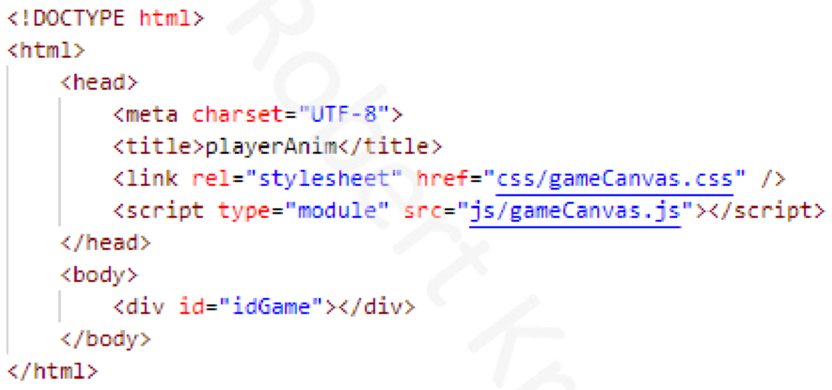
playerAnim

Sprite Animation

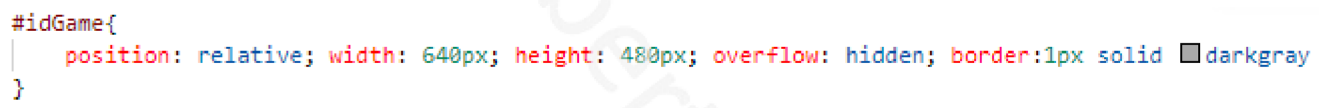
dr inż. Robert Krupiński

playerAnim.html



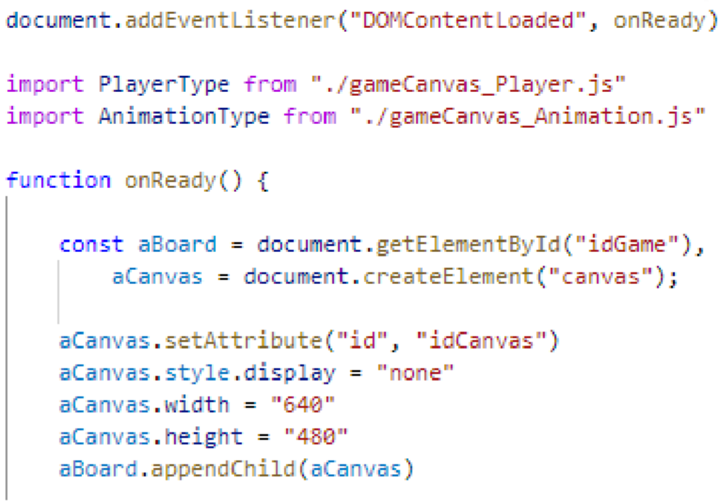
```html
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>playerAnim</title>
        <link rel="stylesheet" href="css/gameCanvas.css" />
        <script type="module" src="js/gameCanvas.js"></script>
    </head>
    <body>
        <div id="idGame"></div>
    </body>
</html>
```

