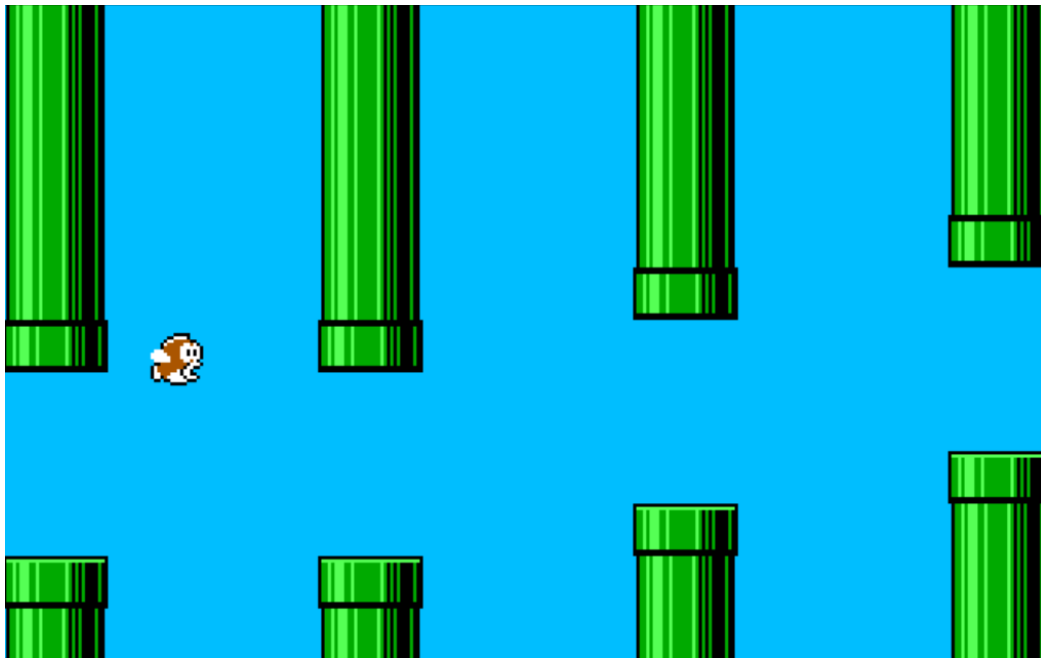


Flappy Bird

מגיש: נועם שטיין
כיתה: יא5
בית ספר: תיכון אלון
מורה: עודי רז וגיל ליברזון
שנת לימודים: 2025



מבוא – לעבודת האסמבלי הכנתי העתק של FlappyBird.

חוקי המשחק – שולטים בציפור שקופצת, נופלת ומתקדמת ימינה עם המצלמה במטרה לא לפגוע בצינורות שיהרגו את הציפור. הצינורות ממשיכים לבוא לנצח כל פעם בגובה שונה וצריכים לשרוד הכי הרבה זמן שאפשר.

הוראות הפעלה ומדריך למשתמש – כדי להפעיל את המשחק צריכים לשים את קובץ Sprites.bin לייד main.exe שמייצרים עם assembler. משחקים עם מקש space וכדי לצאת מהמשחק משתמשים במקש .escape.

תיאור אלגוריתמים מרכזיים:

טעון מקובץ Sprites.bin מידע המכיל תמונות 16X16 אחת אחרי השנייה בformat של DOS שמצוירים עם כמה פרוצדורות שמשתמשות בלולאות.

כדי לדעת מה מצב המשחק מתוך שלושה (מצב כבוי שמחכים למקש space), (מצב משחק שהמשחק פועל) ו(מצב של האנימציה שבה הציפור נופלת לאחר שמתה) יש משתנה סטטי שמחזיק fn-ptr לפרוצדורת update של כל מצב.

יש משתנים סטטיים שמחזיקים את מיקום השחקן והצינורות שלפי מצב המשחק מעודכנים ומאותחלים, ולפיהם מצוירים הציפור והצינורות עם API של הציר.

מזהים שהציפור פוגעת בצינורות לפי הפיקסלים על המסך מכיוון שאין עוד שום דבר כמו רקע וזה מהיר.

פרוצדורות \ מקרויים (לפעמים מהר יותר להשתמש במקרו ולפעמים מהר יותר פרוצדורה):

* יש הרבה פונקציות עזר ולא חייבים לקרוא אליהן.

