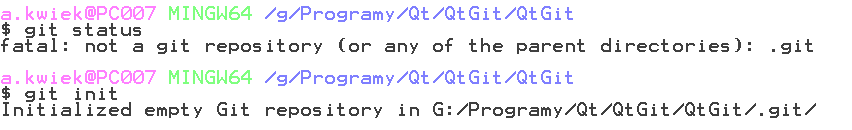
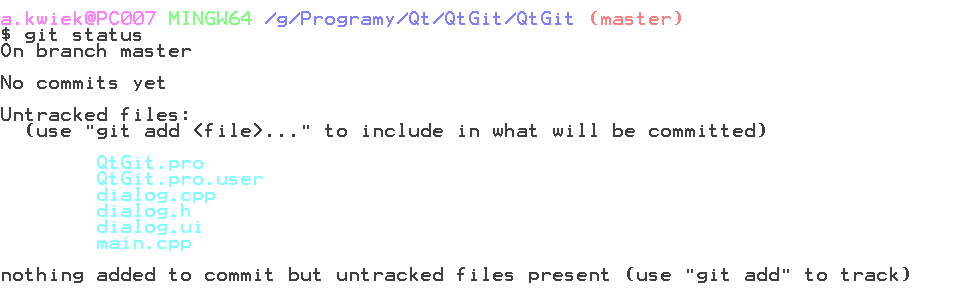
* Utworzyć i zbudować nowy projekt Qt - QDialog

# INICJOWANIE REPOZYTORIUM

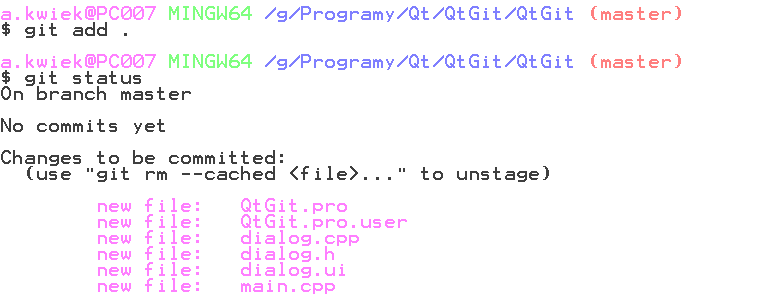
* Zainicjowanie repozytorium
* Pierwszy commit Initial (status, init, log, zawartość katalogu)
* Inicjalizacja repozytorium



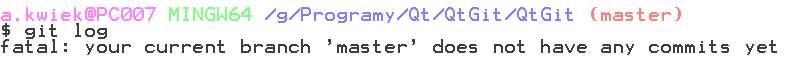
* Sprawdzenie statusu



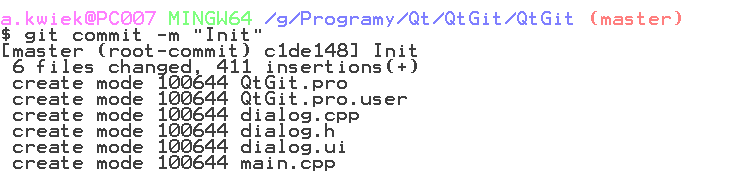
* Przygotowanie commitu (dodanie plików) i sprawdzenie statusu



