



Dokumentacja Projektu grupowego

Raport końcowy

Wydział Elektroniki, Telekomunikacji i Informatyki

Politechnika Gdańska

{wersja dokumentu wzorcowego: wersja 2/2023}

Nazwa i akronim projektu: {nazwa projektu, np: System zabezpieczenia portu przed zagrożeniami terrorystycznymi - SZP} Aplikacja wizualizująca zagadnienia związane z kodowaniem nadmiarowym	Zleceniodawca: {nazwa/nazwisko klienta} dr inż. Bartosz Czaplewski	
Numer zlecenia: {numer zespołu projektowego w ramach Projektu grupowego wg systemu SPG, np. 13@KSSR'2022} 5@KSTI'2023/24	Kierownik projektu: {kierownik zespołu projektowego} Bartosz Kołakowski	Opiekun projektu: {opiekun projektu} dr inż. Bartosz Czaplewski

Nazwa / kod dokumentu: Raport końcowy – RK	Nr wersji: {wersja dokumentu np. 1.00} 2.00	
	Data pierwszego sporządzenia: {data wykonania pierwszej wersji dokumentu} 29.05.2024	
	Data ostatniej aktualizacji: {data wykonania aktualnej wersji dokumentu} 09.06.2024	
	Semestr realizacji Projektu grupowego: 2 {nie zmieniać}	

Historia dokumentu				
Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.00	Wstępna wersja	całość	Jastrzębski Paweł	29.05.2024
2.00	Uzupełnienie załączników	pkt 3	Jastrzębski Paweł	09.06.2024

Spis treści

1	Wprowadzenie - o dokumencie	3
1.1	Cel dokumentu	3
1.2	Zakres dokumentu	3
1.3	Odbiorcy	3
1.4	Terminologia	3
2	Rezultaty projektu	3
2.1	Wprowadzenie – opis ogólny projektu	3
2.2	Cel projektu i planowany zakres realizacji	3
2.3	Faktyczny zakres realizacji projektu i rozbieżności oraz zakres wykonanych prac	4
2.4	Osiągnięte wyniki	5
2.5	Charakterystyka pracy zespołowej	5
3	Załączniki	6

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

{nie zmieniać }

Celem dokumentu jest zebranie istotnych informacji dotyczących całości zrealizowanego projektu w jednym miejscu i zaprezentowanie ich w przejrzysty sposób. Dokument ma na celu przede wszystkim ułatwienie oceny projektu, w szczególności w przypadku zgłoszenia go do nagrody lub wyróżnienia. Przewidywana objętość dokumentu 3-6 stron (części merytorycznej).

Należy również wskazać wykonane prace z podaniem ich krótkiej charakterystyki, wskazać rozbieżności wykonywanych prac w stosunku do planowanych, podsumować prace z wykazaniem pracy zespołowej oraz wyspecyfikować listę dokumentów wytworzonych w projekcie (wersji końcowych – lista obejmuje dokumenty z obu semestrów w wersjach ostatecznych), które zostały umieszczone i zatwierdzone przez opiekuna w serwisie SPG.

1.2 Zakres dokumentu

{określenie, co wchodzi w zakres dokumentu, a co nie wchodzi, ew. wskazanie na dokumenty powiązane}

W dokumencie są opisane nasze rezultaty tworzenia projektu i zmiany z pierwotnym planem.

Dodatkowo załączniki są opisane w punkcie 3.

1.3 Odbiorcy

{określenie adresatów dokumentu, może być to typ odbiorcy; tu: zleceniobiorca (Katedra), członkowie zespołu projektowego oraz wymienione z nazwiska osoby, do których dokument ma dotrzeć}

Zleceniobiorca i klient (dr inż. Bartosz Czaplewski) oraz członkowie zespołu projektowego; Bartosz Kołakowski, Michał Mróz, Paweł Jastrzębski, Maksym Nowak, Piotr Noga

1.4 Terminologia

{wyjaśnienie używanych w dokumencie pojęć i skrótów, oznaczenia używane wewnątrz dokumentu np. oznaczenia wymagań}

Skróty osób w tabelce:

Bartosz Kołakowski - BK

Michał Mróz - MM

Paweł Jastrzębski - PJ

Maksym Nowak - MN

Piotr Noga – PN

2 Rezultaty projektu

2.1 Wprowadzenie – opis ogólny projektu

{W projektach, w których realizowana jest część większego systemu lub urządzenia, wprowadzenie musi zawierać opis całego systemu lub urządzenia w docelowym kształcie oraz wyraźne wyjaśnienie, jakie części tego systemu lub urządzenia były dostępne w momencie rozpoczęcia projektu, a jakie zespół projektowy zrealizował (lub miał zrealizować) samodzielnie. Wprowadzenie musi zawierać ponadto opis części systemu lub urządzeń przewidzianych do realizacji w ramach projektu. Dla projektów stanowiących zamkniętą całość (urządzenie, układ, aplikacja) wprowadzenie musi zawierać ogólną charakterystykę tworzonego rozwiązania (urządzenia, układu, aplikacji) i jego środowiska pracy.}

Projekt polegał na stworzeniu aplikacji komputerowej, która ma na celu wizualizację działania kodów nadmiarowych. Przedstawione kody to kod Hamminga oraz kod Reeda-Solomona. Aplikacja stanowi całość, została zrealizowana od zera. Przede wszystkim musi być dostępna na systemie Windows. Nie był wymagany konkretny język programowania, zdecydowaliśmy się użyć C++.

Aplikacja miała pokazać proces kodowania wiadomości oraz dekodowania z pozwoleniem na zmianę zakodowanej wiadomości w celu symulacji błędów transmisji. Ostatecznym produktem jest plik wykonywalny w Windowsie (.exe).

2.2 Cel projektu i planowany zakres realizacji

{sformułowany cel projektu, ewentualnie modyfikacje celu w trakcie realizacji, przedstawienie początkowych wymagań projektowych postawionych w I i II semestrze, omówienie planowanego sposobu realizacji}

Celem projektu było napisanie narzędzia wspomagającego proces dydaktyczny. Narzędziem tym jest aplikacja wizualizująca zagadnienia związane z kodowaniem nadmiarowym.

Wymagana była wizualizacja 2 kodów – Hamminga w pierwszym semestrze, Reeda-Solomona w drugim. Aplikacja musiała też być dostępna na Windowsie.

Wybierając język programowania zastanawialiśmy się nad C#, Javą i C++. Część osób była zainteresowana korzystaniem z C#, część chciała korzystać z Javy, ponieważ znała ją dobrze i była wykorzystywana w ich pracy, ale po dyskusjach doszliśmy do wniosku, że najlepiej będzie korzystać z C++, gdyż jest to język, który jest wszystkim najbardziej znany.

Jeśli o tworzenie interfejsu graficznego to mając do wyboru Qt, FLTK, GTK i wxWidgets. Po zapoznawaniu się z opiniami i ilością pomocy do danych frameworków zdecydowaliśmy o wybraniu Qt.

Planowany sposób realizacji – język C++, dzielimy się zadaniami z uwzględnieniem naszych silnych i słabych stron, kod dzielony za pomocą repozytorium na GitHubie, do którego każdy z członków ma dostęp.

Planowana lista zadań:

- stworzenie bazy projektu – zainicjowanie repozytorium, pusty projekt Qt
- stworzenie bazy GUI
- implementacja kodu Hamminga
- implementacja kodu Reeda-Solomona
- wizualizacja kodu Hamminga
- wizualizacja kodu Reeda-Solomona
- napisanie testów do kodu Hamminga
- napisanie testów do kodu Reeda-Solomona

2.3 Faktyczny zakres realizacji projektu i rozbieżności oraz zakres wykonanych prac

{przedstawienie procesu realizacji projektu: środowisko sprzętowe i/lub programistyczne wykorzystane w realizacji projektu, rzeczowy opis głównych prac wykonanych w I i II semestrze, z podaniem krótkiej ich charakterystyki; może być tabela, wskazanie rozbieżności w stosunku do planu z punktu 2.2, wykorzystywane dodatkowe narzędzia, takie jak biblioteki, API, zestawy uruchomieniowe czy gotowe moduły, układy elektroniczne itp. (w każdym przypadku należy wskazać, które elementy wytworzono w całości samodzielnie, a które były dostępne z góry); opis musi uwzględniać uszczegółowienie wymagań projektowych i ich modyfikacje wprowadzane w trakcie realizacji projektu.}

Aplikacja została stworzona w języku C++, we frameworku Qt. Używane są pliki .cpp, w których zawarta jest implementacja kodów z ich testami oraz pliki .qml, które opisują wygląd interfejsu graficznego. Udało się spełnić wszystkie wymagania i wykonać wszystkie planowane zadania.

