קודגורו אקסטרים

בשיתוף גבהים

יונתן פרי

ברק גונן

קודגורו אקסטרים

רקע על התחרות

אתגר מבית קודגורו

- **תחרות קבוצתית** בה כל קבוצה מפתחת (בשפת אסמבלי) תוכנה, המשמשת כשחקן בסבבי התחרות
  - מותר להרשם בקבוצות של עד 5 משתתפים
  - אופי התחרות מקנה יתרון לקבוצה על פני משתתף יחיד
- שחקן זה מכונה בשם "**שורד**" ותפקידו פשוט- לשרוד את מהלך המשחק
- ישנן טקטיקות רבות להשגת המטרה, נרחיב על כך בהמשך

קהל היעד

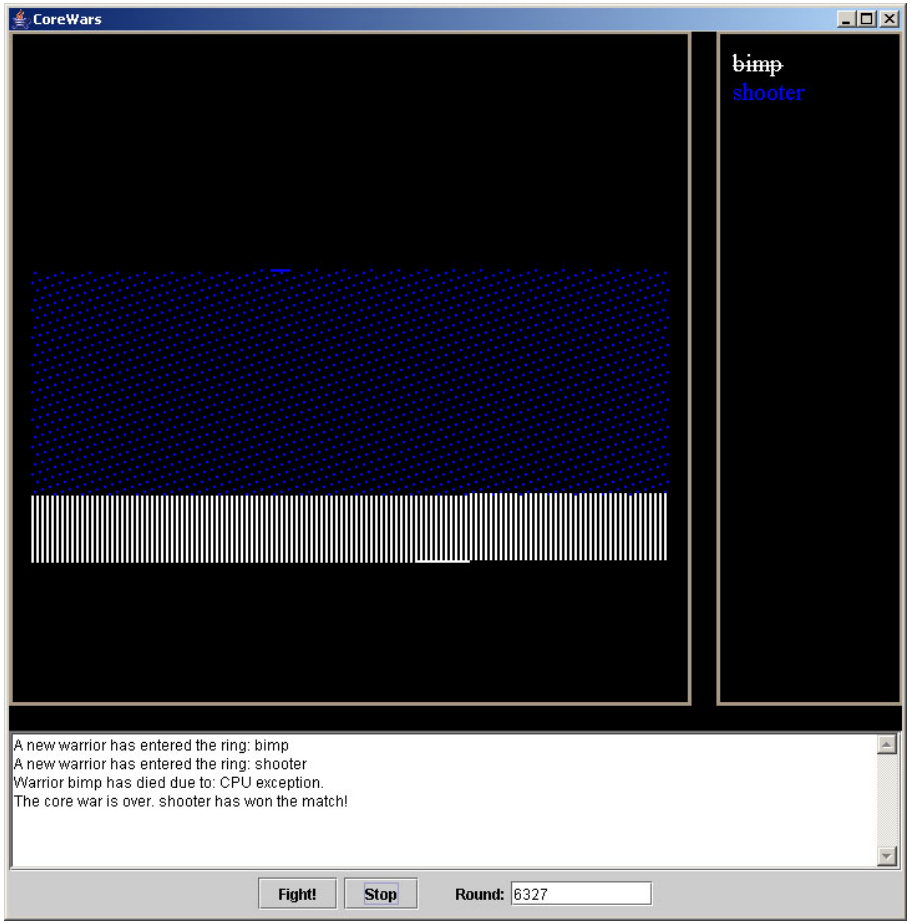
- כולם!
  - בפרט: **אתם ואתן!**
- דרישות הסף:
  - ידע בסיסי באסמבלי (אין צורך לדעת פרוצדורות ופסיקות)
  - מוטיבציה
  - רצון להתנסות בחווית משחק לא שגרתית

מנוע המשחק

- **זירה וירטואלית** בה מתרחש המשחק בין מספר תוכניות
- כל תוכנית ("שורד") נכתבת בשפת אסמבלי 8086, נטענת לזירת המשחק ומורצת במקביל לשאר התוכניות
- פגיעה בשורד נעשית על ידי כתיבת פקודה לא חוקית לאזור בו נמצא אותו שורד
  - כאשר מנוע המשחק ינסה להריץ את הפקודה הלא חוקית, השורד אשר עבורו הורצה הפקודה יפסל
- מנוע המשחק מריץ את השורדים בזירה הוירטואלית מספר רב של הרצות, בכל הרצה מחולק הניקוד בין השורדים שהגיעו לסוף ההרצה "חיים"

זירת המשחק

- 256 שורות- 00 עד FF
- 256 פיקסלים בשורה
- סה"כ 65,546 פיקסלים



	00	01			
00	0000	0001			
01	0100	0101			

שורד

- מטרת כל שורד להגיע לסוף הסיבוב, לשלב חלוקת הנקודות, ולמנוע משורדים אחרים לעשות זאת
- טקטיקה: לאורך השנים נכתבו שורדים רבים. ניתן לסווג אותם:
  - **התקפיים**: שורדים אשר ייעודם לכתוב כמות גדולה של פקודות לזיכרון המשחק במספר תורות מצומצם ככל שניתן, על מנת להביס את היריבים
  - **הגנתיים**: שורדים אשר ייעודם להתחמק מפגיעה של יריבים על ידי טכניקות שונות

קודגורו אקסטרים

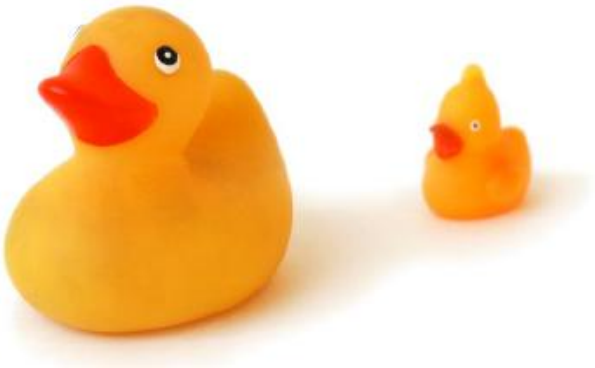
כתיבת שורד

קובץ שלד לכתיבת שורד

- נתבונן בקטע הקוד הבא. בשונה מקובץ שלד "רגיל" (base.asm) בקובץ זה:
  - מודל הזיכרון הוא tiny- מתאים לקובצי com
  - אין DATASEG
  - הגדרת org 100h, כפי שנדרש מקובץ com

```
1 IDEAL
2 MODEL tiny
3 CODESEG
4 org 100h
5 start:
6 ;----- sored code -----;
7 END start
```

כתיבת השורד הראשון



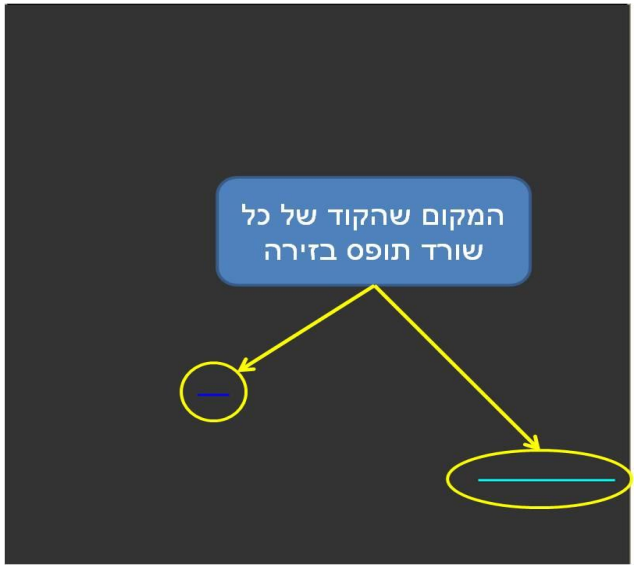
- נתבונן בקטע הקוד הבא

```
start:
        jmp start
```

- שורד זה יבצע פעולה אחת בכל תור, לאורך כל המשחק- הוא "יעמוד במקום"
- התוכלו לחשוב על יתרונותיו של שורד זה? על חסרונותיו?