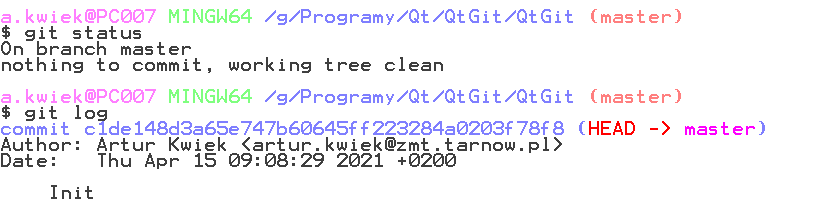
* Sprawdzenie historii comitów



* Commit

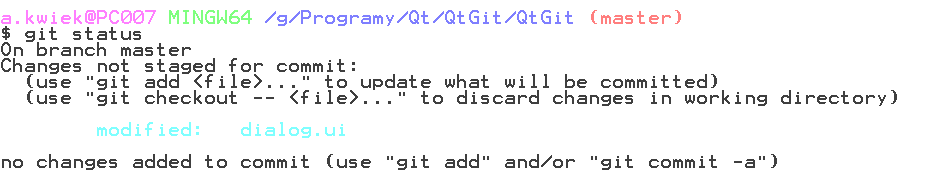


* Sprawdzenie statusu i historii commitów

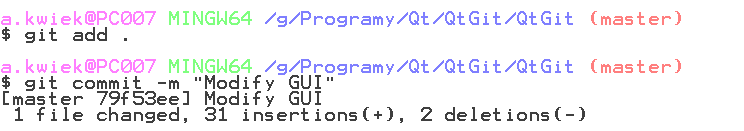


# ZAPISYWANIE ZMIAN

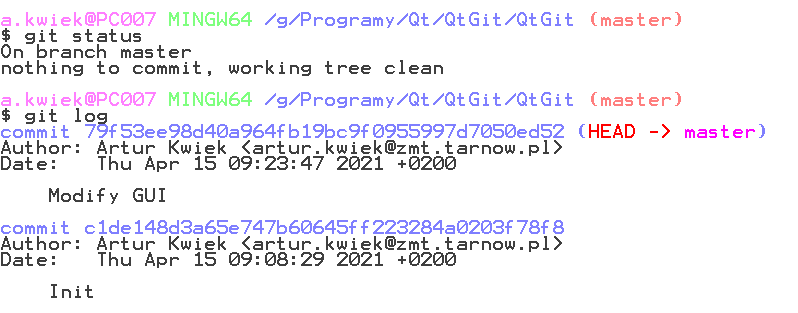
* Dodanie elementów i modyfikacja interfejsu - QButton, QLineEdit, QLabel (status, log, zawartość katalogu)



* Przygotowanie plików do commitu i commit

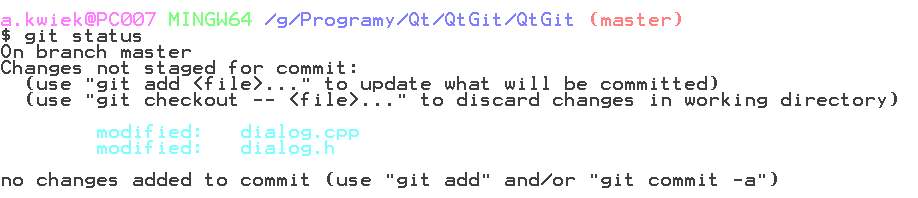


* Sprawdzenie statusu i historii zmian (status, log, zawartość katalogu)

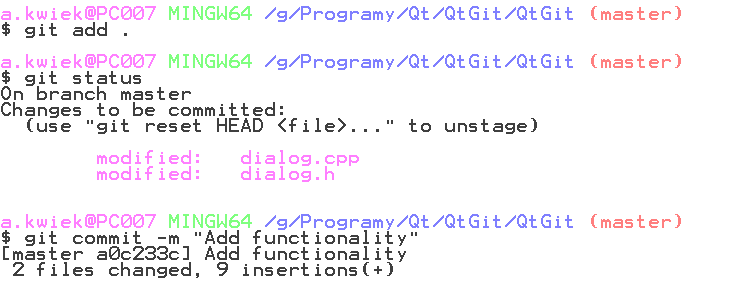


* Dodanie funkcjonalności:

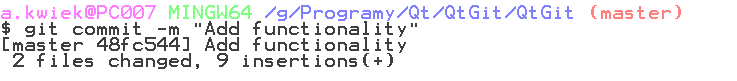
- Po wpisaniu tekstu w textEdit i naciśnięciu buttona, tekst ląduje na etykiecie. (status, log, zawartość katalogu)



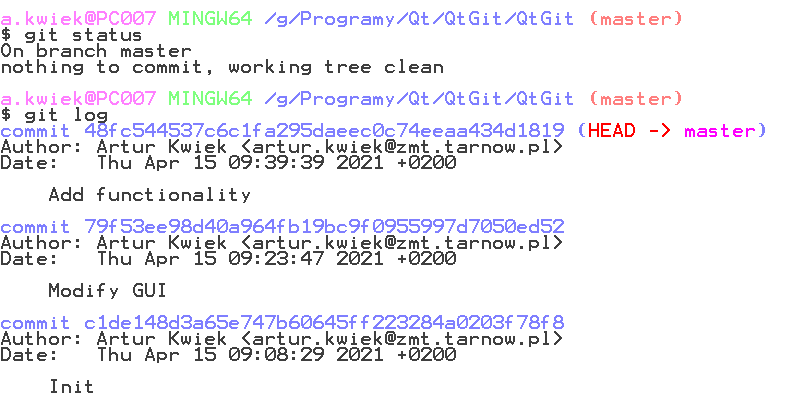
* Przygotowanie plików do commitu (dodanie na stage)