gameCanvas.css

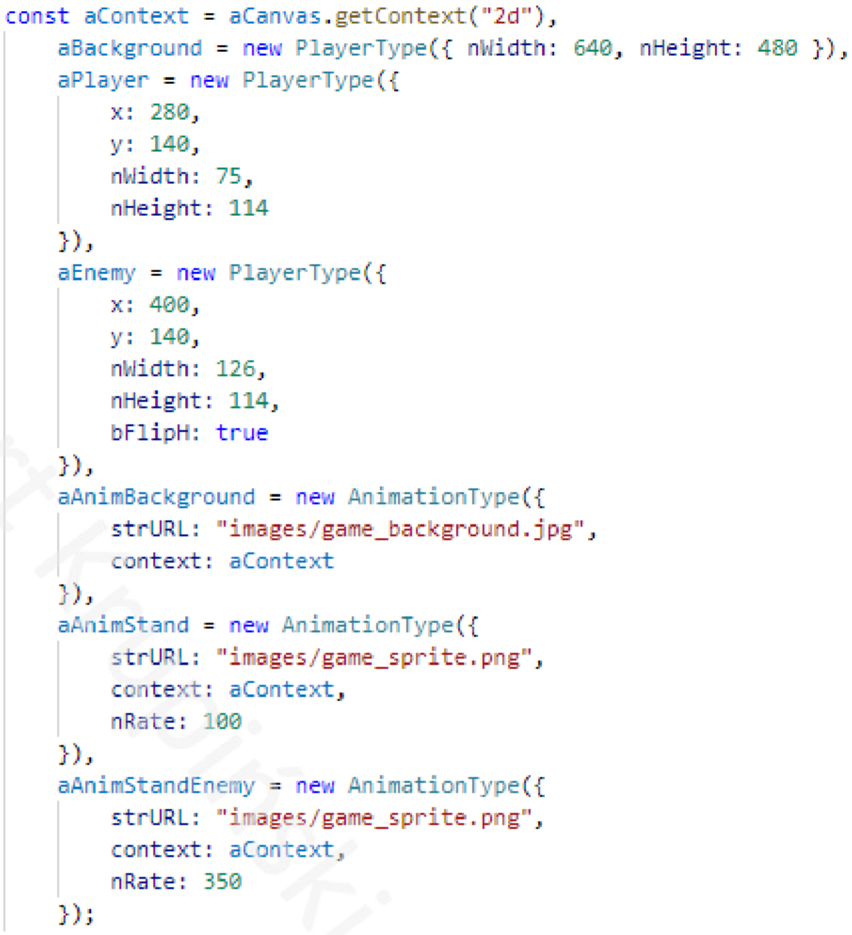


```
#idGame{
    position: relative; width: 640px; height: 480px; overflow: hidden; border:1px solid ■darkgray
}
```

gameCanvas.js

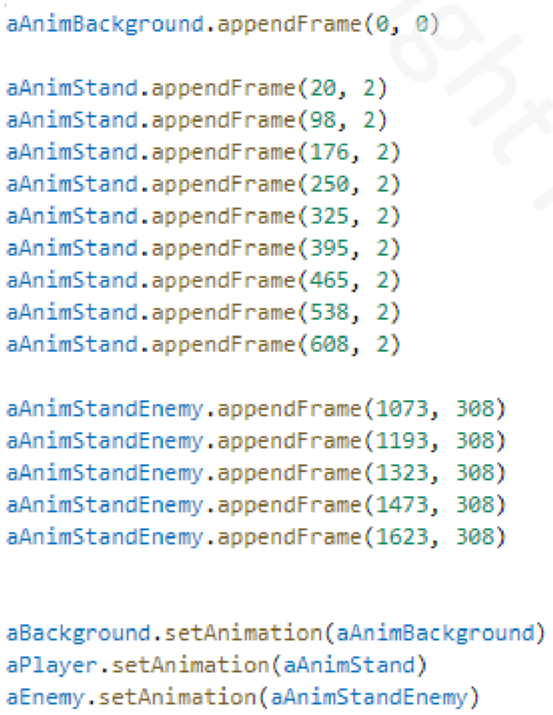


```
document.addEventListener("DOMContentLoaded", onReady)

import PlayerType from "./gameCanvas_Player.js"
import AnimationType from "./gameCanvas_Animation.js"

function onReady() {

    const aBoard = document.getElementById("idGame"),
        aCanvas = document.createElement("canvas");

    aCanvas.setAttribute("id", "idCanvas")
    aCanvas.style.display = "none"
    aCanvas.width = "640"
    aCanvas.height = "480"
    aBoard.appendChild(aCanvas)
```

