INF-1100: Introduction to programming and computer behavior

Assignment 5: The amazing bouncing balls

Eskil Bjørnbakk Heines

November 4th, 2022

1 Introduction

In this assignment, I was tasked to make a classic bouncing ball animation with several balls bouncing. I was given a pre-code where I was to implement the following:

1. SDL-initialization and event handling in main

2. An algorithm for creating, destroying, and drawing the balls

3. Making a linked list

4. Making the bouncing balls algorithm

2 Technical Background

The pre-code uses the Simple Directmedia Layer (SDL) [1] to open a window with a given size where we can implement structures or objects and make changes to them. The pre-code also uses Makefiles [2] to make the compiling process easier for all the different files of code. To solve the tasks in the assignment I have used a variety of functions and statements, if/else statements, for/while-loops, malloc statement, struct commands to make the list and objects, and a variety of commands from SDL.

In the assignment we are tasked to use a list instead of an array. Both an array and a list are data structures that help us have control over the data in the program. Using linked list is mainly because it is easier to have control of the objects data in a list than in an array. It's also easier to both add and remove objects from the list while having control where it is and what it contains.

3 Design and Implementation

To solve all the tasks in this assignment, several algorithms and functions had to be made. I chose to start with creating the balls, then making them bounce, and at last adding them to the list and drawing more balls. I will go through the process of these tasks and what the code does.

3.1 Implementing the main loop

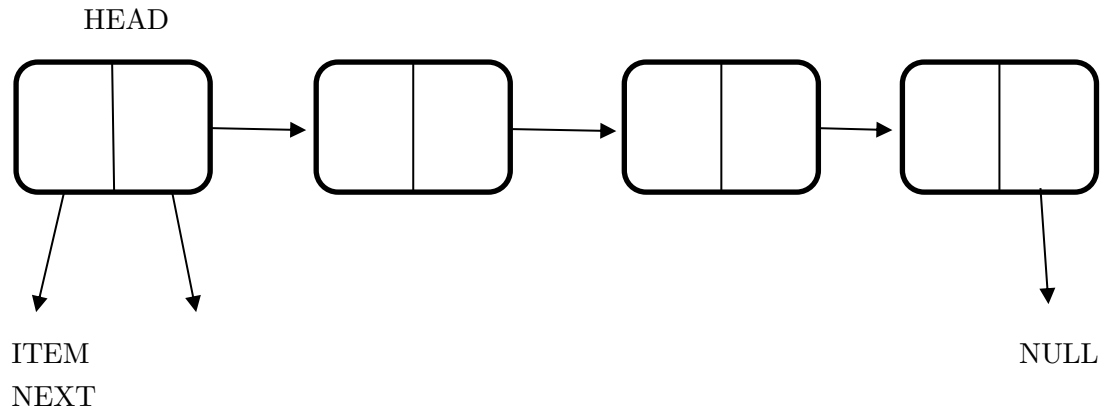
The main loop initializes an SDL-window, but without event handling, the window will not work for more than a second. I implement a loop to always check if something is happening in the code, so it will continue if it is events happening. Without this loop the screen will open and shut instantly. I have also added that if the spacebar is pressed while the balls are bouncing, a teapot will appear and act as a bouncing ball.

3.2 Creating, drawing, and destroying an object

The algorithm for creating a ball will create a piece of data that is allocated to the heap. This assures that the ball created can be stored in a list and be removed if necessary. The object created is pointed to a struct where all the needed variables for the ball are stored. After creating the data for the ball, it must be drawn onto the screen. The pre-code comes with a function for drawing lines on the screen. While looping this function, I use the coordinates given in the pre-code to draw all the triangles that together form a ball. When destroying the ball, the data that is allocated in the creating of the ball, is simply removed from the heap which results in destroying the ball.

3.3 Linked list

The algorithms made so far will only make one ball appear on the screen, and this is where the linked list [3] comes in. After creating the data for the ball and allocating it, I want to store it in a list so I can create another ball. This was a part of the assignment and necessary for creating more than one ball without making a lot of code. I implemented the logic for how a list works. I create a node that contains an item and a pointer to the next node. The first node in the list will always be the head, and the last one will always point to NULL. This helps to know where the start and end is, but also with iterating through the list.



3.4 Bouncing ball algorithm

This is the important algorithm where I put together all the algorithms I already made. Here I had to use logic to put together the algorithms to create and draw several balls, but also make them move across the screen while bouncing off the boundaries of the screen. To make the algorithm easier to understand, I defined some new variables for gravity and screen boundaries. Then, pointing to the list-struct, I looped the drawing of the balls while adding them to the list after they had been drawn. After drawing the balls, I made another loop that checks that if there are balls stored in the list, if there are the balls will move across the screen with gravity and rotation applied to it. When the balls hit the boundaries of the screen, the speed is reversed so the balls bounce back into the screen. Therefore, the balls will never leave the screen, only when they stop bouncing and lay still for 5 seconds. And when there are no balls left in the screen, the program will quit, and the screen is closed.

4 Discussion

The code seems to be a bit buggy. Sometimes when running the program, the outcome is different without changing the code. There also seems to be some holes where it bugs out. Point to a few occasions where the balls randomly stop without a logical explanation.

When the balls stop at the bottom of the screen, they continue to bounce a little. This is because gravity still pushes on them, and speedy never hits a clean 0. This means the balls stop, but never lays completely still.

5 Conclusion

I was tasked to make a classic bouncing ball animation on an SDL-screen using different algorithms. I have implemented algorithms for creating, drawing, and destroying a ball, algorithms for applying the balls to the screen and making them bounce, and algorithms for a linked list to store the data from the balls created. I have a program that creates balls, adds

them to the list, bounces off the boundaries of the screen, stops at the bottom, and is removed after laying still for 5 seconds. I will say the assignment and all the tasks are completed and the program works as intended.

Acknowledgement

Working on the assignment I have discussed the tasks with classmates Kristian Harvey Olsen and Svein Håkon Bjørkli Jansen.

I also have found some code online from someone working on the same assignment, this really helped with solving the logic behind the linked list. Also helped with inspiration on how to solve the main loop as well. Some of the functions in linked list are taken from here. [4]

References

[1] Simple DirectMedia Layer, Simple Directmedia Project (SDL):

https://www.libsdl.org/

[2] Makefiles, GNU Make:

https://www.gnu.org/software/make/manual/html_node/Introduction.html

[3] Linked list, Programiz.com:

https://www.programiz.com/dsa/linked-list

[4] Cprogramming.com, cod e that helped solve the tasks:

https://cboard.cprogramming.com/c-programming/171652-need-some-help-bouncing-ball-algorithm.html