Wykonane zadania:

Nazwa	Opis	Dodatkowe narzędzia
Stworzenie interfejsu graficznego	Bazowy wygląd aplikacji – menu główne, wybór kodowania, wpisanie wiadomości, wybór języka	Stworzone samodzielnie z wykorzystaniem frameworka Qt (pliki .qml)
Implementacja kodu Hamminga	Implementacja kodowania i odkodowywania zgodnie z algorytmem Hamminga	Stworzone samodzielnie w C++
Testy kodu Hamminga	Testy automatyczne poprawności implementacji – sprawdzające, czy wiadomości są dobrze kodowane/odkodowywane	Stworzone samodzielnie z wykorzystaniem framework do testów – Google Test oraz biblioteki boost
Wizualizacja kodu Hamminga	Ukazanie wykonywanych operacji krok po kroku, możliwość zmiany zakodowanej wiadomości przed dekodowaniem. Komunikacja między implementacją oraz interfejsem graficznym	Stworzone samodzielnie z wykorzystaniem frameworka Qt (pliki .qml, gotowy mechanizm komunikacji między .cpp i .qml)
Implementacja kodu Reeda-Solomona	Implementacja kodowania i odkodowywania zgodnie z algorytmem Reeda-Solomona	Stworzone samodzielnie w C++
Testy kodu Reeda-Solomona	Testy automatyczne poprawności implementacji – sprawdzające, czy wiadomości są dobrze	Stworzone samodzielnie z wykorzystaniem framework do testów – Google Test oraz

	kodowane/odkodowywane	biblioteki boost
Wizualizacja kodu Reeda-Solomona	Ukazanie wykonywanych operacji krok po kroku, możliwość zmiany zakodowanej wiadomości przed dekodowaniem. Komunikacja między implementacją oraz interfejsem graficznym	Stworzone samodzielne z wykorzystaniem frameworka Qt (pliki .qml, gotowy mechanizm komunikacji między .cpp i .qml)
Tłumaczenie	Zapis wybranego języka do pliku konfiguracyjnego. Tłumaczenie interfejsu graficznego na podstawie plików z tłumaczeniami.	Samodzielnie stworzony system odczytu tłumaczeń z plików, samodzielne tłumaczenia. Plik konfiguracyjny tworzony automatycznie przez Qt

Początkowo implementacja, wizualizacja i testowanie kodu Reeda-Solomona miało być wykonane w semestrze pierwszym, ale ze względu na skupienie się na ulepszaniu funkcjonalności i graficznego interfejsu użytkownika w programie. Dodatkowo w trakcie pracy nad programem wpadliśmy na pomysł by wprowadzić dwie wersje językowe.

2.4 Osiągnięte wyniki

{opis uzyskanego wyniku końcowego; krótka charakterystyka funkcji/parametrów technicznych; może, ale nie musi, zawierać propozycje ulepszeń, kierunki dalszych prac.}

W menu głównym aplikacji najpierw można wybrać jeden z kodów (pokazany jest też opis merytoryczny kodowania), następnie można wpisać wiadomość, która zostanie zakodowana. Ukazany jest proces kodowania wiadomości, w trakcie którego można dostosowywać prędkość obliczeń. Zakodowaną wiadomość można potem zmienić i wykonać dekodowanie – znowu zaangażowane krok po kroku. Na końcu jest ukazana, jaką wiadomość udało się zdekodować.

Kod Hamminga obsługuje wpisanie wiadomości o dowolnej długości (automatycznie jest dobierana wymagana liczba bitów parzystości). Kod Reeda-Solomona wymaga podania 5 bloków danych 3-bitowych (liczby 0-7), tworzone są 2 dodatkowe bloki.

Potencjalne kierunki dalszych prac to przede wszystkim by mogła być implementacja kolejnych kodowań. Potencjalnie można by też dodać tłumaczenia na nowe języki.