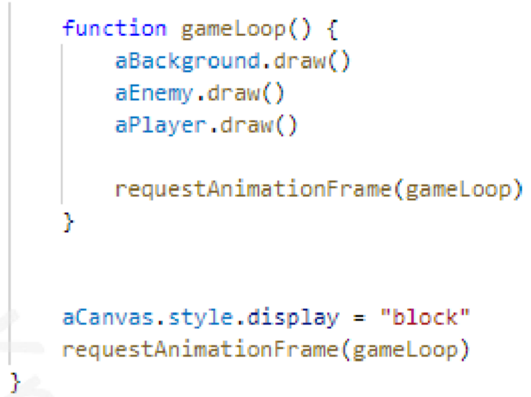


```
const aContext = aCanvas.getContext("2d"),
    aBackground = new PlayerType({ nWidth: 640, nHeight: 480 }),
    aPlayer = new PlayerType({
        x: 280,
        y: 140,
        nWidth: 75,
        nHeight: 114
    }),
    aEnemy = new PlayerType({
        x: 400,
        y: 140,
        nWidth: 126,
        nHeight: 114,
        bFlipH: true
    }),
    aAnimBackground = new AnimationType({
        strURL: "images/game_background.jpg",
        context: aContext
    }),
    aAnimStand = new AnimationType({
        strURL: "images/game_sprite.png",
        context: aContext,
        nRate: 100
    }),
    aAnimStandEnemy = new AnimationType({
        strURL: "images/game_sprite.png",
        context: aContext,
        nRate: 350
    });
```

gameCanvas.js

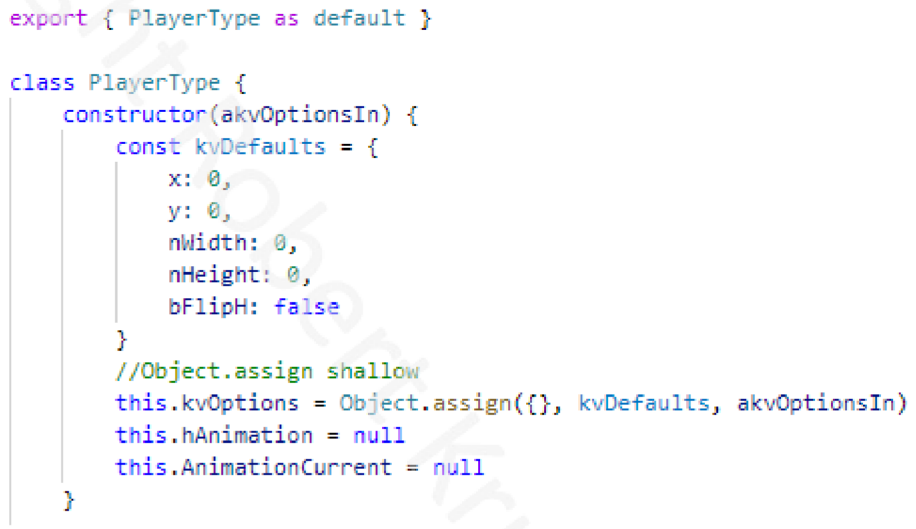


```
aAnimBackground.appendFrame(0, 0)

aAnimStand.appendFrame(20, 2)
aAnimStand.appendFrame(98, 2)
aAnimStand.appendFrame(176, 2)
aAnimStand.appendFrame(250, 2)
aAnimStand.appendFrame(325, 2)
aAnimStand.appendFrame(395, 2)
aAnimStand.appendFrame(465, 2)
aAnimStand.appendFrame(538, 2)
aAnimStand.appendFrame(608, 2)

aAnimStandEnemy.appendFrame(1073, 308)
aAnimStandEnemy.appendFrame(1193, 308)
aAnimStandEnemy.appendFrame(1323, 308)
aAnimStandEnemy.appendFrame(1473, 308)
aAnimStandEnemy.appendFrame(1623, 308)


aBackground.setAnimation(aAnimBackground)
aPlayer.setAnimation(aAnimStand)
aEnemy.setAnimation(aAnimStandEnemy)
```



