Marco Ciancia 1741186

September 2020

1 Project

The idea at the base of the project was to create a classic 2D fighting mini-game with all the basic mechanics: make two distinct character able to move left and right and to hurt each other with different attacks.
I was inspired by various fighting-games but the most of all I take it from "Street Fighter", so I decided to make a pun about it and called the game "Three Fighter" because of the three.js library which I used to create the models and the game scene.

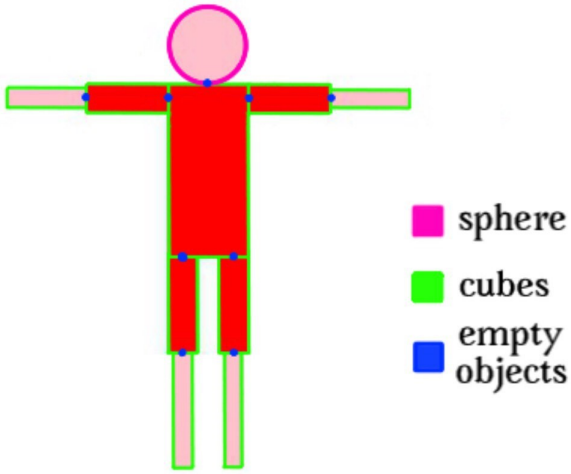
2 Scene, Lights and Camera

I used the three.js "THREE.scene()" method to build a scene for the game and instead of model a stage for the two character to play in I put a texture on the background. Besides the characters in the scene there are also two progress bar,with the name of the fighter they refer to at the bottom, used to keep track of the remaining life.
The camera doesn't need to be moved around so I place it in a fixed point at the center of the scene and shift it a little on the z-axis, the type of camera used is orthographic to see the characters always in the same way, no matter where they are in the scene.
For the light I decide to use two different type of illumination: a directional light placed centered on top of the scene and a weaker ambient light to be able to see even what was not illuminated by the first one.

3 Characters Model

For the characters I decided to create a single model for both of the character and just changing the color of the tunic to not confuse the players, since this is the only thing that make them different from each other I decide to call them by their color: Red and Blue.
In terms of objects the model is made by:

- 1 sphere: the head
- 9 cubes : one for the torso and eight for arms and legs
- 9 empty object : this objects are positioned on the neck(1)/elbows(2)/shoulders(2)/knee(2)/hip bone(2) of the hierarchical model and were required to make easier to control the body movements of the character and thus better animations



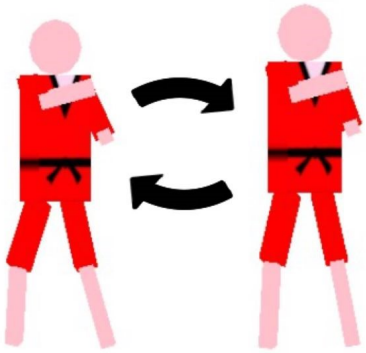
All of the cubes are provided with bounding box to register the collision between the two models and I also add two different texture to the "body-cube" to make it look like the characters are wearing a tunic and a black belt.

4 Animations

For the animations I used the tween.js library and create a "keyframe system": for every animation I create 2/3 important poses or "keyframes" for the character to perform and simply use tween to make the model smoothly transit to a series of poses and return to the initial one. The animations are the same for both of the characters.
The characters, as long as they are not knocked down, are always in an idle animation, when the player press a button or when gets hit by the opponent the animation brutally changes to match the proper action and when it end the character restart the idle animation.

4.1 Idle

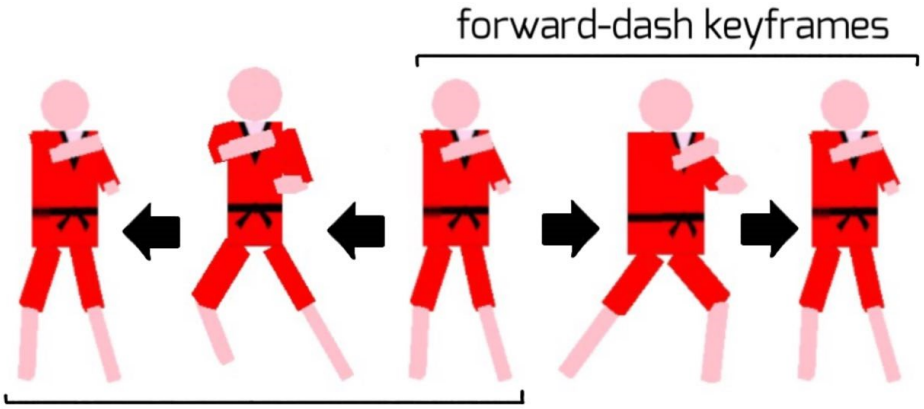
The character must perform some kind of loop movement when it doesn't perform any action or it will not look real: nobody can stay perfectly still continuously. Since our character is a fighter it has to keep his feet ready so I try to simulate a leg springing making the character slightly go up and down.