* Commit



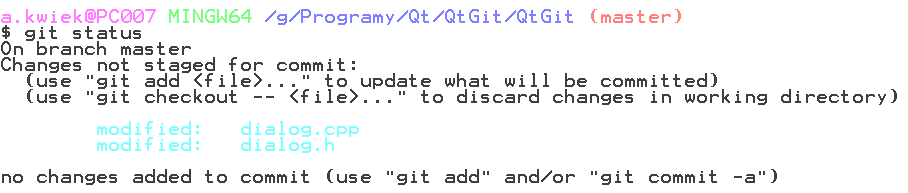
* Sprawdzenie statusu i przegląd commitów



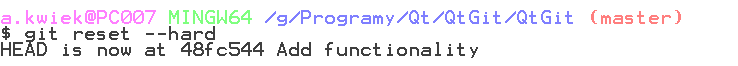
# COFANIE ZMIAN

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

* Wprowadzenie zmian w projekcie
* Po wpisaniu tekstu w pole edit tekst kopiowany jest automatycznie do etykiety label
* Po naciśnięciu buttona pole tekst edit jest czyszczony

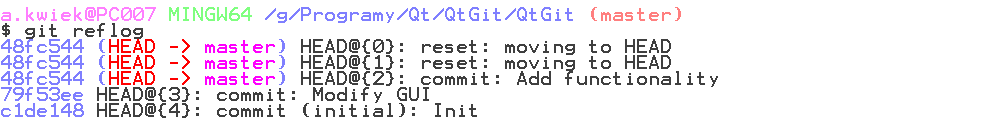


* Cofnięcie bieżących zmian w plikach



Teraz zawartość plików jest taka jak przed zmianami (po commicie „Add functionality”)

* Cofnięcie zmian do wcześniejszego commitu
* Wypisanie skróconych tagów

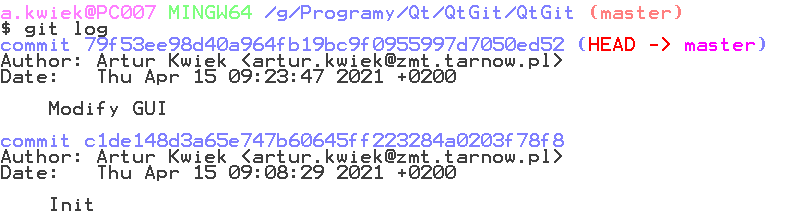


Poruszanie się po historii możliwe jest przez polecenie checkout lub git reset --soft

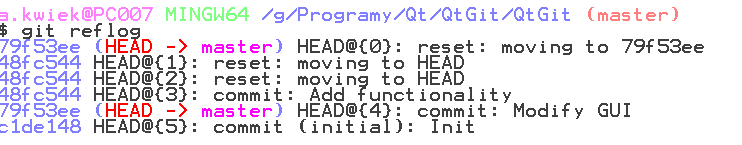
* Powrót do wcześniejszej wersji (stan projektu bez funkcjonalności) (reflog, reset, status, log, zawartość katalogu)



Bieżąca historia commitów odwołuje się do aktualnego dowiązania (log)



Ale pełna historia zmian w repozytorium ciągle jest dostępna

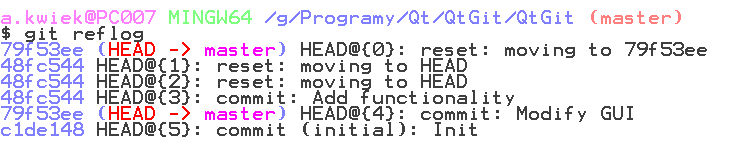


# GAŁĘZIE

* Przegląd gałęzi repozytorium



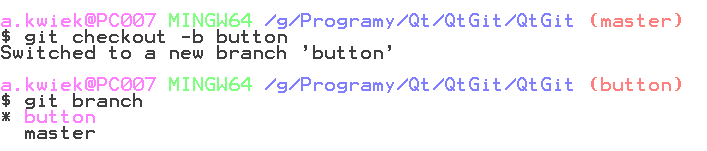
* Powrót do wcześniejszej wersji (Initial) (status, log, zawartość katalogu)



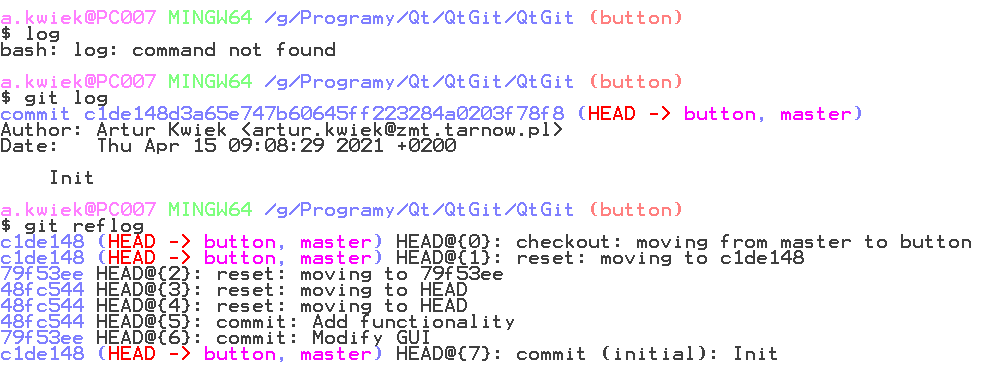


W tym stanie mamy pusty dialog, bez dodanych wcześniej widgetów

* Utworzenie gałęzi button i przełączenie się na nią (checkout -b, branch, status, log)

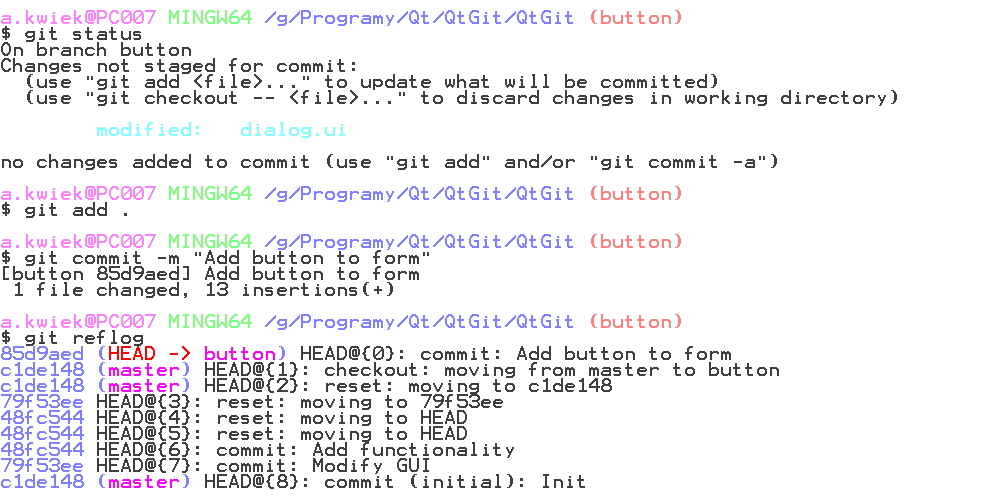


Nowa gałąź po utworzeniu posiada taką samą historię jak gałąź główna

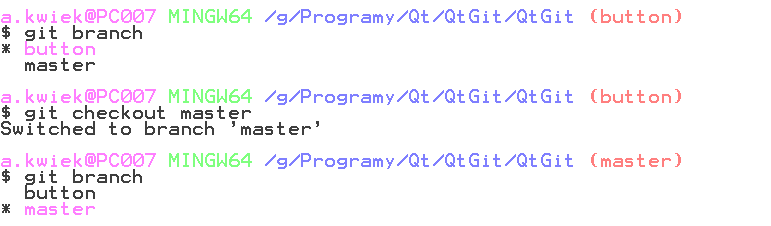


* Dodać do formatki przycisk i wykonać commit

Na gałęzi tworzona jest dalsza historia zmian plików

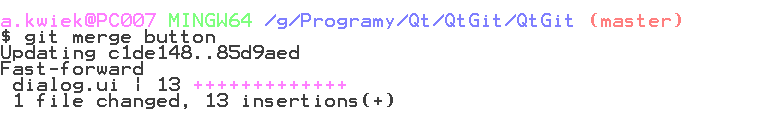


Przełączenie się na gałąź master.



