Wee Dig Dug: Proposed Design

Anand Balakrishnan
anandbal@buffalo.edu

Amrit Singh
asingh42@buffalo.edu

April 11, 2017

Contents

1 Overall Flowsheet

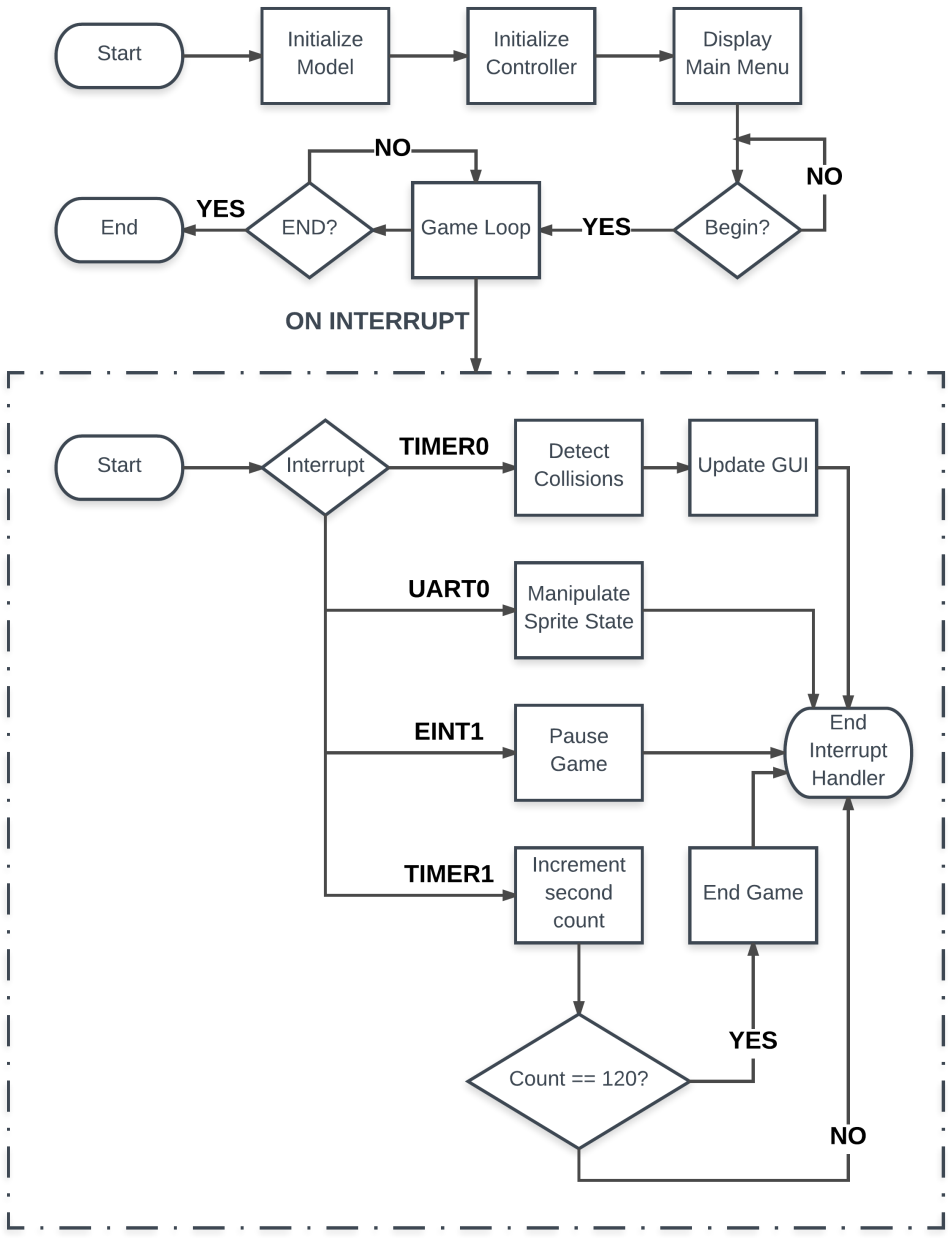

Figure 1: Overall Flowsheet

2 Model: model.s

Maintains the internal representation of the whole game.

2.1 Uses

The model will maintain the following information:

1. Position and state of all sprites on board
2. Position of all the sand using an array
3. Score/Level

2.2 Subroutines

The model contains subroutines that will:

- Initialize model (board and sprites)
- Manipulate model
  - Reset model
  - Remove sprite
  - Change sprite direction
  - Remove sand and change score

2.3 Design

The model consists of:

1. The board: 40 X 64 array of ”blocks”.
   - The board is a 40 X 64 byte array of blocks, each byte representing a boolean for sand, i.e, 1 = sand and 0 = no sand.
   - The array is created by reserving 40*64 = 2560 bytes of space in static memory by using the SPACE or FILL directives.
   - **NOTE:** The size of the array can be changed to make the game more space efficient. This can be addressed in later versions of the game
   - **NOTE:** The size of the board can be changed also, as the final game should work regardless of size.
2. The sprites:
   - Each type of sprite (Dug, his pump, Pookas and Fygars) maintains a position (the top left corner in GUI) and a state, (direction of movement, velocity, DEAD or not).
   - The character sprites (Dug, the Pookas and the Fygars) are each of size 4 X 4 blocks (hence occupying 16 blocks).
   - Dug’s pump is a sprite of height 1 block and variable length. This sprite has an additional state variable to hold length. The length cannot exceed 4 blocks (**NOTE:** To be revised).

3 GUI: gui.s

The GUI reads the model and displays it.

3.1 Function

The GUI is responsible for the following:

- Maintain the state of the screen, i.e., hold representation for Main Menu and Game
- Update itself as and when model is updated.
- Maintain a accurate representation of the model.

3.2 Subroutines

The GUI file will have subroutines that will:

- Draw GUI
- Update GUI

3.3 Design

1. Sand:
   - Each block in the model represents one block of sand in the GUI.
   - The sand is 4 X 4 ”pixels” in the GUI.
2. Character Sprites:
   - Each character sprite is 16 X 16 ”pixels” in the GUI.
3. Draw/Update:
   - The draw and update subroutines will print all variables in the model based on the pre-defined size of the sprites.

4 Controller: controller.s

The controller is the module that will control both, the GUI and the model. It will mainly contain:

- The interrupt handlers for user input and timer.
- Collision detection subroutine
- Update sprite positions subroutine
- Generate pump subroutine.

4.1 Design:

1. For user input:
   - Use FIQ Interrupts to handle user input/keystrokes, as implemented in Lab 6.
2. For game update:
   - On timer interrupt, the controller has to perform collision detection and handling, and update position of sprites.

4.1.1 Detecting collisions:

1. Read coordinate of each sprite in the model.
2. For each sprite, do the following:
   - For the Dug Sprite:
     (a) Sum up the byte values of all the blocks occupied by Dug on the Game board. Add to High Score.
     (b) Set all blocks occupied by Dug to 0
     (c) If blocks occupied by Dug overlap with that of either of the Pookas or Fygars, decrement Dug's life by 1, reset game.
     (d) If collision with wall, do not update position.
   - For the enemy sprites:
     (a) If collision with wall, set random direction.