שלום ליאת פסטרנק

ש"ב 2 נועדו לתרגל אתכם על כתיבת GUI שכולל כמה מהפקדים שפגשתם, לדוגמא - ComboBox, TextBox, Label, Button וכמובן Form.

הפעם אני שולח לכם פרויקט מותחל. הפרויקט הוא בעצם סתם פרויקט WindowsFormApplication שהוספתי לו תיקייה שמכילה תמונות של דגלים של מדינות. עליכם להשלים את הפרויקט ולפתח את הקוד שלו כך שיענה לדרישות המפורטות.

כרגיל, יש להכין את הפרויקט, לכווץ ולהעלות אותו לאתר הקורס. ושוב, כרגיל - עם שאלות על הש"ב הללו תפנו אליי. בשאלות כלליות לגבי C# תיפנו אליי או אל אמיר.

הבודק האוטמטי אמור לענות לכם עם ציון בתוך דקות ספורות מההגשה (האמת שבתרגיל זה התשובה יכולה לקחת קצת יותר דקות - בגלל השימוש בTimer הבדיקה לוקחת יותר זמן אבל לא הרבה דקות). אם לא חזרה תשובה או לא ברורה התשובה או כל שאלה - תודיעו לי שאוכל לבדוק מה "נתקע".

המלצתי האישית היא לבדוק (ואם צריך לתקן) את תוכניתכם לאחר ביצוע של כל אחד מהסעיפים הבאים:

0) שנו את תכונת הטופס ControlBox ל-false כך שהמשתמש לא יוכל להגדיל\להקטין\לסגור הטופס מהפינה הימנית העליונה של הטופס. את

1) הוסיפו ComboBox לטופס Form1. שנו את תכונת ה-Text של ה-ComboBox ל-"...Choose a country"

בשלב הזה הטופס אמור להראות דומה לתמונה הבאה (כמו שאמרתי, אין חשיבות לגודל הטופס ולמיקום ה ComboBox כל עוד רואים אותו כמובן.!)



2) את הפריטים שיציג ה-ComboBox אתם תוסיפו באמצעות קוד שיסרוק את תיקיית Flags שהוספתי לכם ולכל תמונה שתימצאו שם תוסיפו את הפריט המתאים. כלומר, אם בתיקיה Flags ששלחתי לכם בפרויקט מופיעים הקבצים:

Israel.png

United States.png

Brasil.png

אז כאשר תיפתחו את ה-ComboBox הטופס צריך להיראות בערך ככה:



מספר נקודות לגבי הפריטים:

על התוכנה שלכם לסרוק את התיקיה Flags בזמן הריצה ולפי הקבצים שהיא מוצאת להכניס איברים ל-ComboBox. הרעיון הוא שבזמן התכנות עוד לא ידוע לכם בדיוק אילו קבצים יהיו בתיקיה. אני אבדוק את זה ע"י הכנסת קבצי דגלים אחרים לתיקיה בזמן הבדיקה.

חשוב מאד! - הגישה אל התיקייה Flags ואל הקבצים שבה חייבת להיות יחסית למיקום של קובץ ה-executable שלכם (כדי שהבודק יוכל למצוא את הקבצים הללו). כלומר, כיון שה-executable שלכם נמצא בתיקיה bin//Debug ומכיון ששמתי לכם את התיקיה Flags בתוך התיקיה המרכזית של הפרויקט - הגישה אל התיקיה Flags תהייה ע"י הנתיב (path)

/../.."Flags"

והגישה(לדוגמא) אל הקובץ Brasil.png תהייה דרך הנתיב

/../.."Flags/Brasil.png"

אין חשיבות לסדר האיברים ב-ComboBox.

האיברים ב-ComboBox מכילים רק את שם המדינה (ללא הסיומת של הקובץ שמכיל את התמונה).

2) בואו נשתמש בתמונות. לאחר שהמשתמש בחר מדינה כלשהיא על ה-ConboBox עליכם להציג את התמונה של הדגל המתאים שקיבלתם בתיקיה Flags כתמונת הרקע של הטופס. שנו את התכונה BackgroundImage לתמונה הרצויה וכוונו את התכונה BackgroundImageLayout לערך Stretch כדי שהתמונה תיכנס במלואה לתוך הטופס. כך ש(לדוגמא) אם המשתמש בחר ב-ComboBox ב-Brasil אז על הטופס להיראות כמו בתמונה הבאה:



בשלב הזה תוכלו לבדוק שאם תיבחרו מדינות אחרות ב-ConboBox, אז גם התמונה תתחלף. יותר מאוחר בתרגיל לא תוכלו לבצע את הבדיקה הזו כי ה-ComboBox לא יהיה מאופשר או שיהיה חבוי.

