Mega Australian Documentation

This document will present and explain the development environment for the fan made game “Mega Australian”, including explaining the code’s procedures, labels, variables, naming conventions, flow chart, file conventions and the graphic design.

Code Structure and Flow Chart

The code is currently entirely in one file; “bas.asm”. In the future, the code will be divided between files in a structure to be determined.

1. **bas.asm Structure:**
    a. DATASEG:
    Contains all of the games variables/
        i. Saxton - Variables used to control and calculate Saxton related actions (and the payload is also there, will be moved later).
        ii. Sentry - Variables regarding Saxton’s mini-sentry/
        iii. General Enemy - Variables used to act out the enemy procedure for each enemy.
        iv. EnemyX - Variables used to store information on enemies 1-3.
        v. Friendly Bullets - Arrays describing bullets fired by Saxton.
        vi. Hostile Bullets - Arrays describing bullets fired by enemies and to check if an airblast is happening.
        vii. General Bullet - Used to act out the bullet procedures for each bullet.
        viii. Bullet Timers - Variables to check firing cooldowns (could be split to character’s variables accordingly.
        ix. Misc. Enemy - Two variables related to sentry buster’s running animation (should use BulEnTimer instead) and a redundant variable.
        x. Stage Check - A variable to track stage completion and some moving stage objects variables.
        xi. General variables - :(… First six variables are used for gravity and BasicReset calculations. Followed by a list of variables used to open files and print pictures.
        xii. Clock - Variables to keep track of time.
    b. CODESEG before “start” label:
    All of the app’s procedures.
        i. Sada -
    c. CODESEG after “start” label:
    The game levels and acts
        i. Start0 - RealStart: Plays the intro and waits to receive an “enter” press.
        ii. RealStart - Start1: Plays the stage select screen.
        iii. Start1: Scout stage.
        iv. Start2: Soldier stage.
        v. Start3: Pyro stage.
        vi. Start4: Demoman stage.
        vii. Start5: Heavy stage.
        viii. Start6: Engineer stage.
        ix. Start8: Sniper stage.
        x. Start9: Spy stage.
        xi. StartBoss: Boss levels.
        xii. GameOver: Game over screen.

2. **Level and Room structure:**
    a. StartX - The start of the level. Sets ‘es’ to 40h, sets the initial ground level and places Saxton in the correct location.
    b. Before RoomXYLoop - Sets initial enemy types, placement, health, and other stats.
    c. RoomXYLoop -
        i. TickXY - Label, comparison and jump to create a wait loop for a constant fps.
        ii. DefaultRoom procedure - acts out the actions that happen universally every tick.
        iii. Body of the room - In most rooms, comparisons to check Saxton’s location and adjust the ground level. In some rooms adjustments of room related objects such as disappearing blocks and platforms.
        iv. End - Ends with comparisons to check whether the room’s objectives were met and whether Saxton’s in the correct location to go to the next room.
    d. After EndRoomXY - Set up the next room.
    e. RoomXB - The boss room. Some boss rooms have the boss’s AI in the room loop itself, this will be changed in the future.
    f. After EndRoomXB - Play the stage end screen.

3. **Tick structure:**
    a. DefaultRoom - First procedure that happens every tick in every room
        i. Prints the room background.
        ii. Acts out Saxton's move according to player input and other variables.
        iii. Acts out the Mini-Sentry's move.
        iv. Acts out the enemies' moves.
        v. Acts out both the friendly and hostile bullets.
        vi. Checks if Saxton is dead, if so return to stage select and reduce a life.

Variables

To be documented….

Procedures

1. Basic Reset -
   “Resets” a character’s position if its movement causes it to be inside a wall/floor/ceiling.
   [x] and [y] need to be set according to the character’s variables. They will be compared to the ground, ceiling, and walls that are determined by Saxton’s position.
   The procedure is called during Saxton’s moving procedure and during enemies’ tick procedures.
   Will later be changed to act according to an open “MAPX-X.txt” file.
2. HealthBar -
   Prints a Health bar using the location in [HPX] and [HPY], the health in [HP], and in the color inserted in “dl”.
   The procedure is called to print Saxton’s health bar and special ammo bar if he is in a special mode, and to print the bosses’ HP.
   a. BossHealthBar.
   b. WeaponHealthBar.
3. Gravity -
   Moves a character according to their vertical velocity ([VerVal]), accelerates them towards the ground, and checks if they hit the ground. [JCheck] is there to not allow characters to jump while mid-air.
4. ClearKeyboard buffer -
   Clears the keyboard buffer so that each tick gets its own input.
5. PrintBlock -
   Prints a 10pxX10px block in the [x],[y] position colored by [color].
   Is used in certain levels to print a non permanent block.
6. CapturePoint -
   Used to print red on top of a capture point to show it is capped, each activation covers a pixel length of the point.
   Need to insert position to capture through [x] and [y], and color (usually red) through [color].
   Cap1StageL is for the lower part of the point (the one on the ground) and Cap1StageH is for the higher part (the hologram).
   Is called during stages with capture points.
7. PushLoad -
   Moves the payload when Saxton touches it.
   Is called during stages with a payload.
8. CleanBullets -
   Clears all hostile and friendly bullets.
   Is called after every room.
9. CleanEnemies -
   Clears all enemies.
   Is called after every room.
10. PreRoomReset -
    Resets all of the stats before each stage. Should be renamed to PreStageReset.
11. DefaultRoom -
    Set of actions that happen every tick. Starts by printing the background.