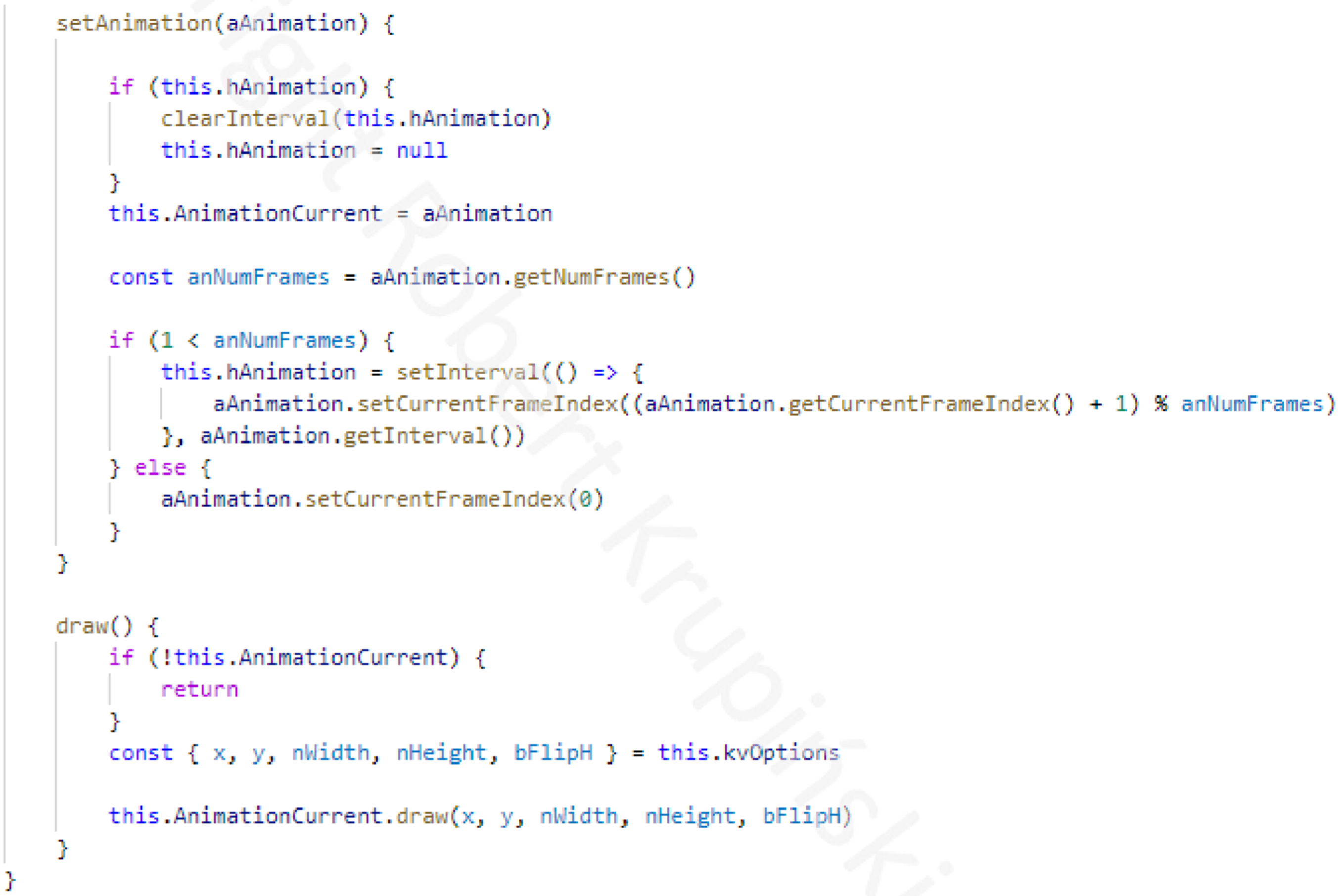
```
    function gameLoop() {
        aBackground.draw()
        aEnemy.draw()
        aPlayer.draw()

        requestAnimationFrame(gameLoop)
    }


    aCanvas.style.display = "block"
    requestAnimationFrame(gameLoop)
}
```

