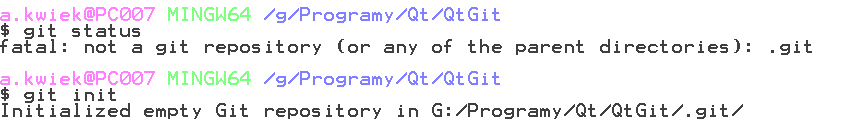
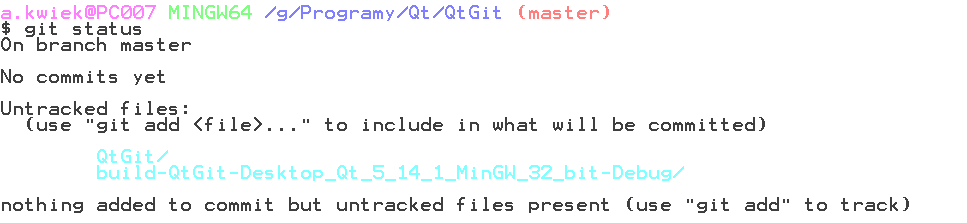
* Utworzyć i zbudować nowy projekt Qt - QDialog

**INICJOWANIE REPOZYTORIUM**

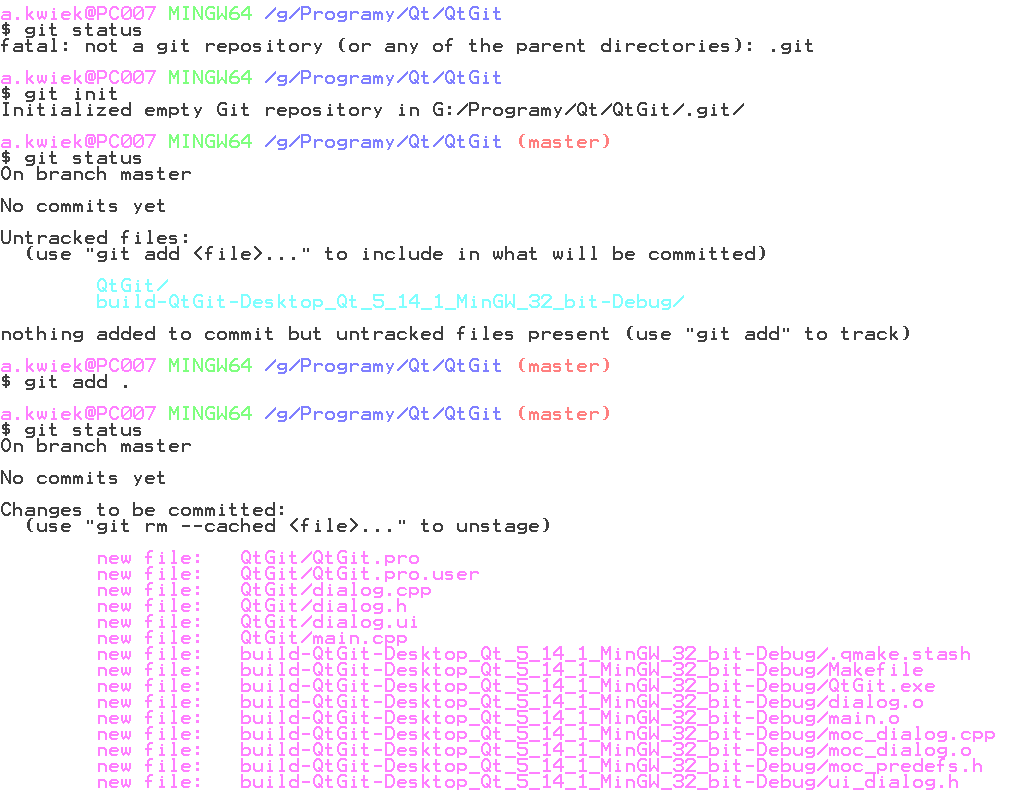
* Zainicjowanie repozytorium
* Pierwszy commit Initial (status, init, log, zawartość katalogu)
* Inicjalizacja repozytorium



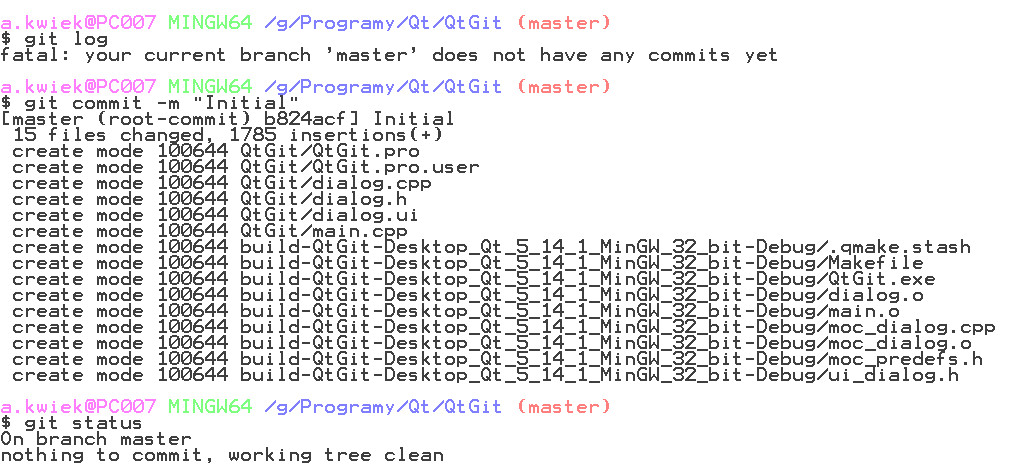
* Sprawdzenie statusu



* Przygotowanie commitu (dodanie plików)

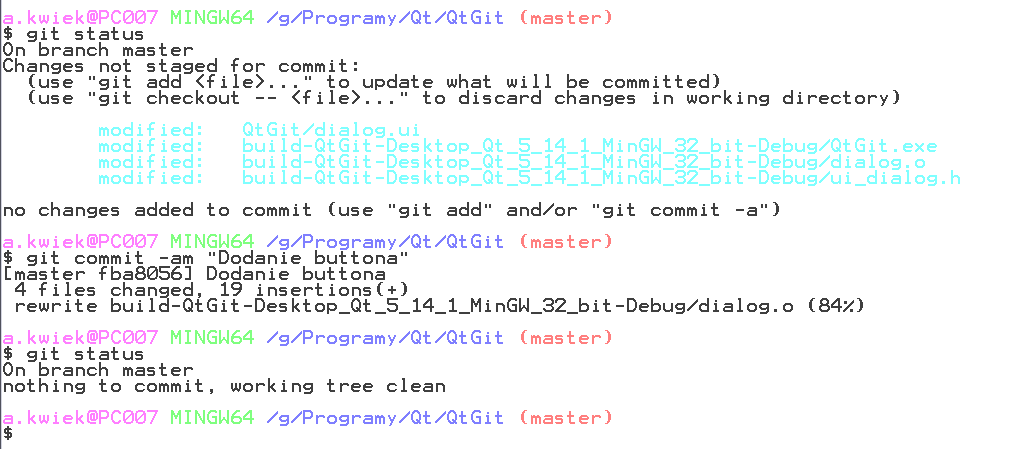


* Commit

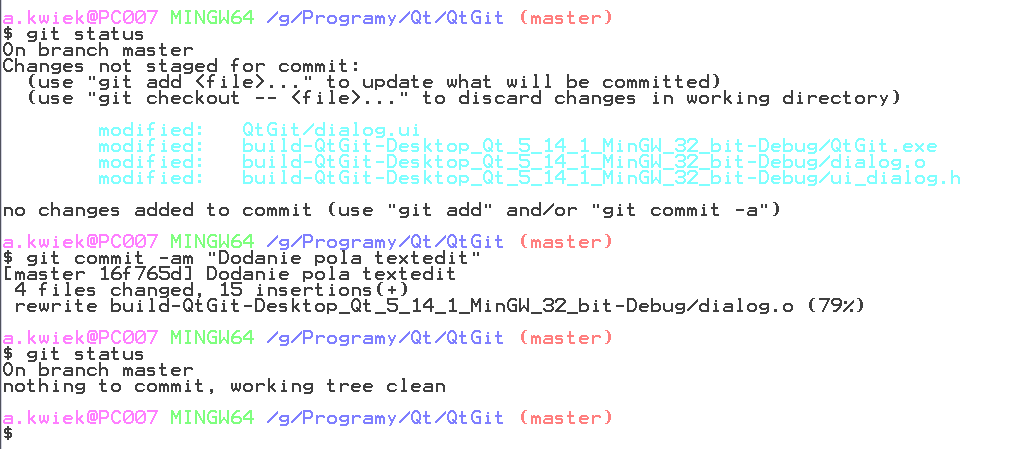


**ZAPISYWANIE ZMIAN**

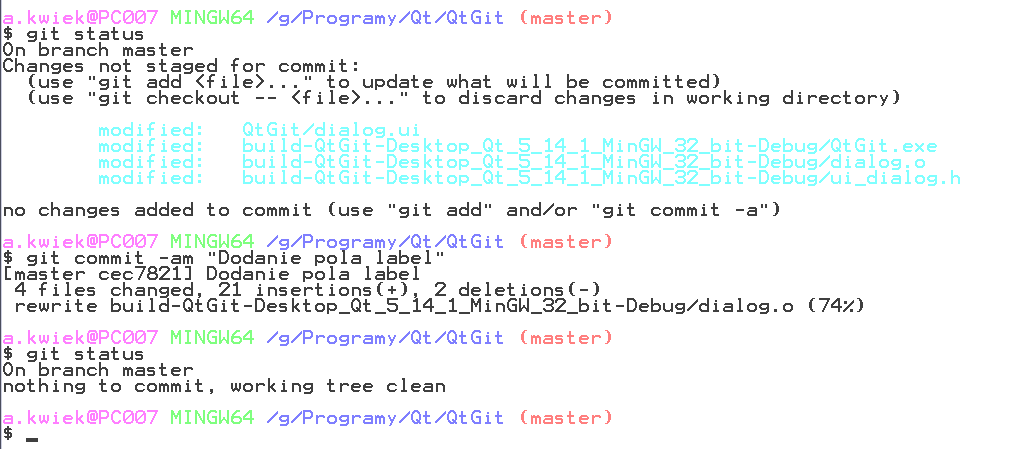
* Dodanie QButtona (status, log, zawartość katalogu)
* Commit (status, log, zawartość katalogu)



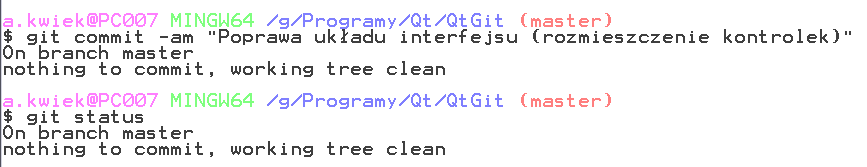
* Dedanie pola edycji QTextEdit (status, log, zawartość katalogu)
* Commit (status, log, zawartość katalogu)



* Dodanie etykiety QLabel (status, log, zawartość katalogu)
* Commit (status, log, zawartość katalogu)



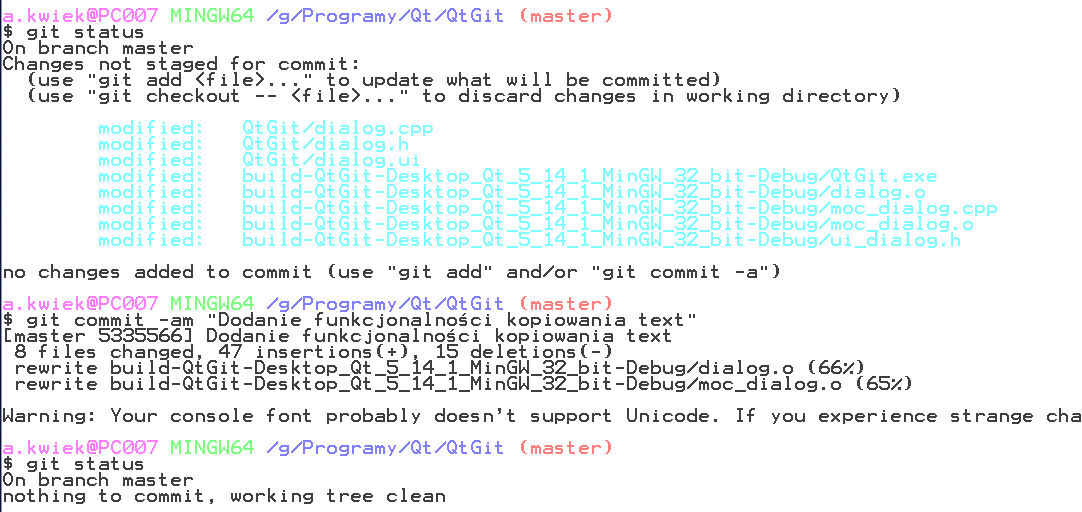
* Poprawa układu interfejsu



* Dodanie funkcjonalności:

- Po wpisaniu tekstu w textEdit i naciśnięciu buttona, tekst ląduje na etykiecie. (status, log, zawartość katalogu)

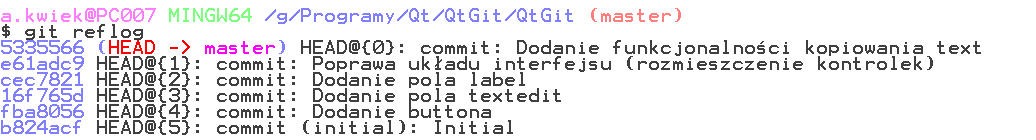
* Commit (status, log, zawartość katalogu)

****

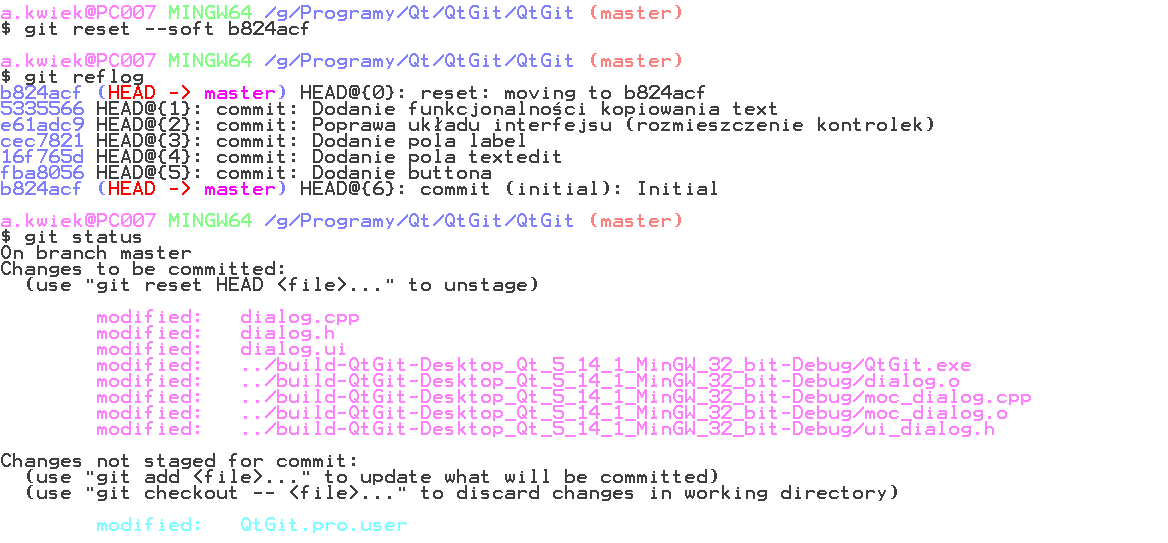
**COFANIE ZMIAN**

<https://cezarywalenciuk.pl/blog/programing/triki-z-git--reset-restore-revert-czyli-cofanie>

* Powrót do wcześniejszej wersji (bez funkcjonalności) (reflog, reset, status, log, zawartość katalogu)
* Wypisanie skróconych tagów



* Cofanie do początku – Initial (soft – zmiany pozostają i występują jako niezacomitowane)

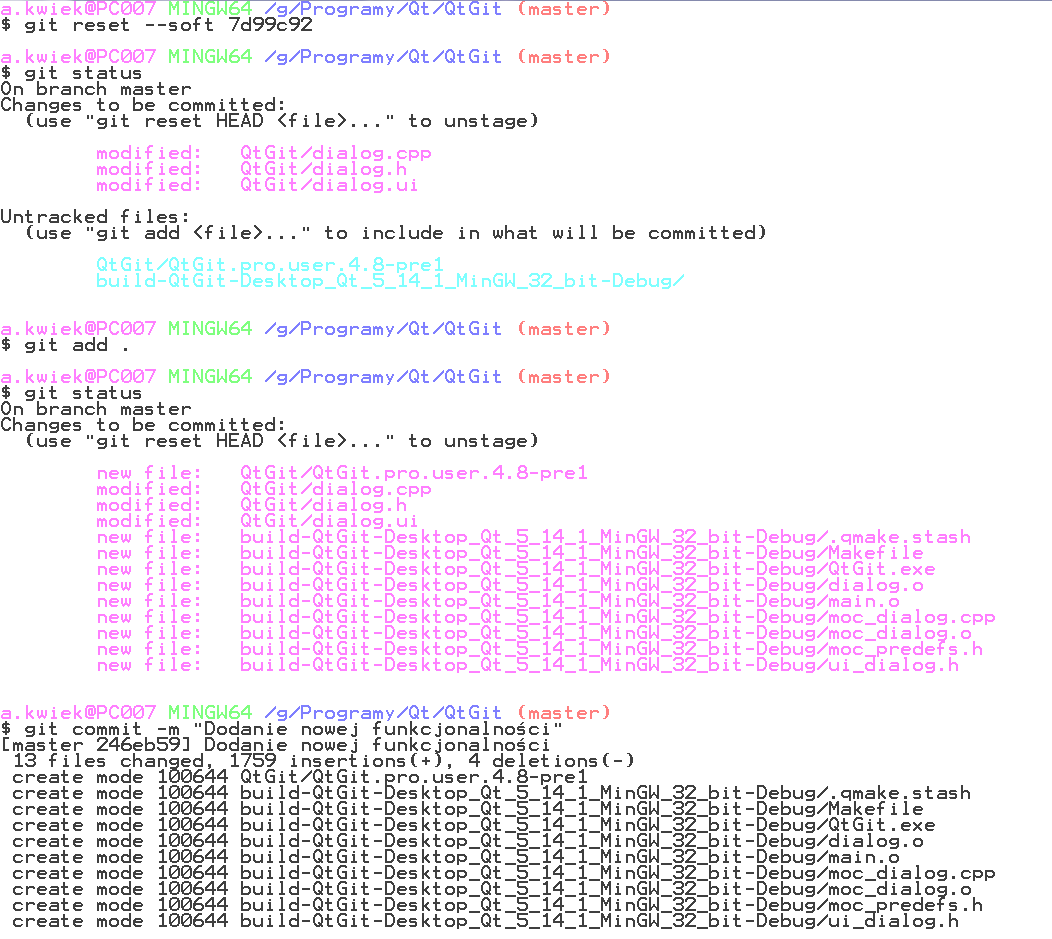


* Dodanie nowej funkcjonalności:

- Po zmianie textu w textEdit, tekst ląduje na etykiecie. (status, log, zawartość katalogu)

- Po naciśnięciu buttona textedit czyszczony.(status, log, zawartość katalogu)

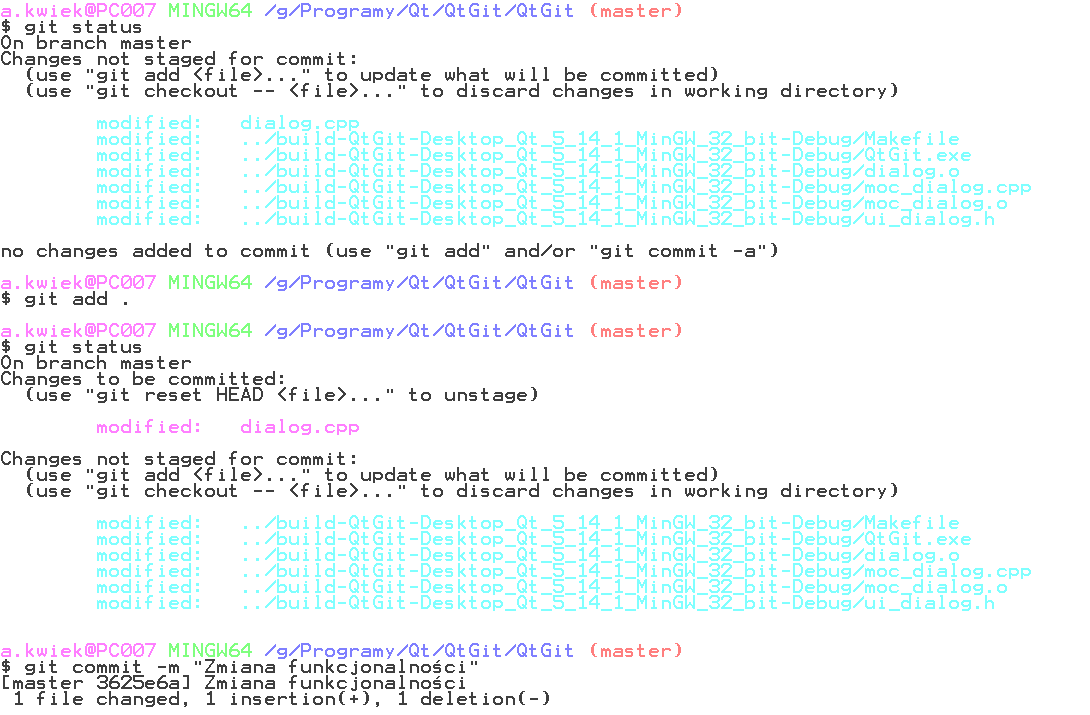
* Commit ((status, log, zawartość katalogu)



* Zmiana funkcjonalności:

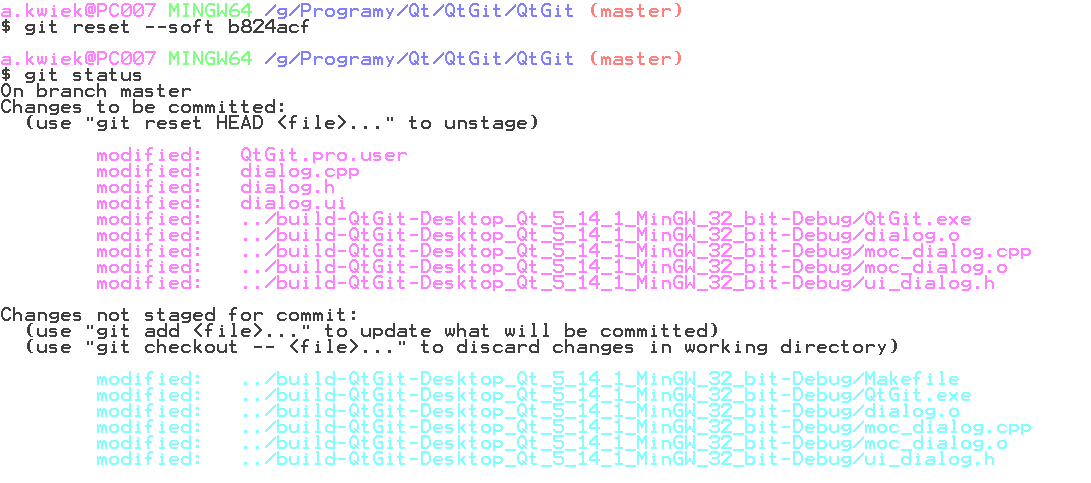
- Po naciśnięciu buttona textedit czyszczony, do label wpisywany tekst „empty”. (status, log, zawartość katalogu)

* Commit (status, log, zawartość katalogu, wizualizacja historii – oneline, author)



**GAŁĘZIE**

* Powrót do wcześniejszej wersji (bez funkcjonalności) (status, log, zawartość katalogu)



* Utworzenie gałęzi textcopy i przełączenie się na nią (status, log, zawartość katalogu)