idle keyframes

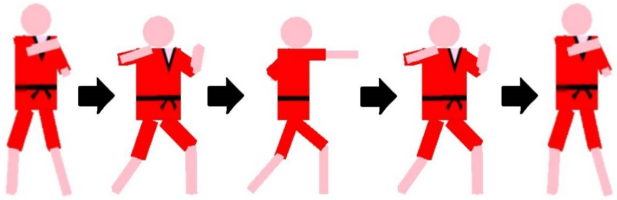
4.2 Forward and Back Movement

The character has to move forward and backward, to keep the things simple it will not be able to walk but just to perform a clear dash motion that will move it by a fixed amount of space.

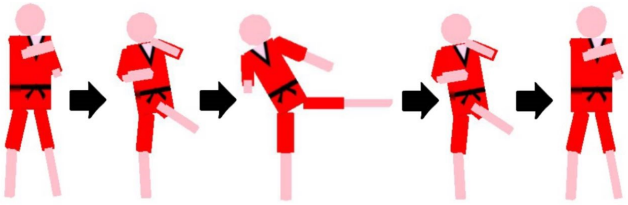


4.3 Punch and Kick

For the attacks I wanted to create two type of actions: a slow one that can reach far and a fast one for close combat,a kick and a punch seemed the most sensible choice.



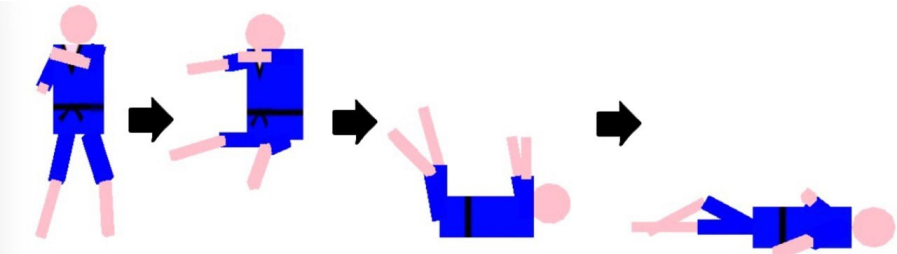
punch keyframes



kick keyframes

4.4 Hit and KO

To make clear that a hit was landed successfully I made a hit animation where the character seems to bend over in pain a little, and when the life of a players goes to zero the character will fall to the ground with a slight knockback due to the last hit received.
The hit animation is very simple, is just two keyframes, but the KO animation is the most complex of all : even if the kick and the punch animation has more keyframes to pass by half of them are just the first ones but in reverse order, the ko animation have 4 unique frames.



KO keyframes

5 The Game

5.1 Structure

The game will use two different html pages, one for the title screen and the other for the actual game, after one of the two player win, by attack the opponent until his life bar goes to zero, the second page can be refreshed to start a new match.
There are no button to click on in both of the pages because I wanted that all the functionality could be controlled with the keyboard without the need of a mouse.
The fight start with both of the character at the same distance from the center, the are invisible edges that prevent the characters to go off-screen, the controls are printed on the background so that they are always visible to be consulted.

5.2 Controls

To recap the character's active actions are : dash forward, dash backwards, punch and kick. We need 8 buttons to make all the things working, 4 buttons for character, and a button for play and restart the game. To make easier to play without the risk disturbing the other player with the hands the buttons were mapped on the two far side of the keyboard in a mirror image and the space bar acts in place of both "play" and "restart", the button will be disabled during the match until one of the two character is defeated so that it can be pressed and accidentally restart the fight in progress.