

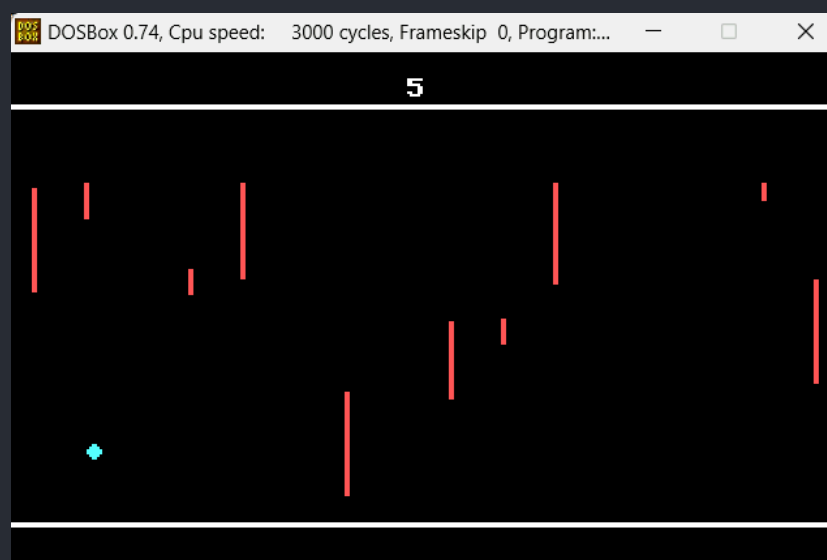
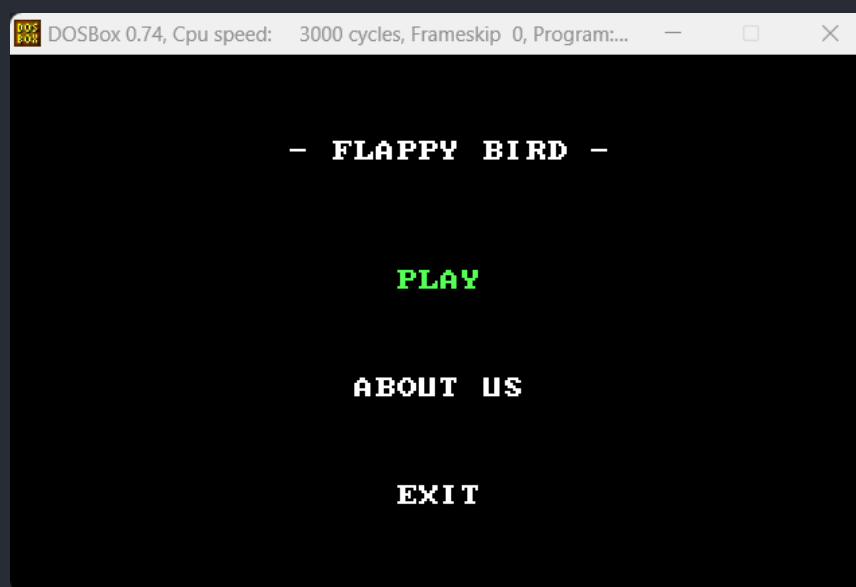
پروژه نهایی ریزپردازنده و مدارهای واسط - پاییز 1402

