

# International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: [www.ijarcsms.com](http://www.ijarcsms.com)

## *WebRTC based Real-Time Game*

**Abhay Katiyar**

Dept. of Computer Science & Engineering  
Jabalpur Engineering College, Jabalpur  
Madhya Pradesh – India

---

**Abstract:** *To build a web based multiplayer game on the principles of WebRTC for modern browsers and help in further enhancement of this technology and look for more applications of it.*

**Keywords:** *Web Real-time Communication, WebSOCKETS.*

---

### I. INTRODUCTION

With the introduction of WebRTC (web Real-time Communication) [2] in modern Web browsers (namely Google Chrome, Firefox and Opera) VoIP calls, file transmission and data channels can be created between the web browsers itself. The modern computers are converging to the web and in the coming future so wills the gaming industry.

This project aims at developing at building a web based real-time multiplayer game using WebRTC, and WebGL such that two players can come online, share a private room and webcam and play using gestures. This project will provide a strong back bone to the online gaming industry and also to the open source web developers who can use the products and continue to improve WebRTC[6] which is still an evolving technology.

The project is feasible for several reasons. First, real-time communication is moving from SOCKETS to WebSOCKETS so most of the modern web browsers support it. If you are having WebRTC enabled browser you can play it irrespective of the device one is on. It adds interportability. Second, since we are using WebGL for graphics which uses GPU for graphic processing most of the part of the CPU can be utilized for other processes. Thus, even heavy games won't slow down the computer. Third, we are trying to achieve virtual reality by using a simple webcam.

The central task is to integrate WebRTC and DataChannel for the game, build the gaming arena using WebGL and threejs and then add game rules for the game play.

### II. PROJECT ASPECTS

Development of this game requires us to understand the following aspects of game development and web programming.

- *getUserMedia API:* One needs to setup video viewing on his device before being able to send the same over the network. gUM API does the same. It finds the most suitable video device and help video to get shown on the user screen i.e. to the room in which user is connected.
- *RTCPeerConnection API:* This creates a signalling server between the two web browsers. One browser sends a JSON offer containing browser info, video and audio device info and candidate info to the other browser and the other browser Responds with the JSON answer as to whether the call is accepted or not. Upon successful execution the user establishes a secure video and/or audio connection.

- *WebGL creation:* We are using WebGL's threejs plugin to develop the arena of the game. i.e. the UI of the room. WebGL is a 3D library used to develop graphics on web pages.
- *DataChannels:* DataChannels were integrated in WebRTC so that information can be shared online in real-time. We are using this to send the score information between the browsers.
- *JS Face detection and Gesture recognition:* In order to play one needs to recognize hand gestures so that the ball can be hit. For this we are using JavaScript's gesture recognition libraries to solve our purpose.

### III. SCOPE

The scope of the project is to provide a platform for developers to improve upon the current state of Web gaming and help in making real-time communication in web more advanced and robust. Building graphics for web has always been a tedious task as all the objects and views need to be self-designed. Integration of gesture recognition adds to the complexity of the project.

The proposed approach uses random key generation to generate a random URL for the room, RTCPeerConnection to get the video, JS Face detection to recognize hand gestures, DataChannels to transfer data packets and WebGL[4] to build the gaming arena.

With the development of this project we aim to learn about:

- the architecture involved in web app dev
- how VoIP works
- using Python's SDK to create web applications and then integrating modules to the same
- 3D modeling in web using WebGL
- working on gesture recognition and face detection
- Using HTML5[6] and CSS3 at an advanced level

The main objectives of the project are:

- To design the gaming arena. i.e. create a view-point, build the table tennis table, design the room, add video streaming to the background front wall, design the balls and racket and create a scoreboard.
- Develop gesture recognition so that it can recognize gestures and hit the ball when ball is reached towards the player accordingly.
- Create a Data transmission channel using WebRTC's Data Channel API.
- Random room generation using a web server language, python 2.7.
- Testing the efficiency and accuracy of the gestures, perform Unit test and check for errors and make project as much stable as possible.

