1 Documentazione / Commenti

1.a Generare la documentazione usando Doxygen e verificare se la documentazione prodotta permette di orientarsi nella struttura del progetto software. Indicare i punti che potrebbero essere migliorati.

1.b Utilizzando la documentazione e i commenti inseriti nei file. ccp e .h verificare se il progetto software aderisce alle specifiche assegnate durante la prima settimana. Indicare le eventuali specifiche non rispettate.

1.c Verificare che il file README introduca correttamente lo scopo del progetto software e che dia sufficienti informazioni per un corretto uso dell’interfaccia a riga di comando. Indicare eventuali mancanze e/o possibili migliorie.

2 Compilazione e prima sessione di test

2.a Verificare se è possibile compilare il progetto

2.b Test dell’interfaccia a riga di comando: l’interfaccia funziona correttamente? L’interfaccia è di facile utilizzo? Quali prove sono state eseguite per fare il test?

3 Seconda sessione di test

3.a Definire una procedura di test che permetta di identificare possibili bug nel codice prodotto per le singole classi e i singoli metodi di ogni classe

3.b Riportare i risultati del test

S5203046 Luigi Thea

1)

**a**. La documentazione permette di orientarsi. Segnalo tuttavia un possibile mia incomprensione, nella sottosezione delle classi compare due volte Right\_trapezoid.

Il main.cpp non ha riportato alcun commento Doxy-Compatibile se non quelli delle variabili, il che renderebbe teoricamente possibile un miglioramento, in ogni caso il codice rimane sufficientemente chiaro.

**b**. Le specifiche vengono rispettate, i commenti mi paiono chiari e aderenti alle specifiche.

**c**. Il ReadMe non è presente nel repository a me assegnata, è stata necessaria la condivisone del link di un altro repository per accedere al ReadMe. In ogni caso il ReadMe introduce lo scopo correttamente. È sintetico ma abbastanza efficace, mancano solo alcune informazioni sui parametri che l'utente può inserire(valori di interfaccia).

2)

**a**. Il codice presentato compila senza alcuna modifica o ritocco, tuttavia il valore di ritorno del main del progetto, il quale torno void, potrebbe creare errori su alcuni compilatori.

**b**. L'interfaccia funziona perfettamente, inoltre stampa anche la figura utilizzando la draw e varie righe di cout. L'interfaccia è molto intuitiva e il “BOX” in cui compaiono le possibilità di azione lasciate all'utente rende il tutto molto chiaro e leggibile.

Nella sessione di test preliminare si è verificato che fosse effettivamente possibile inizializzare i vari poligoni, è possibile visualizzare tutti quelli appena creati tramite

la funzione di stampa, che restituisce anche il numero di oggetti in memoria, ed infine di uscire dalla procedura, previa distruzione degli oggetti creati.

Nessuna di queste operazioni ha fatto emergere problemi di qualsivoglia entità, i valori, calcolati esternamente sono risultati inoltre corretti.

3)

La prima è stata verificare che, se erroneamente inseriti i valori di scelta (tra i poligoni da creare) il codice risponda correttamente rilevando l'errore e riponendo la richiesta. Il tutto funziona correttamente.

Per Seconda ho provato l'inserimento di valori negativi al codice, il codice risponde correttamente generando un errore che ne indica anche l'entità. Per quanto riguarda Rettangolo e Trapezio la scelta progettuale è stata di far rimanere in memoria l'oggetto con i dati NON corretti, questo si traduce in una stampa di un poligono di area e perimetro uguali a 0. Per quanto riguarda la classe rombo, sebbene generato correttamente l'errore i valori stampati sono ambigui e fondamentalmente scorretti (negativi e alfanumerici).

Situazione analoga si presenta sia quando, tramite l'inserimento a riga di comando, vengono passati valori nulli o caratteri della tabella ascii e non numerici: il codice segnala l’errore; tuttavia, i poligoni rimangono privi di significato in memoria.

La situazione del rombo rimane invariata, i valori di ritorno sono errati.

La costruzione, sia per default che passando i valori, e la distruzione degli oggetti funziona correttamente. La distruzione avviene, come da specifiche, solo nel caso si scelga "la exit" tra le opzioni di interfaccia.

Segnalo infine la presenza nell'interfaccia di una eccezione non gestita che fa bloccare il codice. Infatti, allo switch di interfaccia viene passato il valore true come variabile di controllo, portando dunque il while ad essere teoricamente eterno.

Non avendo dunque un controllo sul valore di poligoni creati e presenti in memoria è teoricamente possibile uscire dall'array di cento puntatori, il che si traduce appunto nella generazione di una eccezione e nel termine della simulazione.

Nella Repository di questa file è presente un main.cpp "aggiornato" che presenta commentata tutta la sezione di interfaccia ed una procedura di test già scritta (che include dunque valori negativi, valori nulli e caratteri non consentiti).