ציור:

load_sprites – טוען את התמונות מSprites.bin ונדרש כדי שציור יעבוד אחר כך.

clear_screen – מוחק את המסך לצבע הניתן.

set_draw_x, set_draw_y – משנה את מיקום הציור (פינה שמאלית עליונה), שנשמר בax, bx.

set_draw_width, set_draw_height – משנה את אורך הציור (משומש כדי לא לצייר 16X16 כדי למחוק כשצריך רק טור אחד או שניים, בלי זה הציור לא מהיר מספיק ויותר flicker) שנשמר בcx, dx.

move_draw_right, move_draw_left, move_draw_up, move_draw_down – מזיזים את מיקום הציור בכיוון המתאים בכמות הניתנת כקלט (macro).

set_draw_size – מקרו שמשנה את sprite-index מSprites.bin לציור הבא, נשמר בsi.

draw_sprite, clear_rect – משתמשים בנתונים שהכניסו כדי לצייר \ למחוק מהמסך.

draw_sprite_pushed, clear_rect_pushed – קוראים לפונקציות המקוריות אבל שומרים על הערכים שהיו בכל מיני רגיסטרים (מידע על שימור רגיסטרים כתוב לפני כל פונקציה כמו contract).

פונקציות עזר לציור:

set_draw_di – מאתחל את di לערך שלו לפי מיקום הציור.

draw_sprite_row, clear_rect_row – מציירים שורה ומקדמים את di, si לשורה הבאה.

clip_draw_rect – מצמצם את rect הציור לריבוע המסך ולפי זאת משנה את המיקום, את הגודל, ואת המיקום ההתחלתי בsprite (כלומר ax, bx, cx, dx, si).

קלט שחקן (מקשים):

detect_key – מקרו פשוט מאוד שבודק האם מקש לחוץ בצורה הכי פשוטה שאפשר לפי makecode.

שחקן:

update_player – עושה את כל עדכון השחקן כולל collision-checks.

פונקציות עזר לשחקן:

set_player_draw_rect – מאתחל את מיקום הציור למיקום השחקן.

player_jump_check – משנה את מהירות השחקן (משתנה סטטי) אם נלחץ space.

move_player – מעדכן את מהירות ומיקום השחקן לפי חוקי ניוטון ונגזרות.

jmp_kill_player – נועד לשימוש עם jmp ולא call, הורג את השחקן.

player_out_of_bounds_check – הורג את השחקן אם הוא מחוץ לגובה המסך.

player_collision_check – הורג את השחקן אם הוא פגע בצינור.

draw_player_anim – מצייר את השחקן לפי מודד זמן לאנימציה שלו.

צינורות:

init_pipepairs – מאתחל את הצינורות בצורה אקראית.

draw_pipes – מצייר את הצינורות מבלי לעדכן אותם (כי הם לא תמיד זזים).

update_pipes – מזיז ומצייר את הצינורות.

פונקציות עזר לצינורות:

pipepair_loop – מכין loop-label שעובר על כל הצינורות (index שמור בdx).

set_pipepair_si – מאתחל את si לbyte-offset למערכי pipe-state שכולם שומרים שני בתים לכל pipepair.

init_pipepair_xpos, init_pipepair_ypos – מאתחלים את המיקום של צינור.

set_pipepair_xdrawpos, set_bottom_pipe_drawpos, set_top_pipe_drawpos – מאתחלים את מיקום הצינור לצינורות.

set_bottom_pipe_height_cx, set_top_pipe_height_cx – מאתחל את cx לגובה של הצינור העליון \ תחתון.

draw_pipe_row – מצייר שורה אחת (tile אחד) של צינור ומוחק מהמיקום שבו הצינור היה אמור להיות בקודם.
frame.

draw_bottom_pipe, draw_top_pipe – מצייר צינור עליון \ תחתון.

מצב משחק:

jmp_gameloop_gameplay, jmp_gameloop_wait, jmp_gameloop_death – משתנים את מצב המשחק למתאים.

gameloop_gameplay_update, gameloop_wait_update, gameloop_death_update – מעדכנים את כל מצב משחק.

הרצה:

initialize, init_video_mode, init_datasegment_ds – מאתחלים.

wait_milliseconds, update – מעדכנים.

clean_up – מסיים ומחזיר לtext-mode.

Main – מי יודע.

משתנים:

SpritesFileName, SpritesFileHandle – משומשים זמנית כדי לטעון את Sprites.bin.
SpritesBuf – מידע התמונות.

GameLoopUpdateFn – מצביע על פונקציית העדכון לפי מצב המשחק.

PlayerYPos, PlayerAnimTime, PlayerYVelocity – מייצגים את מצב השחקן.

PipePairXPoses, PipePairBottomHeights – מערכים שמייצגים את המיקומים של כל זוג צינורות.

* כל מידע שאין פה על פונקציות או משתנים אפשר לראות בקוד עצמו שמובן מאוד.