3) עכשיו נהפוך את הטופס שלנו ל-חידה. נוסיף לטופס TextBox שבו המשתמש ייצטרך להקליד את שם המדינה. לפני שהמשתמש בחר מדינה ב-ComboBox ה-TextBox יהיה חבוי כלומר - תכונת ה-Visible תהיה false. לאחר שהמשתמש בחר מדינה והתמונה מוצגת נאפשר לו לכתוב ב-TextBox ע"י שינוי התכונה Visible לערך true. מרגע שהמשתמש בחר המדינה כלשהיא ב-ComboBox והחידה התחילה (היפעלנו את ה-TextBox) יש להעביר את ה-ComboBox למצב חבוי (כלומר לשנות את תכונת Visible של ה-ComboBox לערך false) כדי שהמשתמש לא יוכל להחליף חידה עד שלא סיים לפתור את זאת שכבר בחר.

עכשיו הטופס שלכם (אחרי בחירה של מדינה ב-ComboBox) ייראה בערך כך:

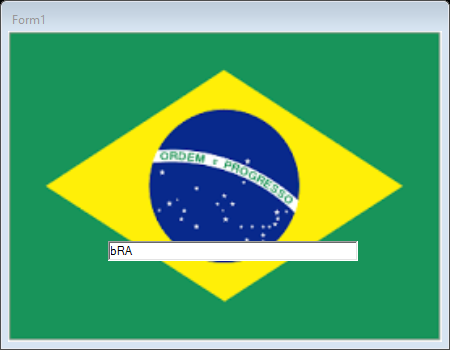


4) הוסיפו Event\_Handler ל-TextBox שייטפל בארוע של TextChanged. ב-Event Handler שתכתבו עליכם להחליט האם מה שכתב המשתמש ב-TextBox הוא התחלה של שם המדינה (כפי שהוא מופיע ב-ComboBox).

ההשוואה לא צריכה לקחת בחשבון האם המשתמש כתב באותיות קטנות או גדולות או במעורבב קטנות וגדולות.

אם המשתמש כתב התחלה שגויה - יש לשנות את הרקע של ה-TextBox לאדום. הרקע יישאר אדום כל עוד המשתמש לא תיקן את הכתוב ב-TextBox להתחלה של שם המדינה.

לדוגמא, אם המשתמש בחר ב-CheckBox במדינה Brasil והכניס ל-TextBox את ההתחלה bRA - ה-TextBox יישאר לבן כי זו באמת ההתחלה של המילה Brasil בהתעלמות מאותיות קטנות\גדולות. אבל אם המשתמש הוסיף את האות U ועכשיו ב-TextBox מופיע המילה bRAU, אז הרקע של ה-TextBox יהפוך לאדום, כמו בתמונות הבאות:





5) נוסיף מונה זמן שיודיע למשתמש כמה זמן עבר מתחילת החידה (תחילת החידה נחשבת מהזמן שבו המשתמש בחר באיזושהיא מדינה ב-ComboBox). לצורך תזדקקו ל-Timer שיעזור לכם לספור את הזמן ולפקד מסוג Label שבו תדווחו על הזמן שעבר. אני מצפה שבכל שנייה שעוברת תשנו את הכתוב ב-Label. לדוגמא, אחרי 5 שניות מתחילת החידה יופיע ב-Label הכיתוב הבא: Your time is:5 seconds. כמובן שלפני שהחידה "התחילה" ה-Label שמציג את הזמן לא צריך להופיע כלל (כנראה ע"י קביעת תכונת Visible שלו ל-false).



ספירת הזמן לא עוצרת גם כשהרקע של ה-TextBox הוא אדום וגם כאשר הרקע הוא לבן.המונה ימשיך לספור את הזמן עד שהמשתמש יסיים לכתוב את המילה הנדרשת. מה שמביא אותנו לסעיף הבא.

6) כאשר סוף סוף המשתמש סיים לכתוב את המילה נכון נעצור את מונה הזמן ובנוסף נודיע לו ע"י Label חדש שהוא סיים לפתור את החידה וכמה טוב הוא עשה. ההודעה שנכתוב לו ב-Label החדש תהיה תלויה בכמה זמן לקח לו לפתור את החידה. ההודעה ב-Label תהיה מנוסחת באופן הבא:

,(מילת תואר) you solved the word (המילה שנפתרה) in (הזמן שעבר) seconds

כאשר (הזמן שעבר) מציין את מספר השניות שעברו מתחילת החידה

כאשר (המילה שנפתרה) מציין את שם המדינה שהמשתמש בחר ב-ComboBox

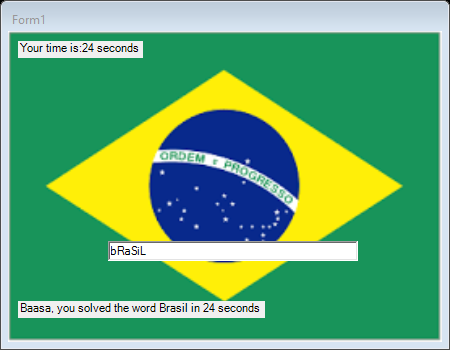
כאשר (מילת תואר) תהייה:

Horrey - אם המשתמש סיים בתוך פחות מ-10 שניות

OK - אם המשתמש סיים ב10 עד 19 שניות

Baasa - אם המשתמש סיים ב 20 שניות או יותר

לדוגמא, אם המשתמש סיים את החידה Brasil ב-24 שניות, הטופס ייראה בערך כך:



7) רמז ראשון למשתמש - הוסיפו קוד שכאשר מבצעים MouseDown על ה-Form ה-TextBox יציג את המילה הנכונה (זו שכתובה ב-ComboBox) וכאשר המשתמש יבצע MouseUp על ה-Form תחזור המילה שהופיעה ב-TextBox לפני ה-MouseDown.

