

מבוא לתכנות - פיתוח משחקים ב Action Script 3.0

כל מה שמעצב משחקים צריך לדעת בשביל לעבוד עם מתכנתים

שעור 3 – משחק טקסט



1P 0 AS3 HIGH SCORE 2P 0

RANK	SCORE	NAME
1ST	00079000	G.F
2ND	00079000	S.G
3RD	00078500	O.S
4TH	00077500	R.G
5TH	00077000	Y.R
6TH	00076000	O.L
7TH	00075500	M.A
8TH	00074000	A.A
9TH	00072500	K.R
10TH	00071500	R.Z

CREDIT 15



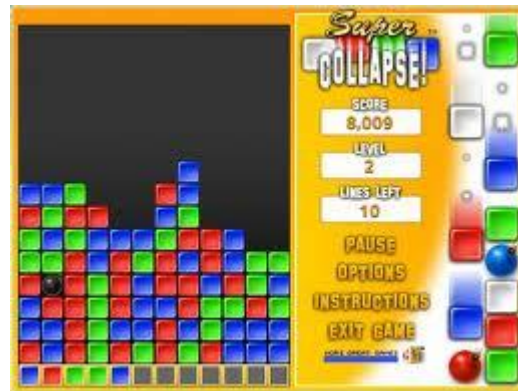
קבוצת Extreme

- מעבר על פתרון "בית ספר"
- מי שרוצה לשפר לאחר שראה את פתרון בית ספר
יוכל להגיש את התרגיל עד יום שלישי הקרוב !
- מי שסיים יתחיל כבר עכשיו היום את התרגיל החדש



תרגיל Extreme

- מטרת התרגיל:
- חלוקה נכונה של משחק למחלקות
- התמודדות עם פיתוח לוגיקה יחסית מורכבת



תרגיל Extreme



- מומלץ לבחור את המשחק שהכי מתחברים עליו (כדאי לבדוק את כל המשחקים ב YouTube)
- זמן הגשה מקסימאלי: 7 ימים לאפיון, 19 ימים למשחק.



עקרונות שפות תכנות - המשך



הצגת פתרון התרגיל

5,000 נקודות

■ כולל חזרה

■ מה זה משתנה?

■ מה זה טיפוס?

■ מה זה אופרטור?

■ מה זו השמה?

■ מה זה תנאי?



תנאים מתקדמים

שם משתנה, או
ערך בולאני

חלקים
קבועים

```
if (Boolean value) {  
    actions  
} else {  
    actions  
}
```

אוסף פעולות



תנאים מתקדמים

- מילה שמורה: else
- בלוק פעולות / הלוק קוד { }.

```
var points:Number;  
points = 100;  
if (points>200) {  
    trace("new high score");  
} else {  
    trace("try again");  
}
```

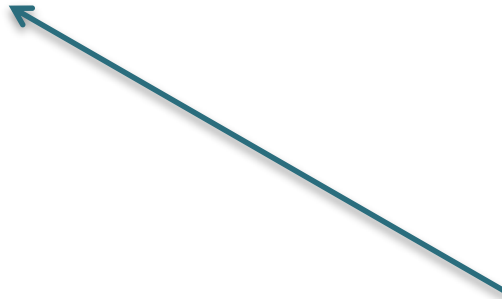


תנאים מקוננים

```
if (Boolean value) {  
    actions  
} else {  
    actions  
}
```

```
if (Boolean value) {  
    actions  
} else {  
    actions  
}
```

```
if (Boolean value) {  
    actions  
} else {  
    actions  
}
```



תנאים מתקדמים

```
var points:Number;  
points = 100;  
if (points>200) {  
    trace("new high score");  
} else {  
    if (points>100) {  
        trace("not bad");  
    } else {  
        trace("try again");  
    }  
}
```



תרגול תנאים

■ שפרו את קוד המשחק שכתבתם על ידי שימוש ב

else ■

■ תנאים מקוננים



אופרטורים לוגיים

■ עד עכשיו למדנו אופרטורים משלושה סוגים, זוכרים ?