מיקום התחלתי בזירה- כללים

- מיקומו ההתחלתי של כל שורד בזירה הוא אקראי ונקבע על ידי מנוע המשחק לפני כל סיבוב
- ניתן לראות את מיקום השורדים בזירה כ"קווים אופקיים" בזירה, אלו המקומות שהם תופסים בזיכרון
- מנוע המשחק דואג לכך ששני שורדים לא ייטענו למקום זהה בזיכרון
- מנוע המשחק מאפשר לכל שורד להיות באורך של עד 512 בתים



תפיסת מקום בזירה

- נבצע שינוי בשורד שלנו, שיגרום לו לתפוס את המקום המקסימלי בזירה

```
IDEAL
MODEL tiny
CODESEG
org 100h
start:
    jmp start
exit:
    db 512-(exit-start) dup (0cch)
END start
```

- start ו-exit מצביעים על מקומות בקוד: start על ההתחלה ו-exit על הסיום. ההפרש ביניהם הוא גודל הקוד. הפקודה האחרונה מחשבת כמה בתים צריך "לרפד" עד הגודל המקסימלי של 512 בתים ומקצה מקום זה בסוף השורד.

כתיבת שורד "תותח"

- עתה נרחיב מעט את יכולותיו של השורד:
  - נסו להבין מה יבצע כעת השורד?
  - האם נראה לכם שהוא ינצח את השורד הקודם?
  - איך אפשר לשפר אותו?

```
IDEAL
MODEL tiny
CODESEG
org 100h
start:
    mov cx, 0cccch
    mov bx,0
write_word:
    mov [bx],cx
    add bx, 2
    jmp write_word
exit:
END start
```

האצת שורד

- זוכרים שאמרנו שבכל תור מנוע המשחק מריץ פקודה אחת של כל שורד? יש דרך לקבל זמן מעבד רב יותר ☺
- לכל שורד מוגדר משתנה בשם Energy, שגודלו 16 ביט. הערך ההתחלתי שלו הוא 0
- אם הערך גדול מ-0, כל 5 תורות מנוע המשחק מחסיר מערכו 1 (אם הערך 0, נשאר 0)
- שורדים יכולים להגדיל את ערך האוגר ב-1 ע"י קריאה לאופקוד wait פעמיים <u>בדיוק</u>
  - מנוע המשחק מזהה שתי קריאות רצופות ל-wait ומריץ אותן בתור יחיד

```
cs:0005 9B          wait
cs:0006 9B          wait
```

האצת שורד- המשך

- בתחילת כל תור, אם האנרגיה שונה מאפס, "מהירות" השורד מחושבת ע"י הנוסחה

speed = log2(energy)+1

  - הערה: אם energy=0 אז speed=0
- הסיכוי ששורד יקבל תור נוסף הוא speed/16
- לדוגמה:
  - השקענו תור בהעלאת האנרגיה ב-1. כעת speed=1. למשך 5 התורות הבאים יש סיכוי של 1/16 לקבלת תור נוסף.
  - השקענו 2 תורות והעלינו את האנרגיה ל-2. כעת speed=2. למשך 5 התורות הבאים יש סיכוי של 1/8 לקבלת תור נוסף, למשך 5 התורות שאחריהם יש סיכוי של 1/16 לקבלת תור נוסף.

קודגורו אקסטרים

טקטיקה מתקדמת: שכפול שורד

שכפול שורד

- "**שכפול שורד**" היא אסטרטגיה נפוצה בתחרויות קודגורו אקסטרים
- בשיטה זו, שורד "משכפל" את עצמו מאזור זיכרון אחד לאזור זיכרון אחר
- לשיטה זו מספר יתרונות:
  - מיקום משתנה לאורך המשחק
  - השורד עלול "לדרוס" את יריביו במהלך ההעתקה
  - יצירת עותקים "מתים" רבים בזירה- מטרות דמה לפצצות חכמות
- **חשוב להדגיש**: בכל רגע נתון ישנו רק "מופע" חי אחד של השורד

שכפול שורד- חסרונות

- פיתוח השורד מסובך ונעשה עוד יותר קשה ככל שמורכבות השורד עולה
- יחס הזיכרון שנכתב לעומת יחס הפעולות אינו אופטימלי
  - במילים אחרות: השורד תופס מקום רב בזיכרון ולכן פגיע יותר
- נדרשת הבנה עמוקה ביותר של התוצר הבינארי

שכפול שורד

```
IDEAL
MODEL tiny
CODESEG
org 100h
start:
    push    cs
    pop     es
    xchg    di, ax
    add     di, 12
    mov     si, di
    add     si, 10
    std
myloop:
    ; This part is copied, then jumped to
    dec     di
    dec     di
    movsw
    movsw
    movsw
    movsw
    movsw
    movsw
    inc     di
    inc     di
    jmp     di
END start
```

- ראשית נכיר את הפעולה movsw: מעתיקה מילה מכתובת ds:si לכתובת es:di, ערכי הרגיסטרים מתעדכנים בהתאם ל-direction flag
- הקוד הבינארי שנוצר מכיל 12 בתים לפני התווית myloop ו-12 בתים אחריה
- 6 פקודות ה-movsw יעתיקו את 12 הבתים שאחרי myloop
- מכיוון שהאוגר di מתעדכן בהתאם, כל שנותר הוא לבצע jmp למיקום החדש

שכפול שורד- הסבר

```
IDEAL
MODEL tiny
CODESEG
org 100h
start:
    push    cs
    pop     es
    xchg    di, ax
    add     di, 12
    mov     si, di
    add     si, 10
    std
myloop:
    ; This part is copied, then jumped to
    dec     di
    dec     di
    movsw
    movsw
    movsw
    movsw
    movsw
    movsw
    inc     di
    inc     di
    jmp     di
END start
```

כיוון ההעתקה

לפני פקודת ההעתקה הראשונה, di מצביע על שורה זו

לפני הקפיצה, "נתקן" את di כך שיצביע על פקודת ה-dec הראשונה

לפני פקודת ההעתקה הראשונה, si מצביע על שורה זו

קודגורו אקסטרים

טקטיקה מתקדמת: כתיבה בקצב כפול

Far Call

- הרעיון: בשפת אסמבלי, כאשר אנחנו קוראים לפרוצדורה בעזרת פקודת call, אנו דוחפים למחסנית 2 **בתים** שמציינים את מיקומנו טרם הקריאה (האופסט מתחילת הסגמנט)
- אם ביצענו קריאת far לפרוצדורה, נדחפים למחסנית הן האופסט והן הסגמנט, סך הכל **4 בתים**
- ... במידה ונחליט שזירת המתמודדים היא המחסנית שלנו, נוכל לדרוס ארבעה בתים בכל תור במקום שניים ☺
  - מהם חסרונות השיטה הזו?

דוגמה: שימוש ב-Far Call

```
1  IDEAL
2  MODEL tiny
3  CODESEG
4  org 100h
5  start:
6      add ax,call_far-start
7      push es
8      pop ds
9      mov bx,50h
10     mov [bx+2],cs
11     mov [bx],ax
12     push cs
13     pop ss
14     mov sp,ax
15 call_far:
16     call [dword bx]
17 END start
```

❖ מקור הקוד ומידע נוסף:

http://goo.gl/0zvLgU

(מומלץ לגלוש מדפדפן כרום)

בהצלחה!


גברהים
חינוך • מצוינות • סייבר


התחרות החדשה
מבית קוד גורו
אקסטרים
X
Code Guru
EXTREME

קודגורו אקסטרים

מנוע, זירה, שורד