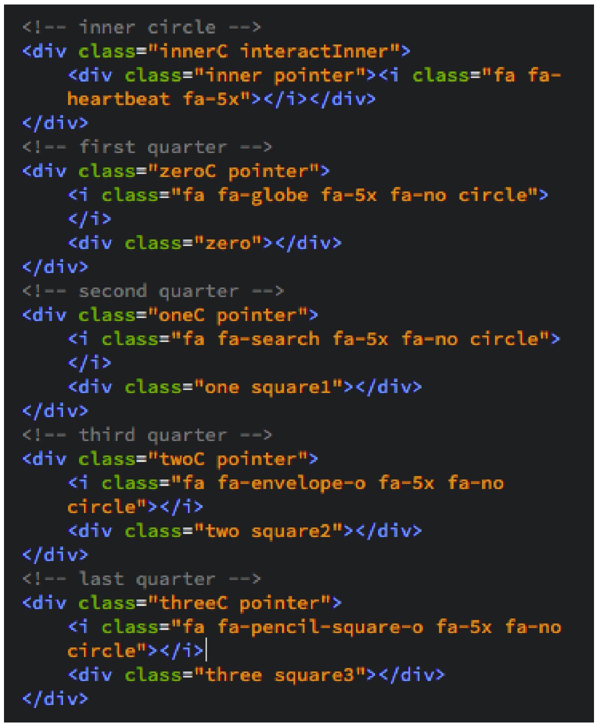
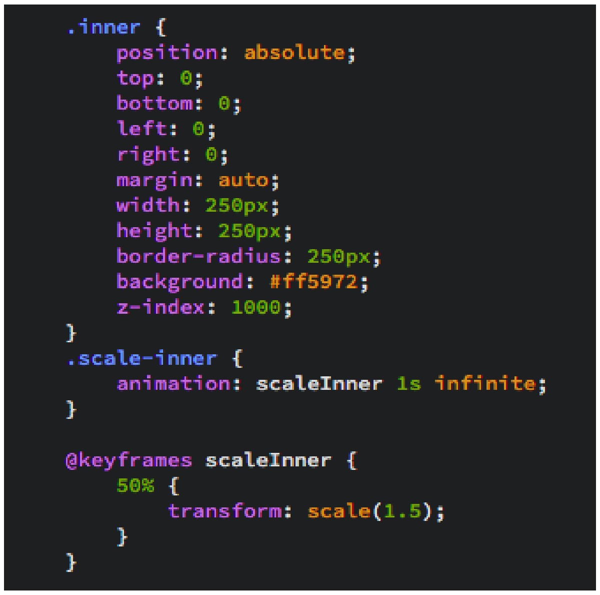


```
<!-- inner circle -->
<div class="innerC interactInner">
    <div class="inner pointer"><i class="fa fa-
    heartbeat fa-5x"></i></div>
</div>
<!-- first quarter -->
<div class="zeroC pointer">
    <i class="fa fa-globe fa-5x fa-no circle">
    </i>
    <div class="zero"></div>
</div>
<!-- second quarter -->
<div class="oneC pointer">
    <i class="fa fa-search fa-5x fa-no circle">
    </i>
    <div class="one square1"></div>
</div>
<!-- third quarter -->
<div class="twoC pointer">
    <i class="fa fa-envelope-o fa-5x fa-no
    circle"></i>
    <div class="two square2"></div>
</div>
<!-- last quarter -->
<div class="threeC pointer">
    <i class="fa fa-pencil-square-o fa-5x fa-no
    circle"></i>
    <div class="three square3"></div>
</div>
```

HTML – Hyper Text Markup Language

We used div elements to build the basic structure for the interactive menu. We assigned divs with the classes for the latter purpose of styling them with css as well as for the purpose of adding interactive functionalities to the DOM trough JavaScript. With the element i we embedded desired font awesome icons which we latter styled in CSS to achieve the appropriate size, color and position.

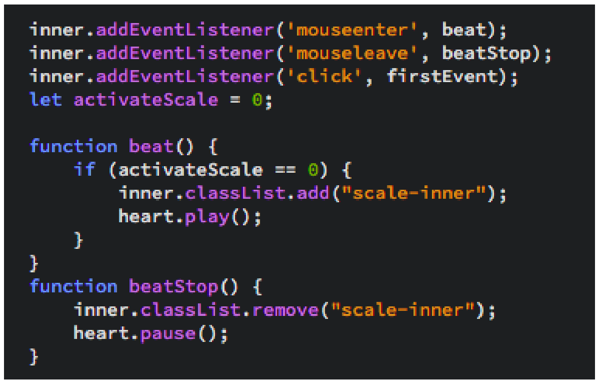


```
    .inner {
        position: absolute;
        top: 0;
        bottom: 0;
        left: 0;
        right: 0;
        margin: auto;
        width: 250px;
        height: 250px;
        border-radius: 250px;
        background: #ff5972;
        z-index: 1000;
    }
    .scale-inner {
        animation: scaleInner 1s infinite;
    }

    @keyframes scaleInner {
        50% {
            transform: scale(1.5);
        }
    }
```

CSS – Cascading Style Sheets

By adding classes to the html elements, we styled divs in the CSS file. We positioned the inner circle in the center of the webpage, assigned it the size, border radius of 250 pixels to achieve the circle look and used a hex color code to add a desired background color. With high z-index we ensured that inner circle will always be the front layer; therefore, it will always appear on top of the other divs.
We added a simple CSS animation to scale the circle to 150 % of its original size.



```
inner.addEventListener('mouseenter', beat);
inner.addEventListener('mouseleave', beatStop);
inner.addEventListener('click', firstEvent);
let activateScale = 0;

function beat() {
    if (activateScale == 0) {
        inner.classList.add("scale-inner");
        heart.play();
    }
}
function beatStop() {
    inner.classList.remove("scale-inner");
    heart.pause();
}
```

JAVASCRIPT

Using JavaScript, we declared the div inner as a variable. We added an event listener which triggers different functions on mouseenter and mouseleave. Function beat adds a css class scale-inner to the div with a class inner and plays a heartbeat sound. On the opposite beatStop function removes the css class scale-inner from the div with the class inner and stops the audio.