HTML5 Frogger Game

Raj Zachariah

HCI Lab  
2116 Hornbake Building  
University of Maryland – College Park  
+1-202-215-3873

rajankz@umd.edu

Tansy Peplau

HCI Lab

2116 Hornbake Building

University of Maryland – College Park  
+1-661-524-5269  
tansy.amelia@gmail.com

**ABSTRACT**

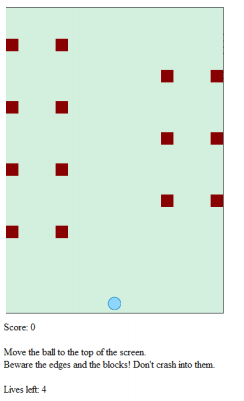


Figure 1: Game Screenshot from a mobile device

In this paper we describe a game developed entirely in HTML5/JS/CSS. The purpose of the project is to design an application in html5 meant to run in a web browser in a mobile device.

**Keywords**

Html5, canvas, web-game, frogger

# INTRODUCTION

# The purpose of this project and report is to fulfill the assignment requirement for the CMSC838F class. The assignment requires a student to create an HTML5 based application that could be run in a web browser on a mobile device. For our project we have decided to create a game, where the motive is to move a ball across the device screen without crashing into enemy units or falling off the screen.

# DESCRIPTION

The game is inspired by the classic arcade game **Frogger**. The authors of the game wanted to play similar game on their mobiles and compare score, which was not possible as there was no Frogger type game that could run on both iPhone/iPod and Android mobiles. And so for the class assignment we decided to create an html5 game instead, that could run easily on most mobiles.

Input methods on a mobile are different from a traditional arcade input or even a simulated input device of a desktop/laptop. To make the input seamless, we have skipped any form of touch-based input to play the game and instead use the accelerometer data from the phone to move the ball across the screen.

For now we have created this as a 2D game, and hence use only the x and y coordinates given by the device’s accelerometer, which is accessed in HTML5 by ‘deviceorientation’ event data. The ‘gamma’ value specifies the Left-Right tilt or the change in X coordinate and the ‘beta’ value specifies the tilt in Front-Back or the change in Y coordinate.

Within the game a user is given 4 lives to start and each time they die, a life is lost. When they run out of lives, the losing streak counter is increased. Winning results in a point increase, and losing results in point loss unless users have 0 points, in which case user stays at 0 points. We believe the points and lives is an interesting feature of the game because it gives a reason for users to play and keep playing the game.

# PROCESS

We started with a list of ideas each with a unique perspective on the use of sensor data and mobile browser uses. We then chose to create an application that uses the accelerometer data. Originally our application would use physics like the box2d example found online[[1]](#footnote-1) as well as data from the accelerometer, and creation of sound. However, the physics proved too difficult to program with the amount of time we had and we could not get objects to interact as expected. We then decided to change the focus and create a mini-game without the use of sound and physics.

We decided to create a mini-game within a browser application and we started programming by understanding and practicing the mobile programming techniques. Our application uses the accelerometer data as well as object creation and movement within a canvas.

To collaborate, we used github to host the code, as well as our own hosting websites to test the application. We interacted both online via email and chatting clients as well as in person writing code together.

We found the original code for the accelerometer and canvas examples on html5Rocks website[[2]](#footnote-2)[[3]](#footnote-3).

# CHALLENGES FACED

We faced a couple of challenges during the development of this assignment. The initial challenge was to learn how to use the canvas element in html5 and to understand how to collect data from the device accelerometer to use within the application.

Through this learning process we learned certain firmware (Android below 4.0) do not support browser accelerometer data. The next set of challenge was to understand the use of a canvas object in HTML5 and constantly update the canvas with moving objects. We also had trouble remembering to clear the background of the canvas to hide any previous images. We had to spend more time debugging code than expected, which resulted in the application not being as developed as we had hoped.

The most challenging issue is that different devices behave differently to the canvas element. Certain devices read the code as expected, while others don’t, which leads to some devices leaving objects on screen after a clear background function is called. We spent a significant amount of time trying to debug our code before we realized the problem was with the Android firmware as the code worked as expected on iOS devices and devices running Android 4.0 or later. Another problem we faced is the reaction to scaling on different devices. For example, when trying to set dynamic canvas dimensions, each device reacts differently and the application is skewed and objects do not behave as expected. We were able to overcome most of the challenges regarding code function, however, more time is needed to fully develop this game.

1. Box 2D example - http://box2d.org/manual.pdf [↑](#footnote-ref-1)
2. Device Orientation Tutorial and example code - http://www.html5rocks.com/en/tutorials/device/orientation/ [↑](#footnote-ref-2)
3. Arc Tutorials - http://www.html5canvastutorials.com/tutorials/html5-canvas-arcs/ [↑](#footnote-ref-3)