IV. HIGH LEVEL DESIGN

A. BLOCK DIAGRAM

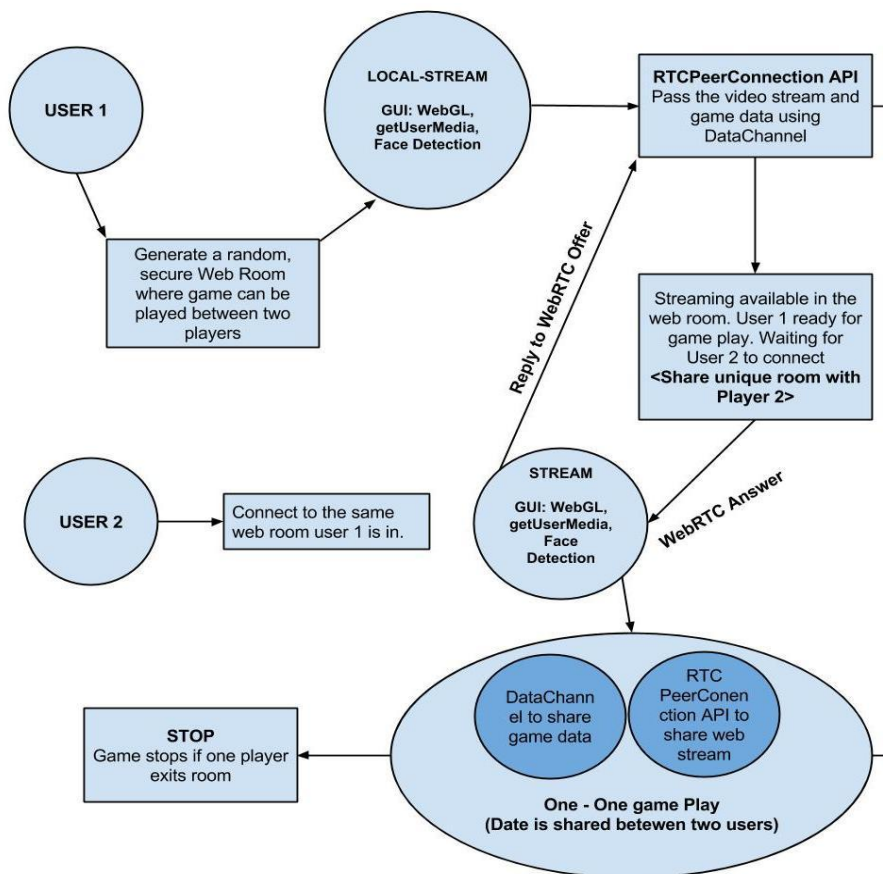


Fig. 1 Block Diagram of the game

B. DATA FLOW DIAGRAM

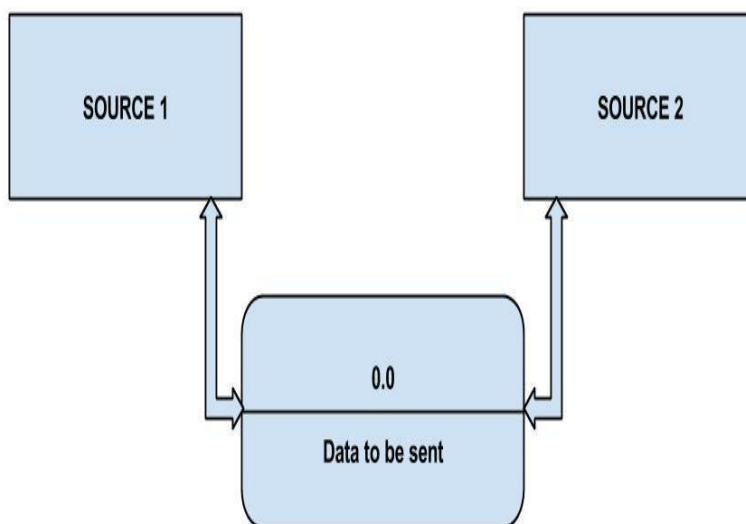


Fig. 2 Data Flow Diagram - Level 0

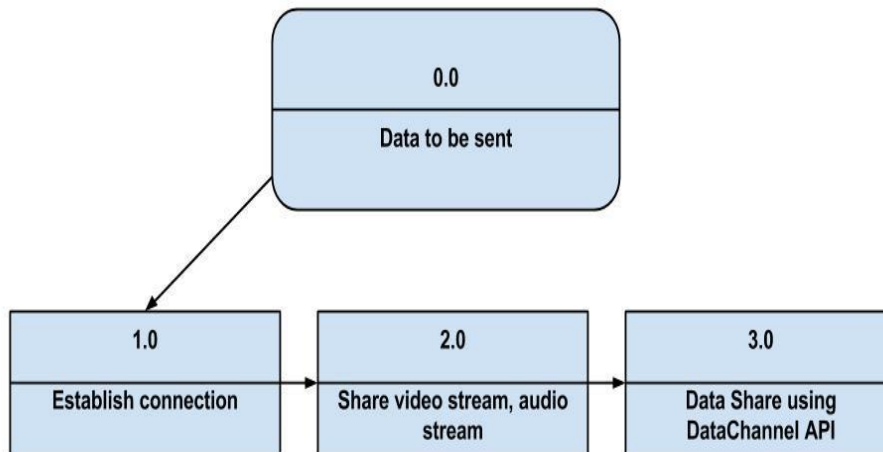


Fig. 3 Data Flow Diagram - Level 1

### C. ARCHITECTURE MODEL

Since the project deals with incremental changes and development over the period of time, it is built using the incremental model of software development process.

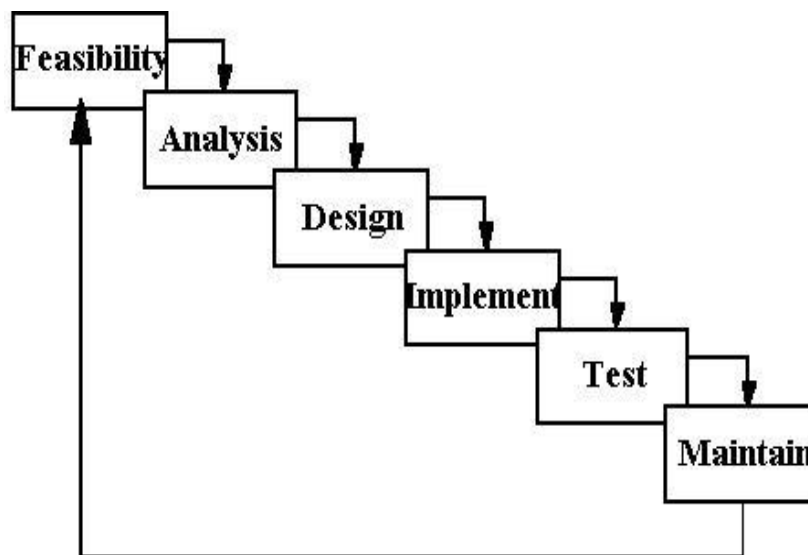


Fig. 4 Incremental Model

Why Incremental Model?

- The main motive of the project is to search for options to make web based gaming Industry more advanced and for this we needed to make changes and rollback at quite a number of time to get better results. So it was best to use Incremental Model.
- Incremental model allows making changes even after the completion of the project from the very start adding flexibility to our work.
- I have defined activities and represented them into separated process phases. All the stages overlap and fed information to each other. It is not simple linear model but involves a sequence of iterations of development activities.
- This model is appropriate for my project as I had to give more time to look for implementation of the DataChannel and WebRTC. This model generally takes less time to complete the software life cycle but requires improvement at later point of time.