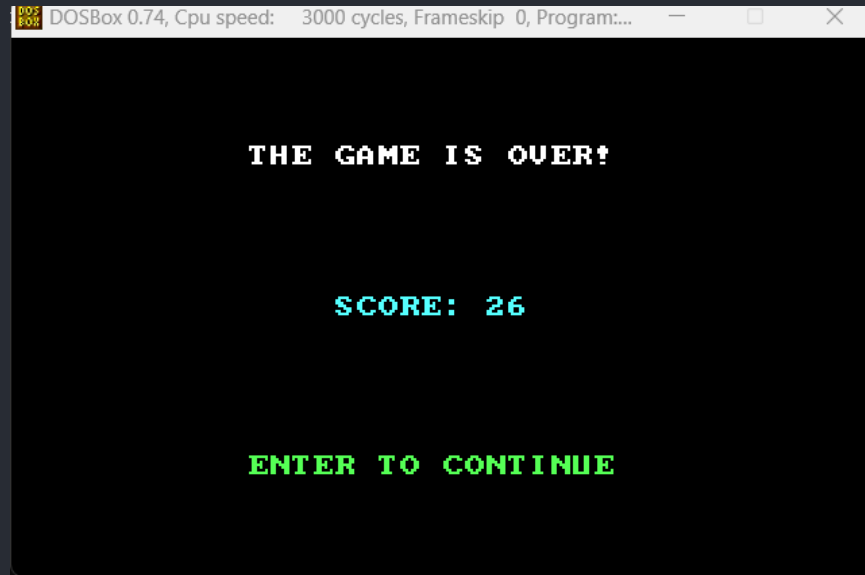
استاد: دکتر عطار زاده

اعضای گروه: شایان کبریتی - عرفان عابدی

شرح کلیات پیاده سازی بازی (DOS 8086):

گرافیک بازی و پیش نمایش آن:





پیاده سازی فیزیک بازی:

کد بازی شامل دو پیاده سازی متفاوت (bf_realistic.asm و bf.asm) می باشد. که یکی پیاده سازی بازی با مقیاس های ساختگی است که گیم پلی جذاب تری دارد. از آن طرف دیگری دارای مقیاس واقعی تری از لحاظ نیرو های وارده به توپ (پرنده) است.

برای مثال،

در پیاده سازی realistic، اثر نیروی گرانش 10 پیکسل بر مجذور ثانیه در نظر گرفته شده است.

حال به واسطه متغیر FPS_CONTROLLER در برنامه، frame rate بازی 100FPS ست می‌شود. بنابراین هر 100 فریم معادل 1 ثانیه است و در هر 1 ثانیه بایستی سرعت توپ را 10 پیکسل کاهش بدهیم. که معادل این است که هر 10 فریم 1 پیکسل از سرعت توپ کم شود. همین معادلات برای نیروی بالارونده (LIFTING_FORCE) توپ نیز صادق است.

در پیاده سازی عادی، این سرعت هر فریم 1 پیکسل کاهش پیدا میکند.

نحوه محاسبه امتیاز:

بعد از هر SCORE_INCREMENT_CONDITION فریم، به امتیاز یکی افزوده می‌شود. این مقدار می‌تواند دستی تغییر بکند و یا در procedure مربوط به INIT_GAME با سرعت مانع ها (BARRIERS_SPEED) سینک شود.

نحوه محاسبه اعداد تصادفی (مربوط به مختصات و اندازه موانع):

برای اینکار از الگوریتم linear congruential generator که به LCG معروف است استفاده می‌کنیم. این الگوریتم دارای پارامترهایی است که با تعیین آن ها میزان شانسی بودن و duration اعداد رندوم متفاوت می‌شود. سعی شده بهترین عدد برای این بخش استفاده شود. این الگوریتم یک SEED و یا عدد رندوم اولیه برای تولید باقی اعداد رندوم دارد. برای بهبود رندوم بودن موانع بازی، این عدد در ابتدای هر بازی برابر تایم فعلی سیستم ست می‌شود.