gameCanvas_Player.js



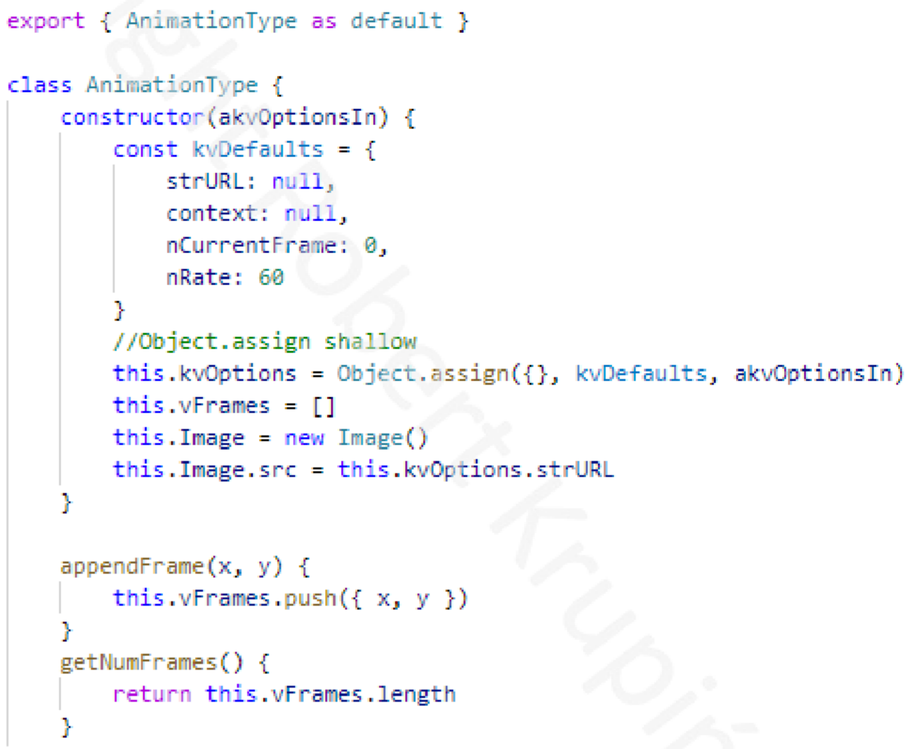
```
export { PlayerType as default }

class PlayerType {
    constructor(akvOptionsIn) {
        const kvDefaults = {
            x: 0,
            y: 0,
            nWidth: 0,
            nHeight: 0,
            bFlipH: false
        }
        //Object.assign shallow
        this.kvOptions = Object.assign({}, kvDefaults, akvOptionsIn)
        this.hAnimation = null
        this.AnimationCurrent = null
    }
```