משתנה	אופרטור השמה	ערך או משתנה		ערך מטיפוס מספר
ערך או משתנה	אופרטור אריתמטי	ערך או משתנה	⇒	
ערך או משתנה	אופרטור בוליאני	ערך או משתנה	⇒	ערך מטיפוס בוליאני



אופרטורים לוגיים

■ האופרטורים החדשים פועלים כך:

ערך או משתנה
בוליאני

אופרטור
לוגי

ערך או משתנה
בוליאני



ערך מטיפוס בוליאני

■ האופרטורים הלוגיים הראשונים שנלמד הם:

■ וגם שהסימן שלו הוא &&

■ או שהסימון שלו הוא ||



אופרטורים לוגיים

```
var points:Number;  
points = 100;  
if ((points>100) && (points<200)) {  
    trace("not bad");  
} else {  
    if (points>200) {  
        trace("very good");  
    } else {  
        trace("try again");  
    }  
}
```



אופרטורים לוגיים

&&

A	B	Res
F	F	F
F	T	F
T	F	F
T	T	T

||

A	B	Res
F	F	F
F	T	T
T	F	T
T	T	T



אופרטורים לוגיים

■ שימו לב לסוגריים יש להם הרבה משמעות

```
If ((true) && (false)) {  
    will I run ?  
}
```

```
If ((true) || (false)) {  
    will I run ?  
}
```

```
If (((true) && (false)) || (false)) {  
    will I run ?  
}
```

```
If ((true) && ((false) || (false))) {  
    will I run ?  
}
```



תרגול תנאים מתקדמים

- שפרו את קוד המשחק שכתבתם.
- כאשר השחקן מנחש את המספר במידה והמספר הוא בין 25 ל 75 תנו הודעה אחת, במידה והמספר קטן מ25 או גדול מ 75 תנו הודעה אחרת.
- השתמשו ב && או ב ||



אופרטור לוגי נוסף

&&

A	B	Res
F	F	F
F	T	F
T	F	F
T	T	T

||

A	B	Res
F	F	F
F	T	T
T	F	T
T	T	T

!

A	Res
F	T
T	F



אופרטורים לוגיים

- האופרטור ! נקרא Not והוא פשוט הופך את הערך.
- האופרטור מאוד מיוחד כי הוא פעול על ערך אחד בלבד ולא על שני ערכים.

```
if (!((!true) && (!false))) {  
    will I run ?  
}
```

```
if (!((!true) && (false)) || (false)) {  
    will I run ?  
}
```



Switch

- במקרים רבים אנו נדרשים לפעול בדרכים שונות במקרים שונים. שיש 2 או 3 מקרים IF מספיק, אך שיש יותר משתמשים ב Switch.

```
switch (variable) {  
    case value:  
        statements;  
    break;  
    case value:  
        statements;  
    break;  
}
```



Switch

```
var stage:Number;  
stage = 1;  
switch (stage) {  
    case 1:  
        trace ("Welcome to stage 1");  
        break;  
    case 2:  
        trace ("Stage 2 is harder");  
        break;  
}
```



Switch

- ב switch ניתן להשתמש רק בערכים מוחלטים ולא בטווחים.
- מאוד שימושי לממש מכונת מצבים.
- ניתן להזין מספר ערכים על ידי הפרדת בפסיקים
- ניתן להגדיר אוסף פעולות בררת מחדל



אופרטורים לוגיים

```
var stage: Number;  
stage = 1;  
switch (stage) {  
    case 1,2:  
        trace ("Welcome to the stage");  
        break;  
    default:  
        trace ("You are good");  
        break;  
}
```



מחרוזות



מחרוזות

- טיפוס חדש מצטרף אלינו: String (מה הטיפוסים האחרים שאנחנו מכירים?).

```
var myString : String;
```

- למשתנה מטיפוס מחרוזת ניתן לעשות השמה כך:

```
myString = "Dudi Peles";
```



אופרטור למחרוזות

- האופרטור + אותו אתם מכירים, עובד אחרת אם מפעילים אותו על מחרוזות.
- בכדי להבין מה הוא עושה נסו לחבר 2 מחרוזות ולהדפיס אותן



הסבות

- מספרים ומחרוזות הם מטיפוסים שונים ולכן לא ניתן לעשות פעולות בניהם.
- אם רוצים להסב מספר למחרוזת יש לעשות זאת על ידי שימוש ב `String()`
- אם רוצים להסב מחרוזת למספר יש לעשות זאת באמצעות שימוש ב `Number()`.

```
var myString:String;  
var myNumber:Number;  
myString = String(100+3);  
myNumber = Number(myString) + 3;  
myString = String(myNumber) + "3";  
trace(myString);
```



length

- פרט לשימוש באופרטורים קיימת עוד דרך לבצע פעולות על משתנים וזה על ידי הסימן "." (מה זה בדיוק נלמד בהמשך, בינתיים רק נשתמש).
- אם נרשום "." אחרי משתנה מטיפוס מחרוזת ואחר כך את המילה length נקבל ערך מספרי של כמות התווים במחרוזת.

```
var myName:String;  
myName = "Dudi Peles";  
var aNum:Number;  
aNum = myName.length;  
trace("my Name is "+String(aNum)+" letters long");
```



substr

- `substr` היא פקודה נוספת שניתן להפעיל על מחרוזות. מטרתה המרכזית לחתוך מחרוזות

```
newString = myString.substr(2,5);
```



