

2D/3D Animated Art 1

Jialong Huang and Jie Lin
{jialong.huang001},{jie.lin003}@umb.edu
University of Massachusetts Boston

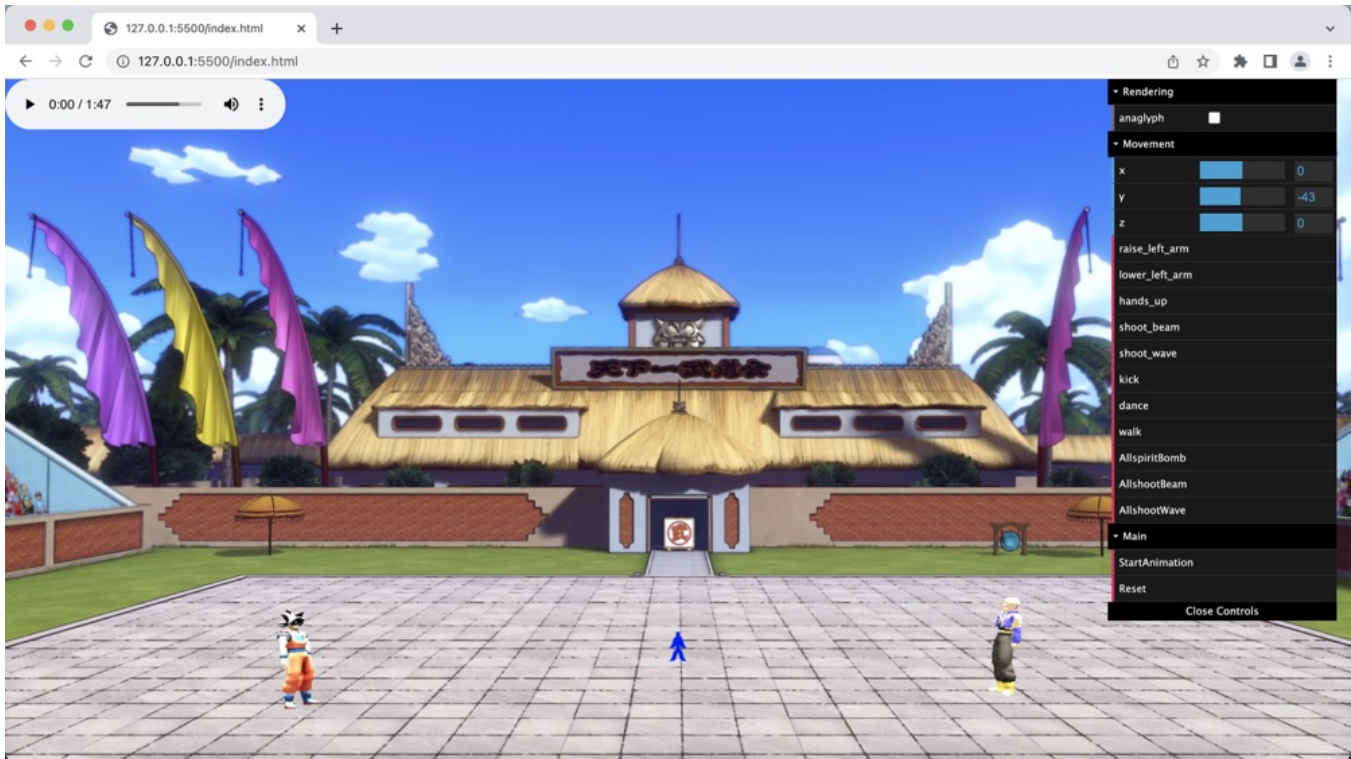


Figure 1: Start Scene

ABSTRACT

We use ThreeJS and Robot class which was provided in the professor's GitHub to build characters' attacks. We also use Mixamo to auto-rigged our models and animations. Then we use Blender to modify our model, such as by separating the body parts for texture. Later we load the model and animations in our threejs.scene.