## V. LOW LEVEL DESIGN

## 1. FLOW CHART AND IMPLEMENTATION

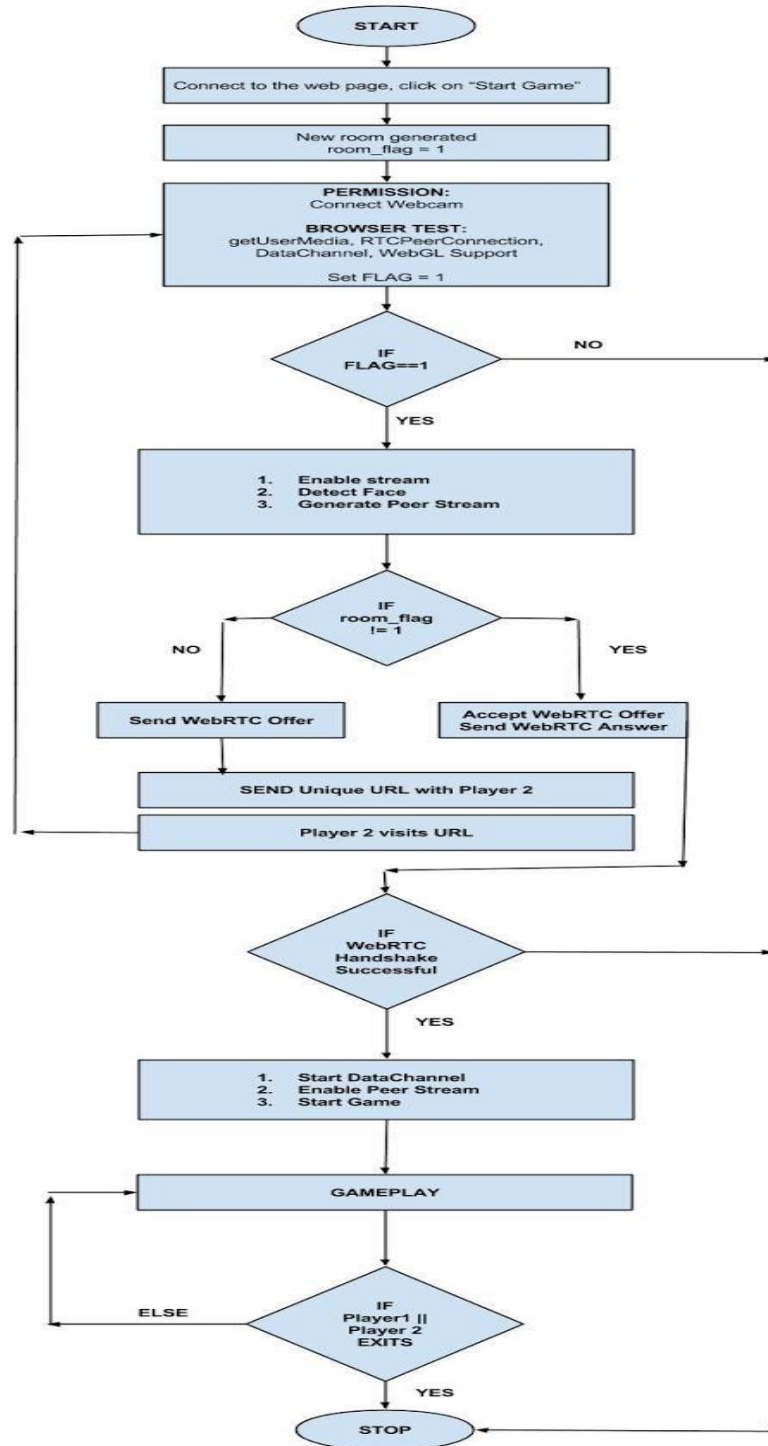


Fig. 5 Flow Chart

## VI. LIMITATIONS

- One of the major limitations is the perfect detection of gestures. It will be improved over time.
- There is no Pause method during the gameplay.
- getUserMedia API can get lag when the network traffic is congested.
- Only supports one-one gameplay as video sharing between more than two streams causes high usage of network bandwidth
- Infinite Gameplay. There is no duration that player wins after X moves

## VII. FUTURE ENHANCEMENT

- Improve gesture recognition
- Add NAT Traversal Support for Multiplayer gaming. i.e. more than two players can share videos and play.
- Add algorithm for ping pong gameplay.
- Improve WebGL graphics.

## VIII. CONCLUSION

The version of this game is v1. With what we have implemented gives us fair depth knowledge of how WebRTC, WebGL, gesture recognition, VoIP and asynchronous data transmission using DataChannel works inside the web browser.

We hope with development of this project more and more developers will be inspired to work in this field.

## References

1. Unmesh Shukla's example <http://www.jecj.in/ufo/unmeshcamcloth.html>
2. WebRTC site <http://webrtc.org>
3. Mr.Doob (three.js developer) <http://mrdoob.com>
4. HTML5 and CSS3 consortium WebGL <http://en.wikipedia.org/wiki/WebGL>
5. Mozilla WebRTC
6. WebRTC basics HTML5Rocks <http://www.html5rocks.com/en/tutorials/webrtc/basics/>
7. TeamOpen CV <http://opencv.org>
8. WebRTC reference <http://goo.gl/nWPIio>
9. Open Tok <http://opentok.com>
10. WebRTC Streaming <http://flashphoner.com/webrtc-streaming-server-for-live-broadcasting-and-webinars/>
11. Scalable WebRTC server <http://stackoverflow.com/questions/18318983/webrtc-scalable-live-stream-broadcasting-multiCasting>

## AUTHOR(S) PROFILE



**Abhay Katiyar**, currently working as Assistant professor in Computer Science & Engineering Department, Jabalpur Engineering College ,Jabalpur Madhya Pradesh(India)