Teraz ponownie mamy stan z pustą formatką

* Połączmy zmiany na gałęzi button z aktualnym stanem gałęzi master



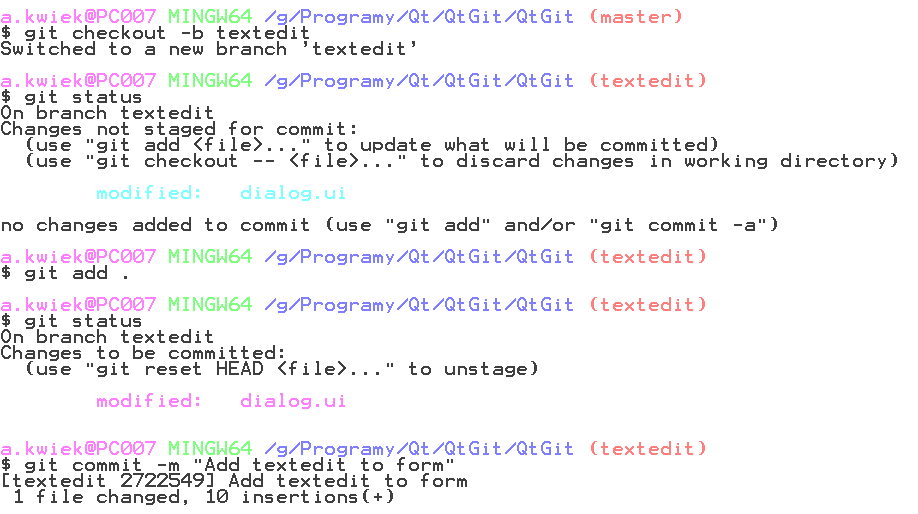
Zmiany z gałęzi button zostały naniesione na gałąź główną

# KONFLIKTY

* Utworzenie gałęzi textedit i przełączenie się na nią (checkout -b, branch, status, log)

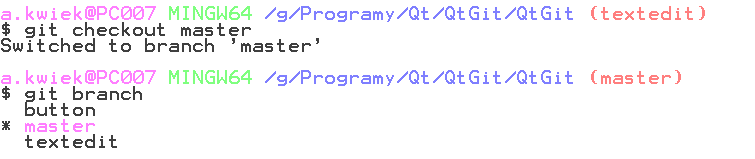


Dodać do formatki pole tekstowe QTextEdit i wykonać commit

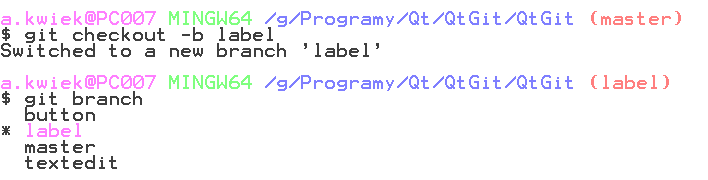


* Przełączenie się na gałąź master.

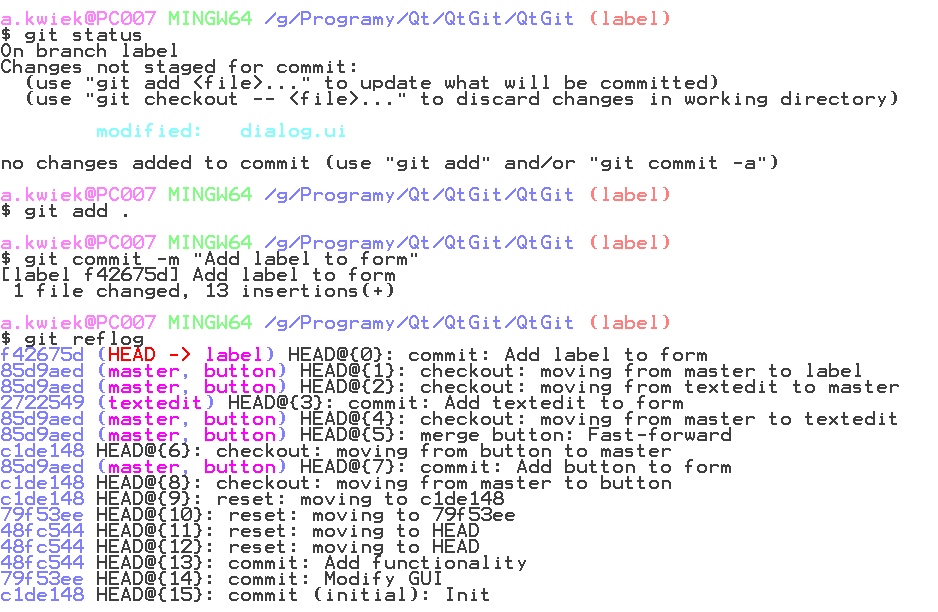
Teraz w repozytorium występują trzy gałęzie, a jego stan bieżący dowiązany jest do gałęzi master



