Assembly Game

# Overview

We’re going to make a game in assembly! Your game needs to hit these requirements.

* Input: Your game needs to check for some user input from the player.
* Title Screen: There should be a title screen before loading into the main game. You will need to get some sort of input to get from the title screen to the main game loop.
* Collision: You need to have some sort of collision detection.
* Physics: There needs to be a physics update to a game entity that includes acceleration (a change in velocity, not just a constant speed. This can be aesthetic and not mechanical)
* Double Buffering: Use double buffering.
* Table Driven Methods (including 7-segment LED): This doesn’t have to be just the 7 segment or a sin/cos table, but some way in which you are using tables to drive your functionality.
* Bitmap: Draw a bitmap and handle an object moving over it (redraw the bitmap behind it)
* Randomness: There should be a degree of randomness within your game. This can be visual and not necessarily in the game’s mechanics (code for random number generator will be provided)
* Fixed Point Math: Your code should use and convey and understanding of fixed-point math.
* Audio: Your game should have some sound effects
* Input/Update/Render Loop: your game should have a game loop that has an input, update, and render. Refer to Realtime Programming Fundamentals lecture.

You will need to come up with a game proposal that fulfills these requirements. Where relevant, convey how your game achieves the above in the proposal.

GAME IDEA: Minesweeper

* Input:
  + When player clicks on tile, tile will change to reveal bombs/number of bombs nearby (left click?)
  + Player can place flag on tiles (right click?)
* Title Screen:
  + Splash screen with logo
  + Start button
  + How to play button
* Collision:
  + Player will click blank tile to reveal number of bombs surrounding that tile (or tiles that have bombs)
  + Player can set flags on tiles that they think have bombs
  + Main menu buttons can be clicked
* Physics:
  + Aesthetic most likely
  + Bomb that bounces and explodes if player clicks a bomb tile
  + Some sort of animation for when player wins?
* Double Buffering:
  + Can be used for loading bitmaps
* Table Driven Methods:
  + 7 segment hex
  + Could be used to calculate tile values based on bombs nearby?
* Bitmap:
  + Splash screen
  + Menu buttons
  + Count for number of bombs left (could use 7 segment display for this)
  + Tiles
  + Bomb numbers
  + Bombs
  + Guy at the top
  + Flags
* Randomness:
  + Depending on difficulty, either pulling from a random set of possible bomb maps, or making bomb placement completely random (could have a difficulty option?)
* Fixed Point Math:
  + Could calculate percent of bombs found
* Audio:
  + Lose/bomb sound effect
  + Win sound effect
  + Sound effect for clicking tiles/buttons
  + Background music
* Input/Update/Render:
  + Game main states will be loading-->main menu-->game play
  + Game play will check if player has clicked a bomb and if player has found all bombs
  + Game will check if player is clicking a tile