

Specifica

Il progetto realizza due giochi popolari, l'utente deve avere la possibilità di scegliere tra i due oppure chiudere il programma.

Hangman deve avere un posto dove l'utente può scrivere, deve mostrare una parte di stickman ogni volta che si sbaglia una lettera. Deve essere in grado di controllare anche intera parola inserita e fare tutti i controlli indipendentemente dalle dimensioni delle lettere.

BrickBreaker deve avere una palla che si muove e distrugge i mattoni e una tavola che utente può muovere per rimbalzare la palla.

Interfaccia è grafica di un design chiaro e semplice da utilizzare.

Studio del problema

Hangman

Come detto prima il progetto è stato ispirato dal gioco famoso e adatto per l'uso personale (senza altra persona che immagina la parola e le altre che aiutano indovinando le lettere varie.

I problemi principali quindi sono

- Ottenere la parola
- Fare possibile la verifica sia di una lettera sola che la parola

La parola nascosta si ottiene da un txt file che ho riempito con parole interessanti/rare trovate su internet e per ogni c'è una spiegazione. Il programma prende una riga casuale di tipo PAROLA/SPIEGAZIONE e la divide. Così dopo un paio di sbagli si stampa allo schermo un "hint" mentre la parola stessa rimane nascosta fino alla sua scoperta oppure fino a quanto intero hangman è visibile.

Per acquisire la lettera o parola dall'utente ho deciso di mettere un textBox nella forma. Ho scelto quello piuttosto che i vari bottoni con le lettere sulla forma perché trovo più comodo usare la tastiera che il mouse per clickare allo schermo.

Per la verifica, se si prova a indovinare una sola lettera devo controllare se c'è e anche la posizione, altrimenti si stampa come una lettera non presente e si aggiunge allo schermo una nuova parte di hangman. Se si prova invece indovinare intera parola e si sbaglia l'utente va informato e può riprovare.

Altre funzionalità aggiunte sono la musica e il help relativo al gioco.

BrickBreaker

I problemi generali erano:

- Movimento
- Limiti

Per illustrare il movimento ho usato timer con intervallo diminuito a 25ms per rendere il movimento più scorrevole.

Anche questo ha la musica e anche i suoni per caso di vincita o perdita.

Altri

In entrambi casi c'erano diversi problemi di grafica ma decisamente quello più grande era RESTART. Per Hangman dovevo fare attenzione alla visibilità delle componenti e se i valori delle variabili (svuotare i string, il punteggio...)

Per il Brick purtroppo non sono riuscita a fare restart, perché non riuscivo a posizionare i mattoni programmaticamente.

Scelte architettoniche

Hangman

Il metodo ScegliParola scorre il file txt e estrae una riga casualmente.

```
ReadToEnd().Split(new[] { '\n' }, StringSplitOptions.RemoveEmptyEntries);
```

Per individuare le righe del testo ho definito uno TextReader tr per scorrere un file txt e individuare solo righe. Poi con

```
string[] soluzione = parolacor.Split('/');
```

la riga va divisa in parola nascosta (fino a /) e il hint che è tutto dopo /.

Poi per la verifica vedo cosa viene inserito in textBox, nel caso di una lettera viene cercata la posizione della lettera inserita per poi essere stampata in alto (label7) e per non avere poi la lettera scritta più di una volta nella sezione YUP (lettere presenti) utilizzo la funzione Contains.

```
StringBuilder clue = new StringBuilder(trattini);
```

Per stampare i trattini bassi corrispondenti al numero delle lettere della parola nascosta e cambiarli per le lettere indovinate nel tempo reale ho scritto un metodo GetClue che crea un string 'trattini'. Poi durante la verifica nel metodo Controlla converte lo string in uno StringBuilder per abilitare la trasformazione del trattino nella lettera.

Per l'inserimento della parola intera va controllato se sia completamente uguale a quella nascosta o meno. In entrambi casi esce un messaggio che informa l'utente del esito. Se ci sono più di 3 errori viene stampato lo hint ottenuto prima dal file.

Il metodo Reset è abbastanza semplice, torna allo stato iniziale intera forma, nasconde i label con aiuto, svuota i string, torna a 0 il numero di errori.

Per chiudere la forma ho usato this.Hide() per fare possibile rientrare dalla forma principale.

BrickBreaker

La tavoletta si muove usando le frecciette della tastiera. Ho apperto l'evento KeyDown in quale verifico quale delle due (destra o sinistra) è premuta con e.KeyCode. Così si muove ogni volta che premo il bottone ma anche se lo tengo premuto. Una volta raggiunti i bordi della forma la tavoletta si ferma.

Per la palla invece aggiungo timer e due integer. Per ogni timer tick la posizione di palla si cambia per 4 pixel sulla asse X e 3 sulla Y. Ogni volta che distrugge un mattone o raggiunge la tavoletta il valore del movimento sulla Y si ribalta.

Metodo GetPunteggio controlla dove passa la palla e se la sua posizione interseca con un button (mattone) lo rimuovo. Se sono tutti rimossi, si vede l'immagine del porto pesarese, un messaggio con scritto HAI VINTO e si sente la musica del vincitore, se invece la pallina è caduta sotto la tavoletta, viene stampato il messaggio opposto e si sente un suono diverso. Come ho detto prima avevo idea di come creare i bottoni novi ma non riuscivo a metterli in ordine e mettere l'immagine giusta, quindi una volta chiusa questa forma non si può riaprire. Con un blocco try catch nella forma principale raccolgo l'exception di forma non esistente (dato che qui per chiudere ho usato this.Close()).

Forma principale

La forma principale contiene un pictureBox clickabile per ogni gioco, un pictureBox per chiusura del programma e un messaggio.

Use cases

Data la semplicità del progetto, per quanto riguarda il numero di classi i use cases non sono limitati.

Use case 1

ID

- UCase1

Actor

- Utente

Precondition

- Ha scelto BrickBreaker

Basic course of events

- Utente può clickare sulla nota per ascoltare la musica
- Utente clicca start per iniziare
- Utente utilizza le frecce per muovere la tavola
- Dipendendo dalla velocità Utente può vincere o perdere, in ogni caso la forma si chiude

Postcondition

- Si riapre la forma principale

Use case 2

ID

- Ucase2

Actor

- Utente

Precondition

- Utente ha scelto Hangman

Basic course of events

- Volendo, Utente clicca nota per far iniziare la musica
- Utente clicca start per iniziare
- Il programma sceglie casualmente la parola dal file
- Utente inserisce una lettera che poi si vede tra i trattini nel caso sia corretta, o tra le lettere no in caso opposto
- Utente inserisce una parola per provare a indovinare la parola nascosta
- La parola viene controllata e Utente è informato sull'esito
- Utente clicca EXIT e la forma si chiude

Postcondition

- Si riapre la forma principale

Use case 3

ID

- Ucase3

Actor

- Utente

Precondition

- Utente ha clickato lo smiley che piange

Basic course of events

- Si chiude il programma.

Postcondition

- Non ci sono.