תוכנות:

VSCode

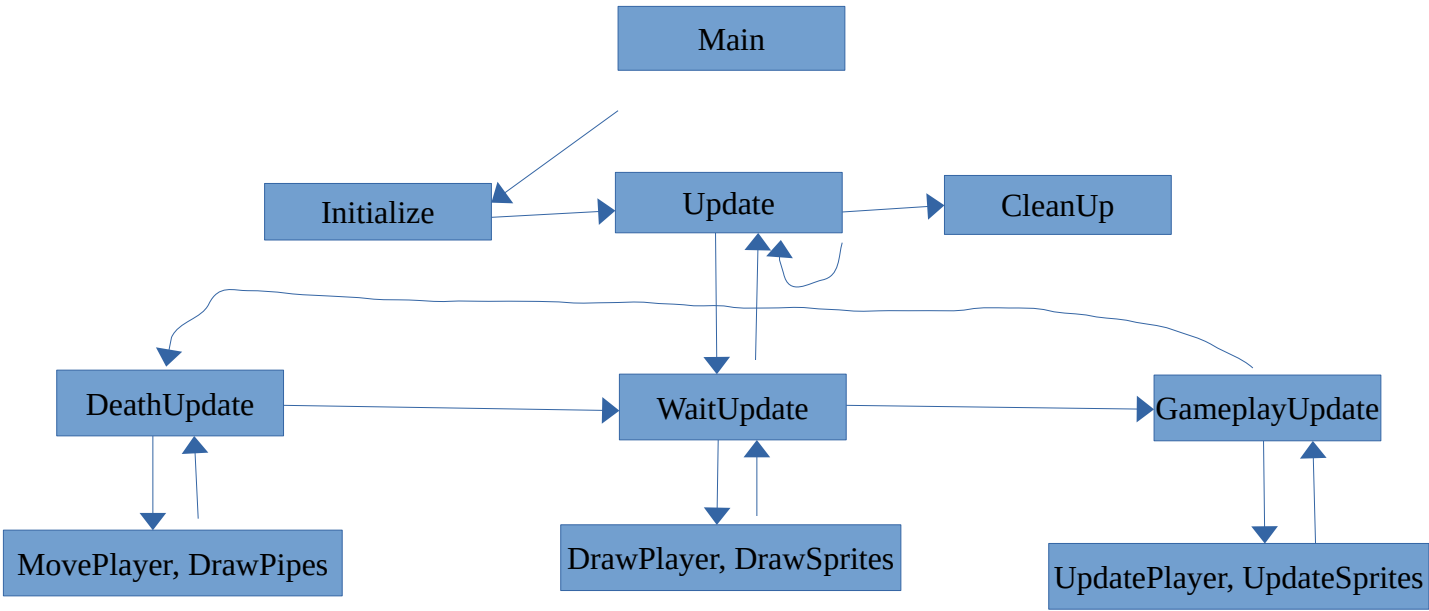
MasmTasm – זה vscode-ext שנותן errors יפים בקוד ומריץ את הקוד עם emulator ו assembler שאני בוחר.

DosBox, TASM

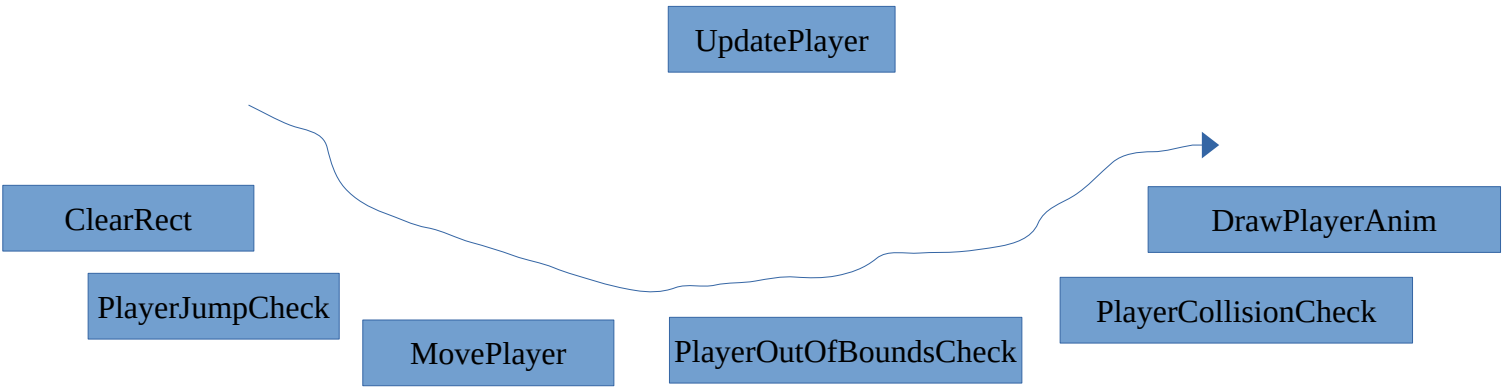
מקורות עזר:

<https://www.felixcloutier.com/x86/dec>

Flow Chart



Player Flow Chart



Sprite Flow Chart