gameCanvas_Player.js



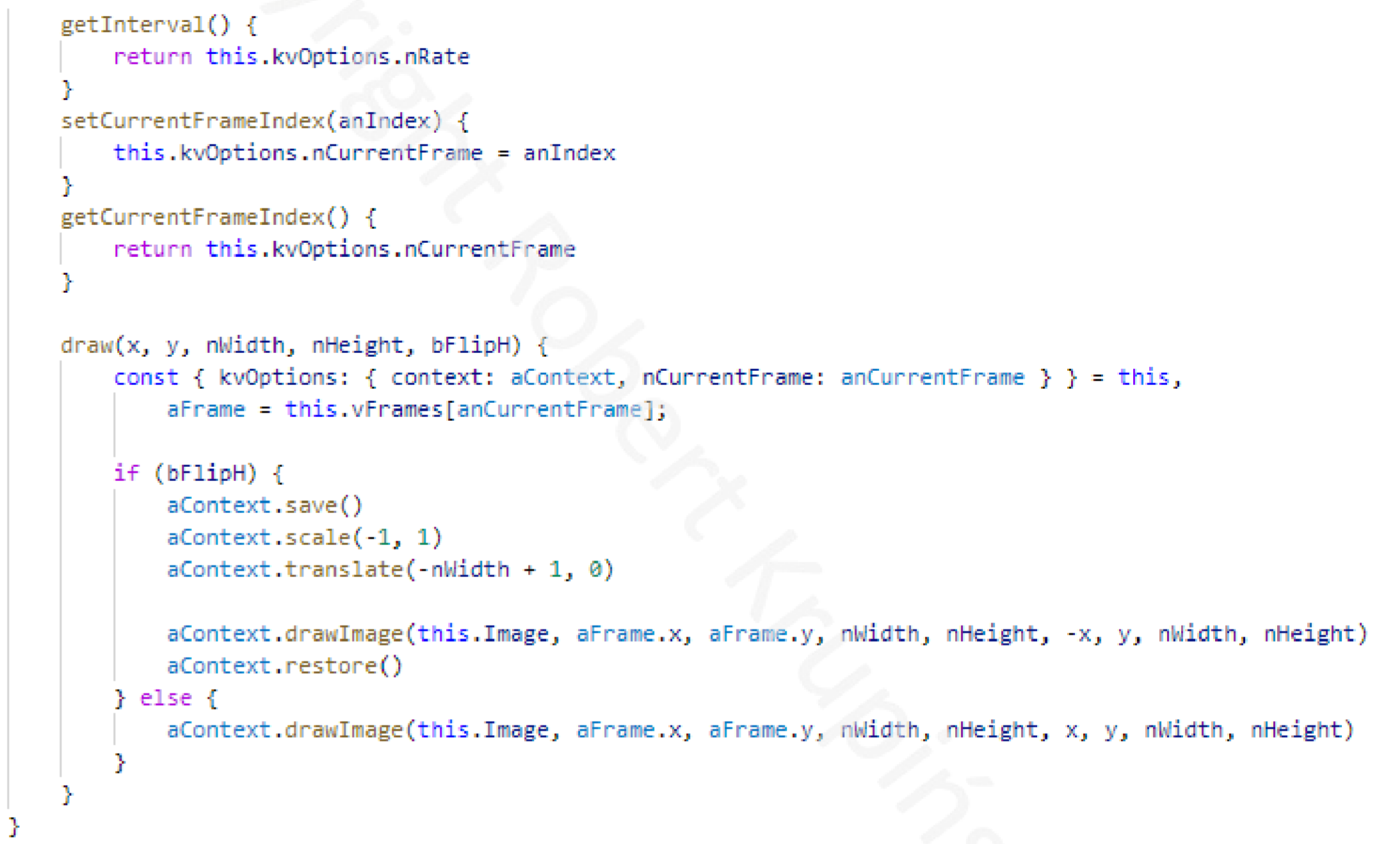
```javascript
    setAnimation(aAnimation) {

        if (this.hAnimation) {
            clearInterval(this.hAnimation)
            this.hAnimation = null
        }
        this.AnimationCurrent = aAnimation

        const anNumFrames = aAnimation.getNumFrames()

        if (1 < anNumFrames) {
            this.hAnimation = setInterval(() => {
                aAnimation.setCurrentFrameIndex((aAnimation.getCurrentFrameIndex() + 1) % anNumFrames)
            }, aAnimation.getInterval())
        } else {
            aAnimation.setCurrentFrameIndex(0)
        }
    }

    draw() {
        if (!this.AnimationCurrent) {
            return
        }
        const { x, y, nWidth, nHeight, bFlipH } = this.kvOptions

        this.AnimationCurrent.draw(x, y, nWidth, nHeight, bFlipH)
    }
}
```

gameCanvas_Animation.js



```
export { AnimationType as default }

class AnimationType {
    constructor(akvOptionsIn) {
        const kvDefaults = {
            strURL: null,
            context: null,
            nCurrentFrame: 0,
            nRate: 60
        }
        //Object.assign shallow
        this.kvOptions = Object.assign({}, kvDefaults, akvOptionsIn)
        this.vFrames = []
        this.Image = new Image()
        this.Image.src = this.kvOptions.strURL
    }

    appendFrame(x, y) {
        this.vFrames.push({ x, y })
    }
    getNumFrames() {
        return this.vFrames.length
    }
```

gameCanvas_Animation.js



```
    getInterval() {
        return this.kvOptions.nRate
    }
    setCurrentFrameIndex(anIndex) {
        this.kvOptions.nCurrentFrame = anIndex
    }
    getCurrentFrameIndex() {
        return this.kvOptions.nCurrentFrame
    }

    draw(x, y, nWidth, nHeight, bFlipH) {
        const { kvOptions: { context: aContext, nCurrentFrame: anCurrentFrame } } = this,
            aFrame = this.vFrames[anCurrentFrame];

        if (bFlipH) {
            aContext.save()
            aContext.scale(-1, 1)
            aContext.translate(-nWidth + 1, 0)

            aContext.drawImage(this.Image, aFrame.x, aFrame.y, nWidth, nHeight, -x, y, nWidth, nHeight)
            aContext.restore()
        } else {
            aContext.drawImage(this.Image, aFrame.x, aFrame.y, nWidth, nHeight, x, y, nWidth, nHeight)
        }
    }
}
```

File location

- www.rmaes.com/students/2020_2021/playerAnim.pdf