The generator is defined by the **recurrence relation**:

$$X_{n+1} = (aX_n + c) \bmod m$$

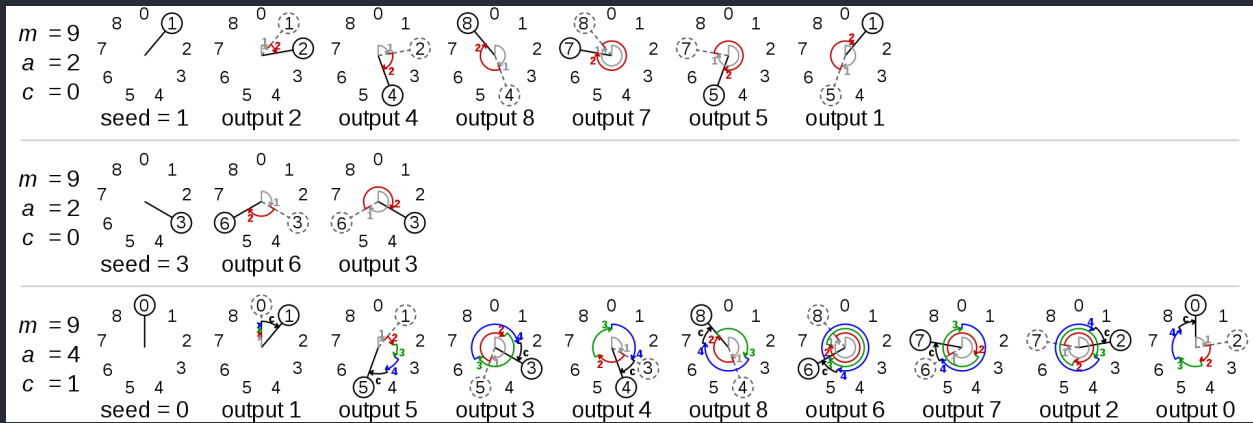
where X is the **sequence** of pseudo-random values,

$m, 0 < m$ — the "**modulus**"

$a, 0 < a < m$ — the "**multiplier**"

$c, 0 \leq c < m$ — the "**increment**"

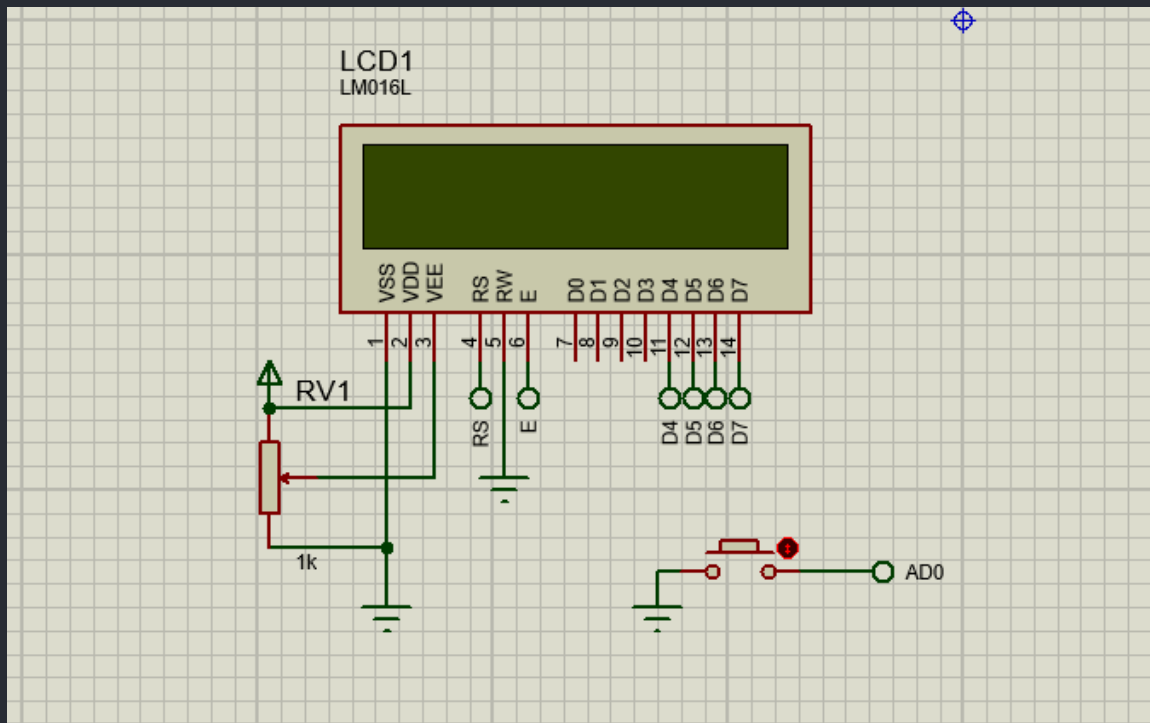
$X_0, 0 \leq X_0 < m$ — the "**seed**" or "**start value**"