שימו לב

- במחשבים אנחנו תמיד מתחילים לספר מ-0 ולא מ-1.
לכן:

```
var myName:String;  
myName = "Dudi Peles";  
trace("First Name:"+myName.substr(0,4));  
trace("Last Name:"+myName.substr(5,5));
```



תרגיל מחרוזות 1

- כתבו תוכנית ש
- שומרת שם וניקוד
- מכניסה אליהם ערכים
- אם השם מתחיל באות a, מכפילה את הניקוד
- מדפיסה את השם ואת הניקוד



search

- search היא פקודה שמטרתה היא למצוא מיקום של מחרוזת בת בתוך מחרוזת גדולה יותר

```
subStringPos = myString.search("a");
```



מיקום, מספרי



המחרוזת בה
מחפשים



מחרוזת הבת
אותה מחפשים



תרגיל מחרוזות 2

■ כתבו תוכנית ש

■ שומרת שם, הזינו שם ושם משפחה מפורדים ב " " .

■ מדפיסה את ראשי התיבות של השם, אות ראשונה של שם פרטי ואות ראשונה של שם משפחה מופרדים בנקודה.



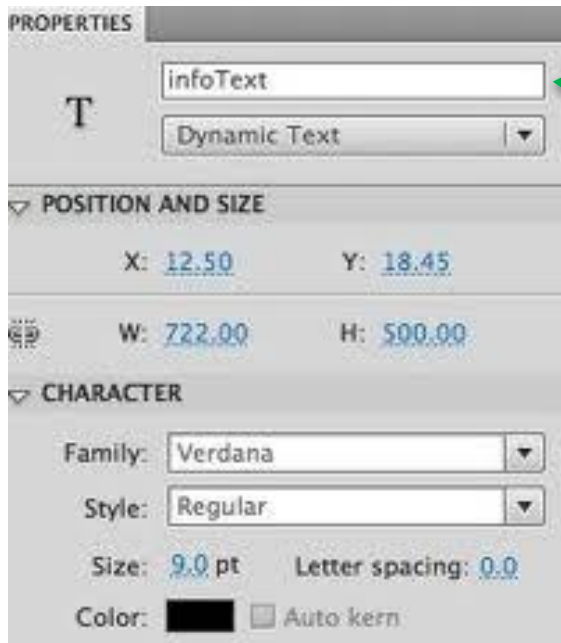
מחרוזות – דברים נוספים

- המחרוזת "n\" היא תו מיוחד שאם נמצא בתוך מחרוזת גורם לירידת שורה.
- משתנים מסוג מספר יכולים להכיל ערך יחיד שאינו מספר והוא NaN. ערך זה ימצא במספר אם לדוגמא מחרוזת שאינה מספר מוסבת למספר.
- ניתן לשאול על ערך מספר אם הוא NaN על ידי פקודת `isNaN`



שדה טקסט

- אני מאמין שכבר הכרתם את האובייקט: שדה טקסט



- אחד המאפיינים של שדה הטקסט הוא השם שלו

- בעזרת השם הזה ניתן לגשת את שדה הטקסט מהקוד. בכדי לגשת לטקסט של שדה הטקסט יש לרשום text.



שדה טקסט

- קיימים 3 סוגים של שדות טקסט:
 - Static Text – טקסט שלא ניתן לשנות בכלל
 - Dynamic Text – טקסט שניתן לשנות בעזרת קוד
 - Input Text – טקסט אותו המשתמש יכול לשנות

```
infoText.text = "Dudi Peles";
```

```
infoText.text = "my Name is "+inputText.text;
```



הסבר על מבנה התוכנית

```
public class Lesson3 extends MovieClip {  
  
    // var definetions  
    var randomNumber:Number;  
    // WRITE MORE VAR DEFINEITIONS  
  
    // Lesson3 constructor  
    public function Lesson3(){  
  
        stage.addEventListener(KeyboardEvent.KEY_DOWN, handleKeyDown);  
        randomNumber = Math.ceil(Math.random()*100);  
        //WRITE MORE CONSTRUCTUR CODE  
    }  
  
    // main function - runs each time a player hit Enter  
    function checkNumber(userNumber:Number){  
        //CODE TO RUN WHEN PLAYER CLICKS "ENTER"  
    }  
  
}
```



תרגיל שדות טקסט

- הפוך את משחק הבול פגישה למשחק בו תשובת המחשב מוצגות במסך ולא ב Trace.



סיכום

- תנאים מתקדמים
- מחרוזות
- שדות טקסט



תרגיל

■ שווי תרגיל: 45,000

נקודות

■ זמן הגשה מקסימאלי: 5

ימים

■ יחולק Achievement

מיוחד למשחק המוצלח

ביותר