12. OpenFile -
    Opens files
13. ReadHeader -
    Reads the header of the files (doesn't do anything with it currently).
14. ReadPalette -
    Reads the palette of the BMP files (doesn't do anything with it currently).
15. CopyPal -
    Copies the palette and changes the game's palette to match.
    Is called when switching Saxton's mode.
16. CopyBitmap -
    Prints out a BMP file, gets the picture width and length through [PicX] and [PicY] and the picture's position through [x] and [y].
    Does not print pixels in the "0" color (transparent background).
17. CopyBKGBitmap -
    Prints out the background. Skips the "255" color (I don't remember why but this can be fun).
18. CloseFile -
    Closes the file.
    Any time a picture is to be printed, procedures 12-18 are to be called in succession with only one of 15-17 being called according to the requested action.
    Other file types will be given their own ReadHeader and other procedures in the future.
19. Print0Tower -
    Helps play the opening scene.
20. Enemy Logic -
    All enemy logic procedures get their enemy's stats through [EnX], [EnY], [EnJCheck],[EnVerVal], [EnLastPose], [EnWidth], and [EnHeight] (all of these can and should be replaced in the future by their respective non-En variables).
    Most enemy logic procedures skip most of their body when Saxton is invisible.
    a. SentryBuster -
       Checks if Saxton is to the right or to the left and runs accordingly.
       Uses [SBCheck] and [SBCheck2] to pick the running frame to print out (can be replaced by one [BulEnTimer]).
       Checks if Saxton touches the Sentry Buster and explodes if true.
       Calls Gravity and BasicReset to act on the Sentry Buster.
       This procedure is very inefficient, and will be fixed in the future.
    b. HSentry -
       Does not move. Shoots a double bullet every 25 ticks in the direction matching the Sentry's direction.
    c. Yeti -
       Every 25 ticks does a dash between his two walls defined by [EnJCheck] and [VerVal]. While dashing checks to see if it hits Saxton and damages accordingly. Does not change [EnY] location.
       Maybe I'll change it to a Gibus Ghost and make the yeti a different enemy.
    d. Bombonomicon -
       Every 5 ticks rises 2 pixels and changes its charging sprite. On the 40th shoots 2 rockets.

e. Spy9B -
   The logic for the Spy boss on stage 9.
f. Demo4B -
   The logic for the Demoman boss on stage 4.

21. FriendlyBullets -
    Goes over the array of friendly bullets, for every bullet:
    a. Checks type (type 0 means no bullet).
    b. Checks if should be erased (reached the ends of the screen).
    c. According to its type, checks if it hits any enemy.
    d. Moves the bullet.
22. HostileBullets -
    Same as FriendlyBullets, but checks to hit Saxton.
23. AddFriendlyBul -
    Adds a friendly bullet to the first empty bullet spot (first spot where the type is 0).
    Gets bullet's stats from [AddBulType], [AddBulX], [AddBulY], and [AddBulRight].
24. AddHostileBul -
    Same but for hostile bullets.
25. Enemies -
    For enemy 1-3, checks if alive and calls the correct enemy logic according to its type.
26. Sentry -
    The logic for Saxton's mini-sentry.
    Drops the mini sentry to the floor and shoots every 15 ticks.
27. Move -
    Gets user input through "al" and performs the following actions:
    a. Checks if the user switched "modes" and changes the palette if so.
    b. Prints the relevant ammo bar.
    c. Runs Saxton's fire damage cycle and prints a fire sprite behind him.
    d. Checks invisibility cycle.
    e. Processes a down-key input (Slide/Rocket Jump/Jarate).
    f. Processes a space-bar input (Shooting)
    g. Processes the shooting cooldown.
    h. Processes an up-key input (Jump)
    i. Processes a left/right-key input or the releasing of one of them.
    j. Processes an ESC-key press (return to Stage Select minus 1 life).
    k. Prints Saxton in updated location.
    l. Prints Saxton's health bar.

Label Naming Conventions

To be documented….

File Types Conventions

1. **ASM files -**
   Assembly code for 8086 processors.
2. **BMP files -**
   All should have the same (or similar) palette.
   a. BKGSS(X) and **palettes** - These files hold the color palette for the game. The color palette is changed whenever Saxton changes his “mode”.
      The color palettes are identical except for the final color which is different in each palette.
      The changing color is used in Saxton’s sprite to change the color of his hat and boots between modes, and in the special ammo bar.
      The palette’s “0” color will not be printed in any picture, it is used as an empty background.
   b. BKGX-Y - The background for all the rooms in the game. If a certain color is to be added, it should be added to all “BKGSS” files’ palette in the same location (unless you want a stage that is complete darkness when using a certain mode which is also cool I think I’ll make a boss stage like that). All BKG files are 320 pixels wide and 200 pixels long.
   c. All other pictures - Every other picture should be 21 pixels wide and 24 pixels long (will be changed in the future) with extra space being filled out with the “0” color.\
3. **TXT files -**
   a. MAPX-X - In the future, will be used to check collisions. Starting from the top left and going right, a “1” bit signifies a collision pixel and a “0” bit signifies a pixel that can be walked through.