شبیه سازی کنترلر:

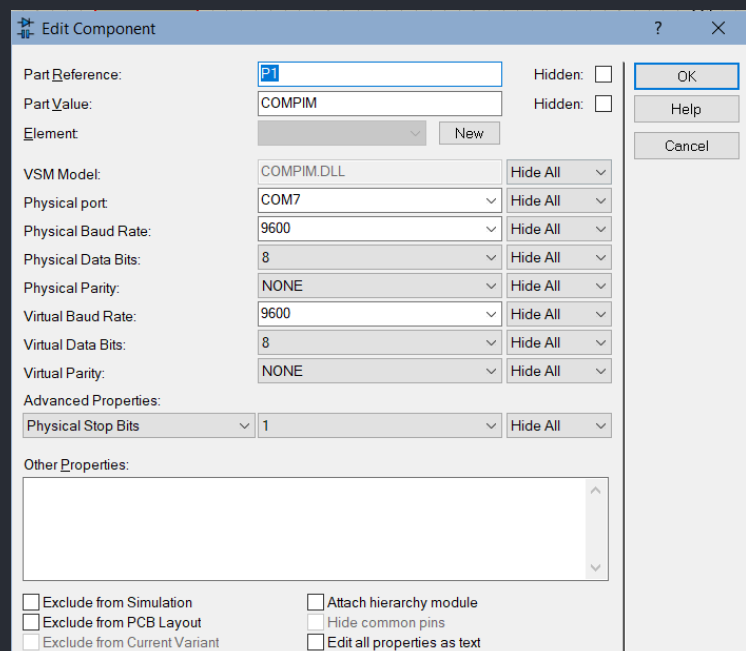
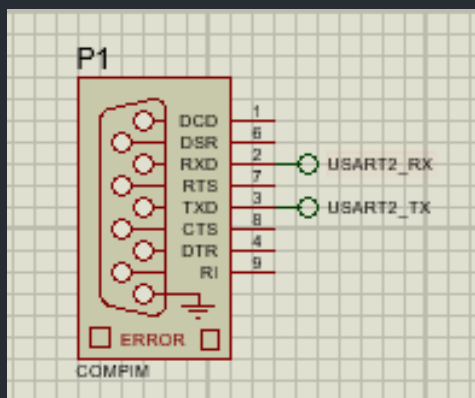
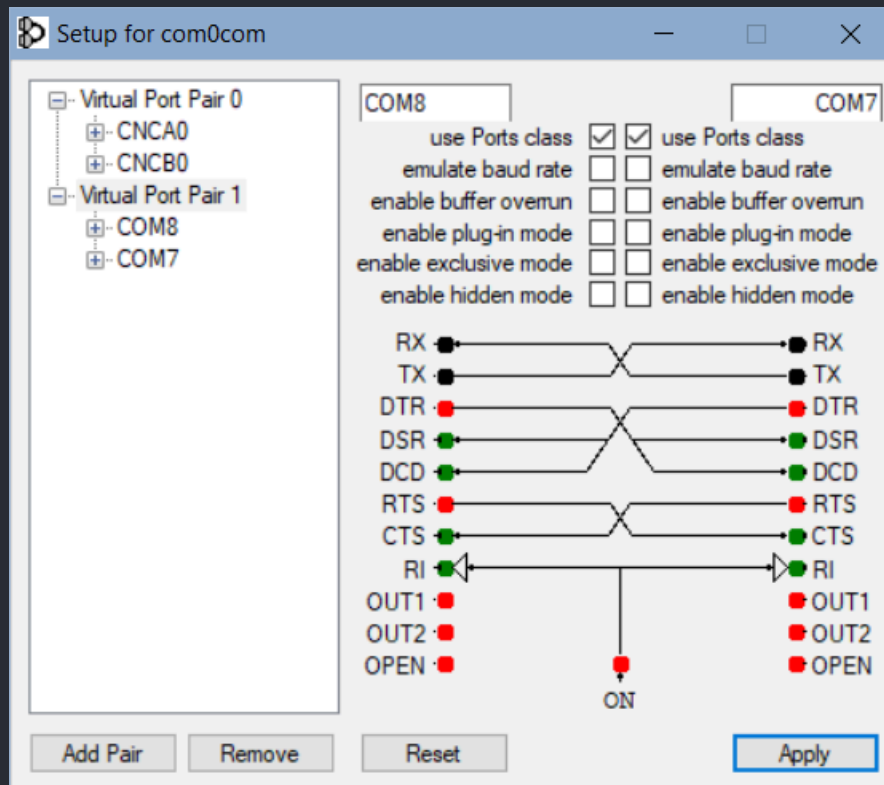
با فشردن دکمه فرمان 'J' از طریق UART و پورت مجازی به بازی منتقل میشود.