* Utworzenie gałęzi label i przełączenie się na nią (checkout -b, branch, status, log)



Dodać do formatki pole tekstowe QLabel i wykonać commit (wyświetlić bieżącą historię zmian w repozytorium)



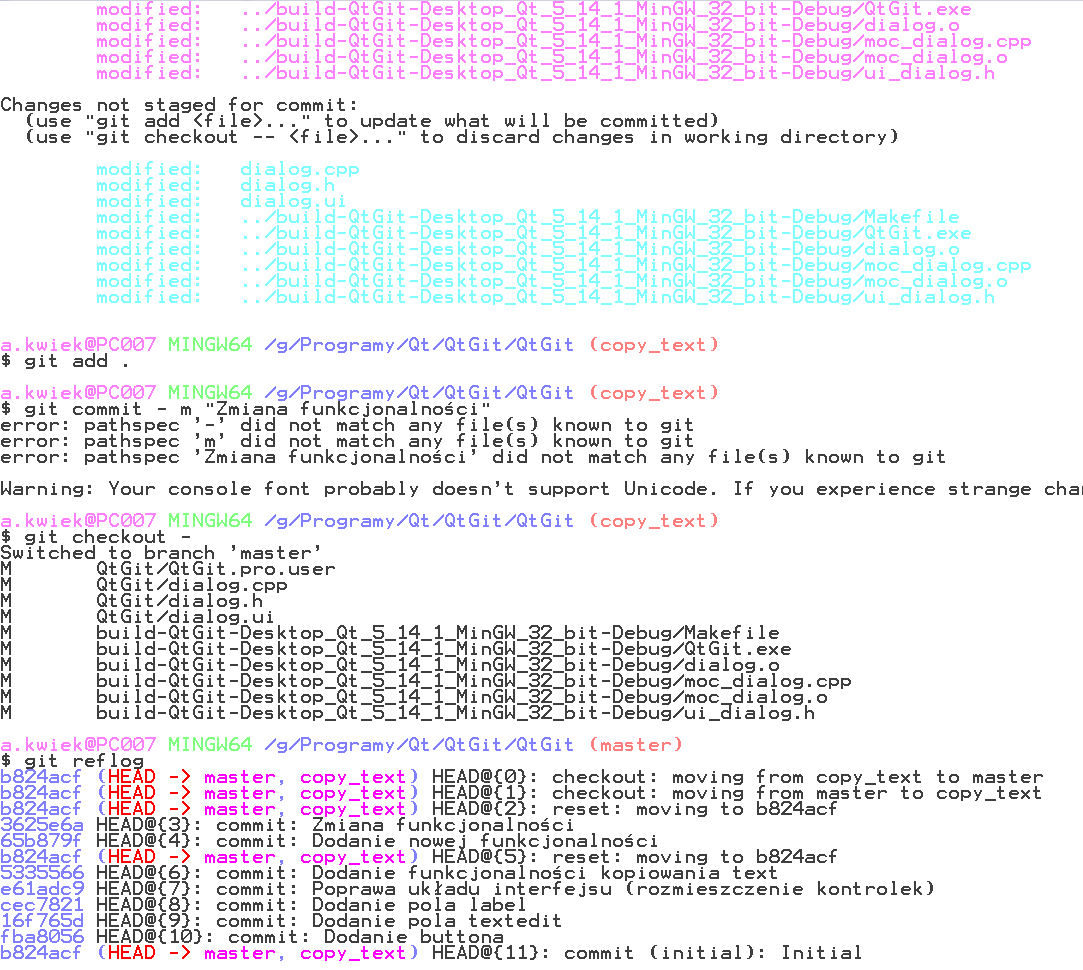
* Dodanie funkcjonalności:

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

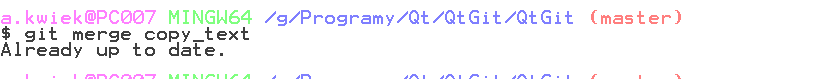
* Commit (status, log, zawartość katalogu)



* 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)

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