Analiza systemu

Analiza systemu została przeprowadzona przed implementacją rozwiązania. Podczas tej fazy stworzono wymagania, które będą stanowiły o tym kiedy projekt można uznać za zakończony. Ich definicja była konieczna, aby ukierunkować tok prac.

1. Szczegóły implementacyjne