

Paper Toss Game!

Tony Chau
tony.chau002@umb.edu
University of Massachusetts Boston



Figure 1: Screenshot of web page when you first load in.

ABSTRACT

This is a game where you try to toss a paper ball into the trash bin. You use the right and left arrow keys to move the arrow on the

screen to the direction you want to toss at. You hold space bar to charge/change the power of the toss. Once you let go of space bar key, the paper ball will be released and tossed.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CS460, Fall 2020, Boston, MA

© 2020 Copyright held by the owner/author(s).
ACM ISBN 1337.
<https://CS460.org>

KEYWORDS

WebGL, Visualization, Threejs, Game

ACM Reference Format:

Tony Chau. 2020. Paper Toss Game!. In *CS460: Computer Graphics at UMass Boston, Fall 2020*. Boston, MA, USA, 3 pages. <https://CS460.org>

1 INTRODUCTION

I picked this project because I was always into web games and app games. This paper toss game was an app game that I enjoyed a lot as a kid. This app was removed from the app store 2019, so I wanted to recreate the game. Additionally, I have always wanted to learn the fundamentals and skills to create a game like this.

2 RELATED WORK

My idea for the final project came from an very old app called Paper Toss [2]. This project used Three.js [1].

3 METHOD

This project is created with Three.js. The scene was created with two planes that are connected with each other at a 90 degrees angle by PlaneBufferGeometry. With one being vertical and one being horizontal, it creates the floor and the wall of the scene. The wall is mapped with a beautiful picture of UMass Boston and the floor is mapped with an image of concrete. The goal of this game is to get the highest score in throwing paper ball into the trash bin without missing. There are two TextBufferGeometry for creating the scoreboard and the power setting. The score goes up whenever you get the paper ball into the trash bin. The power of your throw goes up when you hold onto space bar and the paper ball will be tossed when you release the space bar key. The paper ball is created with SphereBufferGeometry. The arrow is created with a skeleton helper. I choose to use the skeleton helper because the arrow is multi parts. It has a cone head and a long body. This way I can move the whole object as a whole. The arrow moves when the user presses on the left and right arrow key. The paper ball will move with the arrow head. In the code, after you release the space bar key, it will take in consideration the power and angle to determine where it will be tossed. The trash bin is created with a LatheBufferGeometry with a recycling bin image. A bounding box is created, so we can calculate if the paper ball hits the trash bin. If the ball hits the bounding box of the trash bin, the score goes up and the user gets to toss again. If they miss, the losing screen comes up with the current score and the game ends.

3.1 Implementation

The paper ball is created with SphereBufferGeometry with an image of a crumbled piece of paper. The ball gets tossed depend on power and the angle of the x value. The power determines how far it will travel before it starts to drop. Initially, it will travel at the same Y-axis because of the power, then it gets decreased by 10 till it hits the floor or wall. The X-axis is determined by the rotation of the arrow head. See code below.

```
function ballThrow() {
  if (power > 0) {
    sphere.position.x += -r.root.rotation.z*10;
    sphere.position.z -= 10;
    power -= 2;
  }
  else {
    sphere.position.x += -r.root.rotation.z*10;
    sphere.position.z -= 10;
    sphere.position.y -= 10;
```

```
}
}
```

The trash bin created with a LatheBufferGeometry with a trash bin image. The trash bin gets its location randomized on the floor plane, so the trash bin will spawn at a different location to make it harder for the user.

The text in with the score board and the power setting are created with a TextBufferGeometry. The score goes up by 10 points when the paper ball hits the trash bin's bounding box. The power setting changes when the space bar key is pressed down.

The arrow is created with a skeleton helper. By pressing the left and right arrow keys, it will rotate the arrow head to the left or right. It is done by using the function rotation.x with a global variable that we keep track, so we know the rotation from the original position.

The wall and floor is created PlaneBufferGeometry. It is mapped 3000 width and 1000 height to cover the whole screen.

3.2 Milestones

3.2.1 Milestone 1. I wanted to create all the scene and all the objects that I needed in the scene before I did all the calculations, so the first thing I did was to create the two planes, the sphere, the arrow, and the font objects

3.2.2 Milestone 2. I needed to decide on the layout of the game such as where does the score board goes, where does the power setting go, and where should the ball and arrow placed.

3.2.3 Milestone 3. Create the mechanics of the game. Things like how does the arrow move, how does the ball get tossed, create the a bounding box for the ball to hit and testing if the physics of the game makes sense.

3.2.4 Milestone 3. Making the game more appealing to the eyes. Adding meshes with images. Making the arrow's color and the trash bins's color pop out more. Messing with the text buffers to complement the surroundings.

3.2.5 Milestone 3. Testing the game to make sure it runs smoothly and successfully.

3.3 Challenges

- Challenge 1: In class, we learned about slerp with quaternions. In the beginning, I used that for the movements of the arrow. I soon realize I had to take another approach because the arrow will always slerp to the end position. At the end, I used the rotation.x function to rotate the arrow with a changing variable, depending if left or right arrow is pressed.
- Challenge 2: Creating the mechanics of the paper toss was tough. At first, I imagined that I had to use actual mathematics with physics, such as velocity, gravity, etc, to compute the distance of the throw. As I was trying out the ball toss, I made it really simple. Ball moves forward depending on power and drops when there is no more.

4 RESULTS

The final result of my project is a simple game that you can play in your down time. This game is highly addictive because you want



Figure 2: Here we have an image of the losing screen.

to keep getting a high score. However, I wish I had more time. Graphically, I believe I could have done more to make it look more prettier.

5 CONCLUSIONS

In conclusion, this final project has taught me a lot. I have always wanted to do game development and I believe I just took my first

steps with this course. By doing this project, I realized that a game might look simple, but the developers probably put many hours and spend a lot of money to make a popular game. Creating this game took a lot more effort than I thought it would. And still, if I had more time, I believe I could have made it better. I loved working on this final project. Like I said before, I always wanted to become a game developer and CS450 has prepared me to take the first steps in what I want to do when I graduate.

REFERENCES

- [1] Ricardo Cabello et al. 2010. Three.js. URL: <https://github.com/mrdoob/three.js> (2010).
- [2] Backflip Studios. 2009. PaperToss. URL: <https://paper-toss.en.softonic.com/iphone> (2009).