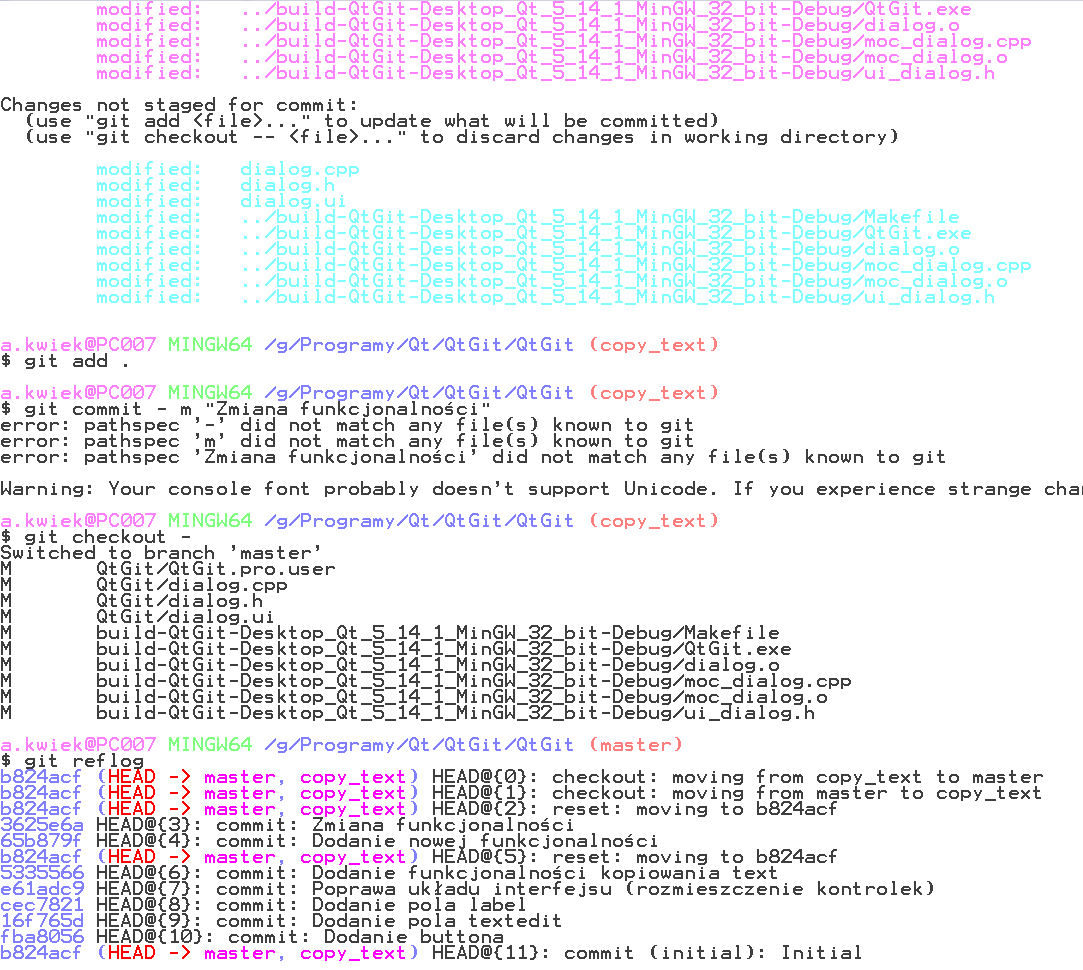
* Dodanie funkcjonalności:

- Po wpisaniu textu w textEdit i naciśnięciu buttona, text ląduje na etykiecie. (status, log, zawartość katalogu)

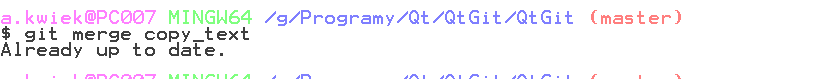
* Commit (status, log, zawartość katalogu)



* Przełączenie się na master (status, log, zawartość katalogu, sprawdzenie galezi)



* Połączenie funkcjonalności(status, log, zawartość katalogu)



* Powrót do wcześniejszej wersji (bez funkcjonalności)
* Utworzenie gałęzi textcopy i przełączenie się na nią
* git status (czy potrzebny commit?)
* Dodanie funkcjonalności:

- Po wpisaniu textu w textEdit text ląduje na etykiecie. (status, log, zawartość katalogu)

* Przełączenie na master. (status, log, zawartość katalogu)
* Utworzenie gałęzi textclear i przełączenie się na nią(status, log, zawartość katalogu)
* git status (czy potrzebny commit?)
* Dodanie funkcjonalności:

- Po naciśnięciu buttona text w textEdit jest czyszczony. (status, log, zawartość katalogu)

* Przełączenie na master.
* Połączenie funkcjonalności z gałęzi textcopy. (status, log, zawartość katalogu)
* Próba połączenia funkcjonalności z textclear. (status, log, zawartość katalogu)

----------------