2.5 Charakterystyka pracy zespołowej

{tu należy rzeczowo wskazać, kto co wykonywał, w jaki sposób kierowano projektem, jak przydzielano zadania, wskazać wkład poszczególnych osób w proces tworzenia produktu, dokumentowania, obsługi systemu SPG, współpracy z klientem, opiekunem itp.; należy wykazać jak realizowano pracę zespołową; może być tabela}

Zadanie	BK	MM	PJ	MN	PN	Rozpoczęcie	Zakończenie
stworzenie bazy projektu		X				31.10.2023	5.11.2023
stworzenie bazy GUI		X			X	31.10.2023	15.11.2023
implementacja kodu Hamminga		X				03.11.2023	01.12.2024
implementacja kodu Reeda-Solomona			X			21.02.2024	30.04.2024
wizualizacja kodu Hamminga		X	X		X	05.11.2023	10.01.2024
wizualizacja kodu Reeda-Solomona	X		X			01.03.2024	31.05.2024
napisanie testów do kodu Hamminga		X				05.11.2023	22.11.2024
napisanie testów do kodu Reeda-Solomona	X	X				01.03.2024	31.05.2024
napisanie tłumaczenia na język angielski					X	01.04.2024	31.05.2024

Dokumentacja	BK	MM	PJ	MN	PN
Harmonogram	X		X		
Plakat	X				
Prezentacja	X				
Raport semestralny	X				
Raport końcowy	X		X		
Dokumentacja techniczna	X		X		X

Przed przejściem do realizacji zadań zaplanowaliśmy pracę grupową tak, żeby poszczególne zadania były przydzielone zgodnie z możliwościami danych osób, ale również z uwzględnieniem tego, żeby praca była rozłożona w miarę równomiernie. Ułożenie harmonogramu zadań było takie, żeby większość zadań realizować na początkach semestrów, a nie na końcu, gdyż wtedy jest okres nauki do sesji i ciężko byłoby nam wtedy realizować projekt. Celem było nieodkładanie pracy na ostatnią chwilę.

Współdzielenie kodu było zrealizowane za pomocą platformy GitHub. Pull request musiał być zaakceptowany przez przynajmniej 1 osobę z grupy (poza autorem), co pozwoliło zapewnić dobre standardy kodu.

Sprawną pracę grupową była też możliwa dzięki regularnym spotkaniom w celu pracy nad projektem (motywowały nas do tego wykłady z projektu grupowego) i szybkim sposobem komunikacji (przez Messengera oraz na uczelni – jak mieliśmy inne zajęcia).

Dodatkowo dzięki regularnym spotkaniom z opiekunem projektu (co 2-3 tygodnie) i wysiłkom w celu regularnego pisania dokumentacji łatwiej było nam przygotować projekt i prezentację na seminarium projektowe.

Na spotkaniach z opiekunem na ogół pojawiał się cały zespół, chyba że któraś z osób akurat nie miała w danym terminie czasu. W trakcie tych spotkań, opiekun przekazał nam dużo cennych uwag i możliwych ulepszeń naszego programu, czym staraliśmy się zająć w pierwszej kolejności.

System SPG był obsługiwany przez kierownika projektu – pilnował on terminów wgrania dokumentów, przygotowywał lub sprawdzał każdy z dokumentów i wgrywał go do systemu. Po wgraniu dokumentu, dawał opiekunowi znać (o co poprosił nas opiekun – ze względu na brak powiadomień po wgraniu)

3 Załączniki

{dotyczy wszystkich dokumentów z SPG - pełna lista aktualnych-końcowych wersji dokumentów, wypracowanych w trakcie realizacji projektu przez 2 semestry, wraz z wyszczególnioną aktualną wersją raportu końcowego; w szczególności:
- umieszczamy tutaj listę dokumentów z 1 semestru (zgodną z treścią raportu semestralnego), ale w najnowszych wersjach (jeśli podlegały korekcie, rozszerzeniu itp. i oznaczono je nowymi numerami wersji)
- umieszczamy tutaj listę dokumentów z 2 semestru w najnowszych wersjach, w tym również wpisujemy raport końcowy}

Tabela. 3.1. Specyfikacja opracowanych dokumentów

L.p.	Nazwa dokumentu	Nazwa pliku umieszczonego w SPG
1	Semestr 2 - Harmonogram szczegółowy	HarmonogramSzczegolowySemestr2.doc
2	Semestr 2 - Plakat informacyjny	PlakatSemestr2.doc
3	Semestr 2 - Dokumentacja techniczna produktu	PG_WETI_DTP.docx
4	Semestr 2 - Prezentacja wyników	PrezentacjaWynikowProjektGrupowySemestr2.pptx
5	Semestr 2 - Raport końcowy	PG_WETI_RK.doc