KEYWORDS

Dragonball, 3D animation, Visualization, thrill, Threejs.

ACM Reference Format:

Jialong Huang and Jie Lin. 2022. 2D/3D Animated Art 1. In *CS460: Computer Graphics at UMass Boston, Fall 2022*. Boston, MA, USA, 3 pages. <https://CS460.org>

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>

1 INTRODUCTION

We chose this project because we both had a passion for Dragon Ball, our favorite animation from childhood, so we wanted to make our own Dragon Ball animation. Therefore, we chose the most representative Tenkaichi Tournament as the background and the theme song of Dragon Ball as the background music. There is also Son Goku's signature moves Spirit Bomb and Kame Hame Ha.

2 RELATED WORK

Here you can cite existing related work like XTK [5] or Three.js :

Three.js[4], Javascript, CSS, HTML, Blender[2], Mixamo[3], Model Tutorial[7], Particle Explosion[8], Robot Model Assignment[6], Goku's Model[9], Future Trunk's Model[1]

3 METHOD

Methods in our project in detail:

1.playModifierAnimation(from, fSpeed, to, tSpeed) used for switching animation state. from is usually in an idle state, to where the animation we want to use.

2.setHalfVolume() used for initializing volume.

3.randomVal() is used for generating random values.

4. `check_beingHit_byBeam(robot)` used to check collision between the beam and the robot in a line.
 5. `cast_beam(robot)` used for robots to shoot beams in a line.
 6. `check_beingHit_byBall(sphereMesh)` used to check the collision between the ball and robot in a circle.
 7. `cast_ball(robot, ty, tz, t_time)` used for robots to throw a ball for a short distance.
 8. `cast_wave(robot)` is used for robots to shoot a beam and generate an explosion effect at the target location.
 9. `removeItemFromArray(arr, value)` used to remove the object from an array.

3.1 Implementation

Please tell the reader how you implemented the project. You can include code snippets that you want to highlight. Don't include the whole code.

For a character to use their abilities:

We use ThreeJS's geometry and material to generate a mesh that helps us to build characters' attacks. For example, they can throw a ball or a wave depending on their current active state.

For Load models and animations:

We use THREEjs's texture loader to add the texture to the model and loader.load function to add the model to the scene.

```
const MODEL_PATH = 'goku3.glb';
let goku_txt = new THREE.TextureLoader().load('body.png');
goku_txt.flipY = false;

const goku_mtl = new THREE.MeshPhongMaterial({
  map: goku_txt,
  color: 0xffffff,
  skinning: true });
loader.load(
  MODEL_PATH,
  function(gltf) {
    // A lot is going to happen here
    model = gltf.scene;
    let fileAnimations = gltf.animations;

    scene.add(model);
  },
  undefined, // We don't need this function
  function(error) {
    console.error(error);
  }
);
```

And we also use `THREE.AnimationClip.findbyname` to find the animation we modify in Blender and call it to run the animation.

```
let idleAnim = THREE.AnimationClip.findByName
```

```
(fileAnimations, 'idle');
let bombAn = THREE.AnimationClip.findByName
(fileAnimations, 'spiritbomb');
```

3.2 Milestones

How did you structure the development?

3.2.1 Milestone 1. We brainstormed different designs using the robot and the scene that we used in project 7 and see which one will be the best to perform the scene.

3.2.2 Milestone 2. We added skills for robots so that they are able to cast skills like spirit bomb and Kamehameha.

3.2.3 Milestone 3. We added particle explosions to simulate the explosion effect without using any models.

3.2.4 Milestone 4. We find the background, background music, and some animation for the robots.

3.2.5 Milestone 5. Then we find the model of Son Goku and the model of Trunk. So we decide to use these two models as our major characters to start the fight.

3.2.6 Milestone 6. We use Mixamo to rig and animate models and then import them into Blender to separate body parts in order to add textures.