همچنین با تغییر امتیاز در بازی فرمان 'P' ارسال میشود و یک امتیاز به امتیاز های کاربر اضافه میشود و در LCD نمایش داده میشود. با پایان بازی فرمان 'O' ارسال میشود و 'GAME OVER' برای کاربر نمایش داده میشود.



تعریف پورت مجازی جهت اتصال شبیه سازی به بازی:

با استفاده از نرم افزار com0com دو پورت مجازی را به هم متصل میکنیم. و در شبیه سازی از قطعه COMPIM استفاده میکنیم و آن را به شکل زیر تنظیم میکنیم.



دریافت و ارسال فرمان در اسمبلی:

INT 14 - AH = 01h SERIAL I/O - TRANSMIT CHARACTER

AL = character

DX = port number (0-3)

Return: AX = port status (see AH = 00h above)

INT 14 - AH = 02h SERIAL I/O - RECEIVE CHARACTER

DX = port number (0-3)

Return: AL = character received

AH = RS-232 status code (see AH = 00h above)

```

TRANSMIT_STATE PROC NEAR
    MOV AH, 00H ; INITIALIZE USART
    MOV AL, E3H ; BAUD RATE 9600, PARITY 0, STOP 1, WORD BITS LENGTH 8
    INT 14H

    MOV DX, 0 ; Select Serial1

    CMP GAME_OVER, 1
    JNE GAME_NOT_OVER_FLAG
    MOV AL, 'O'
    MOV AH, 01H ; TRANSMIT CHARACTER for GAME OVER
    INT 14H
    RET

    GAME_NOT_OVER_FLAG:
    MOV AL, 'P'
    MOV AH, 01H ; TRANSMIT CHARACTER for PLAY
    INT 14H
    RET
TRANSMIT_STATE ENDP

```

```

RECEIVE_JUMP PROC NEAR
    MOV AH, 00H ; INITIALIZE USART
    MOV AL, E3H ; BAUD RATE 9600, PARITY 0, STOP 1, WORD BITS LENGTH 8
    INT 14H

    RECIEVE:
    MOV DX, 0 ; Select Serial1
    MOV AH, 02H ; RECEIVE CHARACTER
    INT 14H

    RET
REVEIVE_JUMP ENDP

```

با استفاده از INT 14H ارتباط سریال ایجاد میکنیم. همچنین قبل از اجرای بازی با اجرای دستور زیر

`serial1=directserial realport:com7`

پورت سریال را در DOSBOX تنظیم میکنیم.

اتصال مجازی پورت ها:

