

# 3D Trino-Rex

Tejasvi Rao Vemula and Sai Prathap Reddy Byreddy  
{TejasviRao.Vemula001@umb.edu},{S.Byreddy001@umb.edu}  
University of Massachusetts Boston



Figure 1: 3D Trino-Rex Game.

## ABSTRACT

The main objective of this TrinoRex game is to avoid the obstacles by either jumping or moving right / left. The goal is to survive as long as possible and earn the best scores.

## KEYWORDS

Three.js, Cannon.js, WebGL

### ACM Reference Format:

Tejasvi Rao Vemula and Sai Prathap Reddy Byreddy. 2022. 3D Trino-Rex. In *CS460: Computer Graphics at UMass Boston, Fall 2022*. Boston, MA, USA, 3 pages. <https://CS460.org>

## 1 INTRODUCTION

TODO: This is the 3D version of the pixelated dinosaur survival chrome game which is played so often by everyone when the internet is off. We have gone through several websites and References as mentioned in the below.

---

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 2022, Boston, MA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 1337.

<https://CS460.org>

## 2 RELATED WORK

This Game is based on the Cannon.js which is known as the 3D physics engine and Three.js [1].

Credits: <https://threejs.org/> ;

<https://github.com/schteppe/cannon.js>

## 3 METHOD

We have all played the chrome Dino game which appears when you try to visit a website while disconnected from the Internet. We took that as a reference to create our 3D Trino-Rex game. Comparatively our game is similar to the chrome Dino game, but the former is 3D and later is a pixelated version.

3D Trino-Rex game is relatively simple minded. It has a lot of objects such as Tyrannosaurus, obstacles such as cacti, poles and a score counter. To start the game, you need to press "Start" button on the screen. After pressing the start, the Trino will automatically start running towards the obstacles. The main goal of this game is to avoid the obstacle from colliding with the Tyrannosaurus. This actually looks very easy, but if you start playing the game you will find it difficult because of the running speed of the Trino and a lot of obstacles.

When the Trino comes across any of the obstacles, you can move it to right, left or jump. To jump over the obstacles, you should press the up arrow. Likewise, if you want to move it to right and left, you can press the right, left arrows respectively. The score

gets increasing as long as you play the game. The game will be over as soon as you crash any of the objects. If you want to play again, you can press the "retry" button on the screen.

### 3.1 Implementation

```

#ifdef BONE_TEXTURE
uniform highp sampler2D boneTexture;
uniform int boneTextureSize;
mat4 getBoneMatrix( const in float i ) {
    float j = i * 4.0;
    float x = mod( j, float( boneTextureSize ) );
    float y = floor( j / float( boneTextureSize ) );
    float dx = 1.0 / float( boneTextureSize );
    float dy = 1.0 / float( boneTextureSize );
    y = dy * ( y + 0.5 );
    vec4 v1 = texture2D( boneTexture, vec2( dx * ( x + 0.5 ), y ) );
    vec4 v2 = texture2D( boneTexture, vec2( dx * ( x + 1.5 ), y ) );
    vec4 v3 = texture2D( boneTexture, vec2( dx * ( x + 2.5 ), y ) );
    vec4 v4 = texture2D( boneTexture, vec2( dx * ( x + 3.5 ), y ) );
    mat4 bone = mat4( v1, v2, v3, v4 );
    return bone;
}

```

3D Version Tyrannosaurus implementation

```

#ifdef DOUBLE_SIDED
reflectedLight.indirectDiffuse += ( gl_FrontFacing ) ? vIndirectFront : vIndirectBack;
#else
reflectedLight.indirectDiffuse += vIndirectFront;
#endif
#include <lightmap_fragment>
reflectedLight.indirectDiffuse *= BRDF_Diffuse_Lambert( diffuseColor.rgb );
#ifdef DOUBLE_SIDED
reflectedLight.directDiffuse = ( gl_FrontFacing ) ? vLightFront : vLightBack;
#else
reflectedLight.directDiffuse = vLightFront;
#endif
reflectedLight.directDiffuse *= BRDF_Diffuse_Lambert( diffuseColor.rgb ) * getShadowMask();

```

The game looks like Trino running opposite to the objects and those objects comes facing towards us.

```

vec3 RRTAndOOFit( vec3 v ) {
    vec3 a = v * ( v + 0.0245786 ) - 0.000090537;
    vec3 b = v * ( 0.983729 * v + 0.4329510 ) + 0.238081;
    return a / b;
}

vec3 ACESFilmicToneMapping( vec3 color ) {
    const mat3 ACESInputMat = mat3(
        vec3( 0.59719, 0.07600, 0.02840 ),    vec3( 0.35458, 0.90834, 0.13383 ),
        vec3( 0.04823, 0.01566, 0.83777 )
    );
    const mat3 ACESOutputMat = mat3(
        vec3( 1.60475, -0.10208, -0.00327 ),    vec3( -0.53108, 1.10813, -0.07276 ),
        vec3( -0.07367, -0.00605, 1.07602 )
    );
}

```

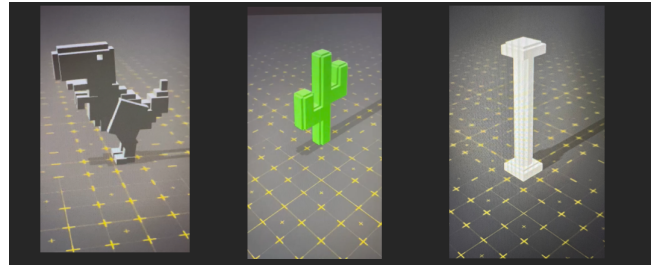
The Dragon position is fixed, but the platform on the floor seems like moving and on that moving platform. The objects like cactus come against the dragon.

### 3.2 Milestones

How did you structure the development?

**3.2.1 Milestone 1.** The Tyrannosaurus, Cactus and other objects like poles are the .glb version. glb version is similar like .gltf file. But, unlike gltf file glb file objects are provided with features like shadow, animation. These are the objects rendered in the project.

**3.2.2 Milestone 2.** Cannon.js is the main library used in this project. This is totally depends on the physics library implementations. Cannon.js provides great functionalities like a) gravity whenever the Tyrannosaurus jumps it comes down to the floor. And b) object collision: whenever the Tyrannosaurus is collided with cactus or pole like objects the game is over.



**3.2.3 Milestone 3.** Coming to the Front-end interface of the game. The Game is provided with start button, score, and levels. The score gets increased as far as the Tyrannosaurus moves by escaping the objects and when it is collided with objects the game is over. After game over, we can see the invert(100

**3.2.4 Milestone 4.** When we click the start button, the game gets started and for escaping the objects the Tyrannosaurus is moved by arrows keys like move up, down and jump. after the object collision game is over and there is an option like retry button to restart the new game with score level 0.

### 3.3 Challenges

Describe the challenges you faced.

- Creating this game itself was pretty challenging for us. We read lot about this in google and other websites and worked on this. But it was exciting and fun to work.
- We faced lot of problems in learning cannon.js. This is new library and implementation code is very strange and lot of calculations are involved.
- Initially when we tried to use jump funtion, the Tyrannosaurus used to stay in air and faced some difficulties for placing the Trino in to original position
- In addition, we came across issues like placing the random objects, the main motive of the game is to escape from the objects

## 4 RESULTS

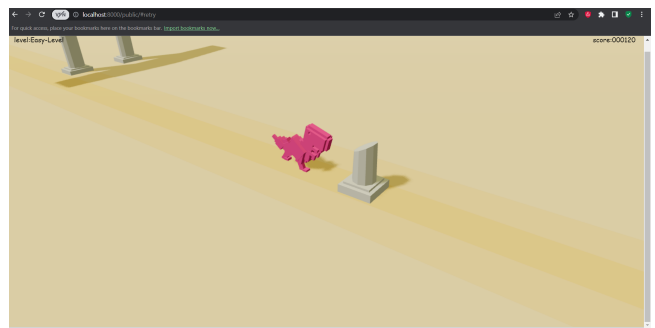


Figure 2

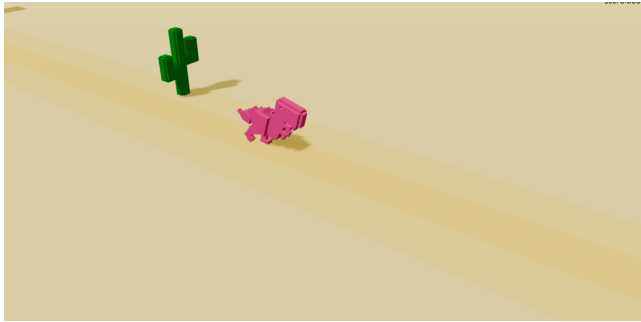


Figure 3

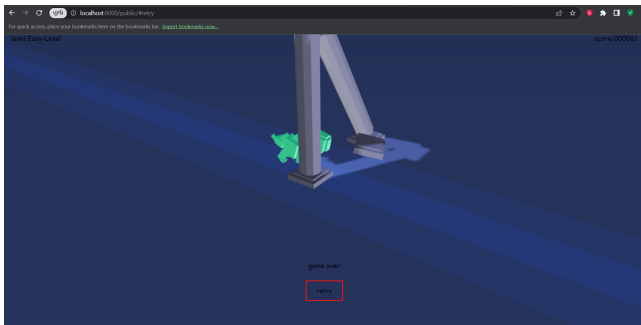


Figure 4

#### 4.1 Our Github pages and Github link for source code are below:

<https://prathap-024.github.io/cs460student/Final%20Project/3D%20Trino-Rex/public/>

<https://github.com/Prathap-024/cs460student/tree/main/Final%20Project>

<https://tejasvirao-024.github.io/cs460student/Final-Project/>

<https://github.com/TejasviRao-024/cs460student/tree/main/Final-Project>

## 5 CONCLUSIONS

To sum up, The pixelated version game is our motivation behind for doing this project. We are very excited for achieving this progress in the game. If the objects are too high to jump we can use left or right key to avoid obstacle instead of jump. And, mainly the shadow effect and objects which we are used are of glb version files and seems like very clear animated. In Future, We might add few exciting features and some difficult situation while scoring high.

## REFERENCES

- [1] Ricardo Cabello et al. 2010. Three.js. URL: <https://github.com/mrdoob/three.js> (2010).
- [2] <https://threejs.org/>
- [3] <https://www.youtube.com/watch?v=hBiGFpBle7E> explains hoe to build objects using Three.js and cannon.js
- [4] <https://github.com/rossning92/t-rex>
- [5] <https://github.com/stubbies>

[6] <https://github.com/schteppe/cannon.js> we can find lot more examples of cannon.js objects creation and implementations from the above link

[7] <https://www.youtube.com/watch?v=KJ38qCwFdy8t=123s>