3.2.7 Milestone 7. We start to replace robots with models and connect them to the animation.

3.3 Challenges

Describe the challenges you faced.

- Challenge 1: We have a hard time finding good DragonBall character models and we don't know how to import them into our scene.
- Challenge 2: When we import the library to load the model, it always says that is the wrong path of the library. Even though we have the same library path from the examples we found online.
- Challenge 3: The sources we find for models have different textures files for different body parts but after auto-rigged in Mixamo we only the whole body as one part. Then we only have one texture that can add to the model.
- Challenge 4: We are new to Blender and we don't know how to put two animations into one model and separate body parts.

4 RESULTS

We successfully imported the model into the scene and added the animation and textures. We also add an additional death animation like image 2

5 CONCLUSIONS

Describe your final conclusions in 1-2 paragraphs. Please double-check that you removed all instructions of this template in all sections - including this one. Good luck!

Basically what we doing:

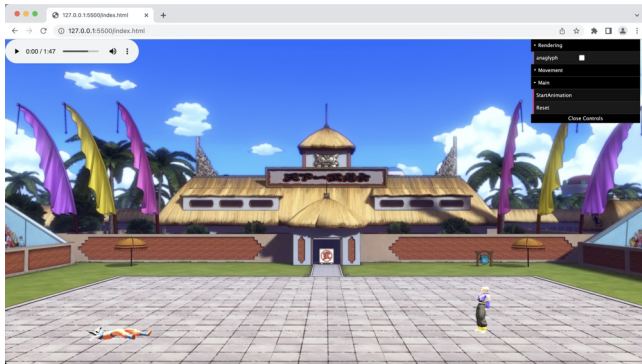


Figure 2: An example image.

1. We made robots before we handled models to see if it is possible to build characters' attacks.
2. Implement models and animations with our robot.
3. Use robots to use characters' attacks and play models' animations simultaneously.
4. Create a Start Animation Button to play the animation.

6 GITHUB LINK

GitHub link:

<https://gzhjl2000.github.io/cs460student/finalproject/>
https://jielin0o0.github.io/cs460student/Final_Project/

Source Code Link:

<https://github.com/gzhjl2000/cs460student/tree/main/finalproject>
https://github.com/JieLin0O0/cs460student/tree/main/Final_Project

REFERENCES

- [1] Unknown Author. 2022. GameCube - Dragon Ball Z Budokai - Trunks Future. URL: <https://www.aigei.com/item/longzhuqi.html> (2022).
- [2] Unknown Author. Unkown Year. Blender Model Editor. URL: <https://blender.com/> (Unkown Year).
- [3] Unknown Author. Unkown Year. Mixamo Animation Model. URL: <https://www.mixamo.com/> (Unkown Year).
- [4] Ricardo Cabello et al. 2010. Three.js. URL: <https://github.com/mrdoob/three.js> (2010).
- [5] Daniel Haehn, Nicolas Rannou, Banu Ahtam, P. Ellen Grant, and Rudolph Pienaar. 2012. Neuroimaging in the Browser using the X Toolkit. *Frontiers in Neuroinformatics* (2012).
- [6] Daniel Hann. 2022. Robot Three.js. URL: <https://gist.github.com/haehn/8fa2642f001ef6c142ddf4f0bbca3e7a> (2022).
- [7] Kyle Wetton. 2019. Load Model Three.js. URL: <https://tympanus.net/codrops/2019/10/14/how-to-create-an-interactive-3d-character-with-three-js/> (2019).
- [8] Xanmia. Unknown Year. ParticleExplosion. URL: <https://codepen.io/Xanmia/pen/nqyMgJ> (Unknown Year).
- [9] zai. 2022. Goku Low poly base form v1. URL: <https://www.cgtrader.com/free-3d-models/character/fantasy-character/goku-low-polybase-form-v1> (2022).