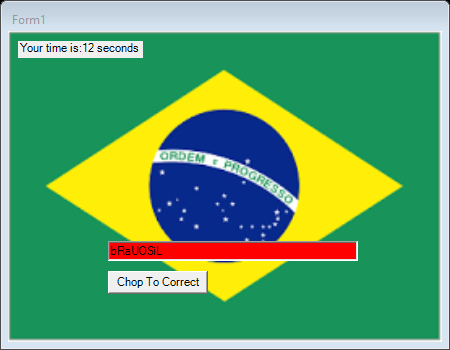
חשוב מאד מאד מאד !!! - לחיצת MouseDown שכזו לא צריכה לגרום לחידה להיפתר. לכן עליכם לדאוג שבמקרה של MouseDown, אפילו שהמילה הנכונה כתובה ב-TextBox עליכם להתעלם ממנה ולא להגיד למשתמש שהוא פתר נכון את החידה. אחת האפשרויות לעשות זאת (יש כמה וכמה כאלה) היא להוריד את ה-Event Handler שמטפל ב-TextChanged של ה-TextBox כאשר רוצים התעלמות שכזו. הורדת ה-Event Handler מתבצעת ע"י:

textBox1.TextChanged -= textBox1\_TextChanged;

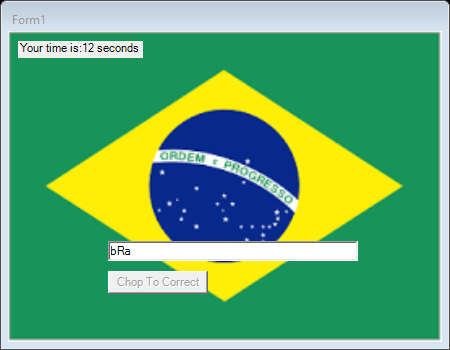
וכמובן להוסיף את ה-Event Handler מחדש אחרי שהחזרתם את המילה הנוכחית של המשתמש בסיום ה-MouseUp על ה-Form.

8) רמז שני למשתמש - הוסיפו כפתור חדש שהטקסט עליו יהיה "Chop To Correct". כל עוד המשתמש לא טועה (כל עוד הרקע של ה-TextBox הוא לבן), הכפתור החדש יהיה במצב לא מאופשר, כלומר תכונת Enabled של הכפתור החדש תקבל ערך false.

הוסיפו קוד שדואג שבכל פעם שהמשתמש טועה (הרקע של ה-TextBox הוא אדום) הכפתור החדש עם הטקסט "Chop To Correct") מופיע ומאופשר (Enabled). כאשר המשתמש לוחץ על הכפתור הזה, המילה שב-TextBox מתקצצת (יורדות אותיות מהסוף שלה) עד שהיא נהיית נכונה (עד שהיא נהיית באמת התחלה של המילה שב-ComboBox). לדוגמא, בתמונה הבאה מופיע הטופס במצב של טעות של המשתמש עם הכפתור החדש:



לאחר שהמשתמש ילחץ על הכפתור המילה שב-TextBox תתקצץ ל-bRa, הרקע של ה-TextBox יחזור להיות לבן וכפתור ה-Chop To Correct יחזור להיות לא מאופשר כמו בתמונה הבאה:



לסיכום מספר הבהרות:

1) כאשר החידה פועלת (המונה פועל) על ה-TextBox להיות כמובן גלוי ומאופשר. כאשר המשתמש סיים לפתור את החידה - על ה-TextBox לחזור למצב חבוי.

2) כאשר החידה פועלת (המונה פועל) והמשתמש שוגה (הרקע של ה-TextBox הוא אדום) על כפתור ה-Chop To Correct להיות במצב גלוי ומאופשר (Visible and Enabled). בכל מצב אחר עליו להיות במצב לא מאופשר.

3) כאשר החידה פועלת (המונה פועל) על ה-ComboBox להיות במצב חבוי. רק כאשר המתשתמש סיים לפתור את החידה ה-ComboBox חוזר להיות גלוי ומאופשר כדי לאפשר למשתמש להתחיל בחידה חדשה.

4) כאשר המשתמש בוחר להתחיל חידה חדשה (לאחר שסיים לפתור את הקודמת) יש להעלים את ה-Label מסעיף 6 שמבשר על סיום החידה (כי התחלנו חידה חדשה) ויש לאפס את מונה הזמן ל-0 לפני שמתחילים שוב את ספירת הזמן.

זהו, מספיק, לא?