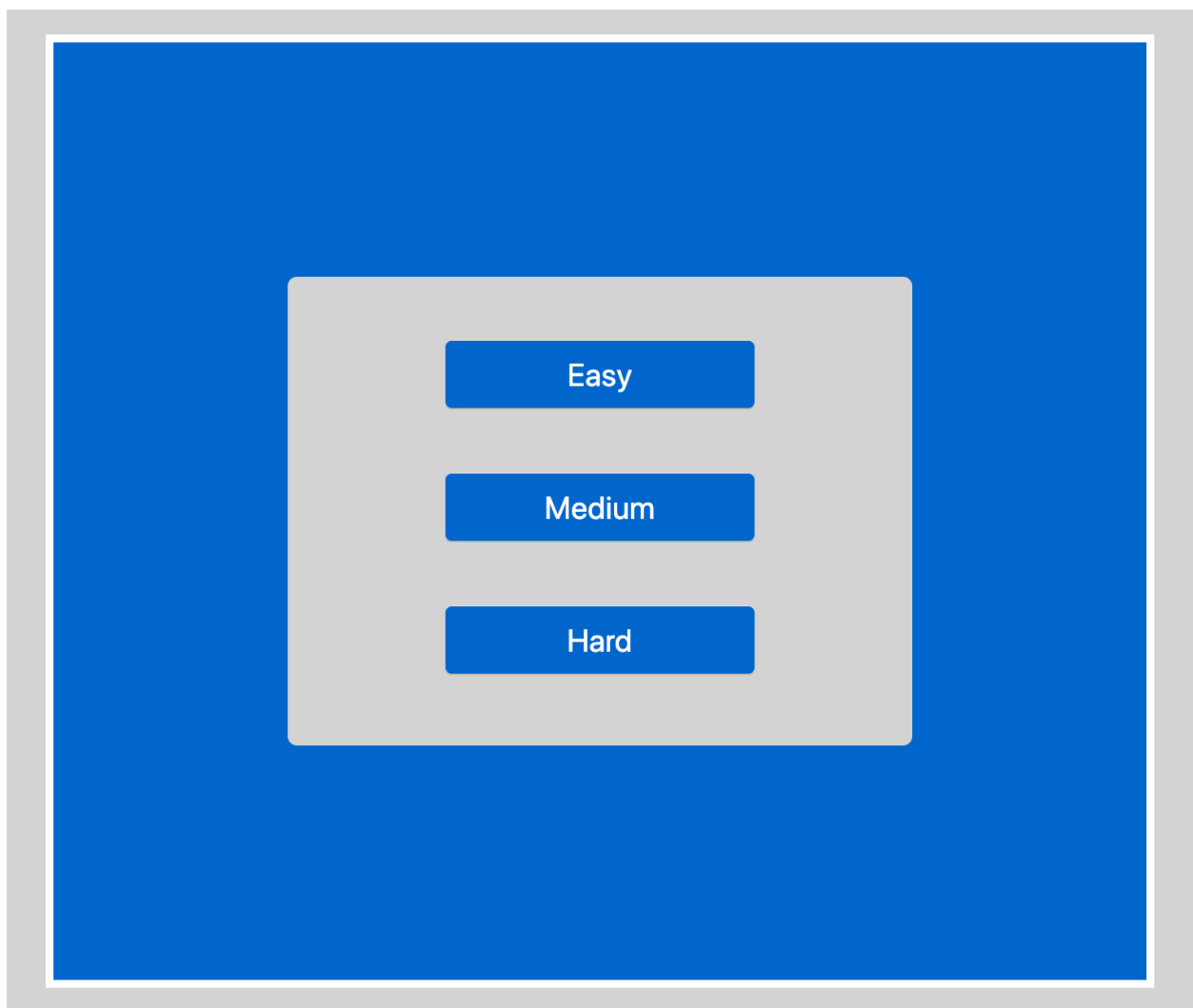


## 3.laboratorijska vježba iz računalne grafike

Za 3. vježbu iz računalne grafike napravio sam igru pong. Igra se može vidjeti na linku <https://jurikos.github.io/pong/> . Moguće ju je pokrenuti i lokalno na svome računalu. Nakon što se klonira ovaj git repozitorij potrebno je ući u folder s 3. laboratorijskom vježbom. (Ako još nemate instaliran nodejs <https://nodejs.org/en/> /ili yarn package manager potrebnno ih je instalirati na računalu). Zatim se u komandnu liniju upisuju naredbe

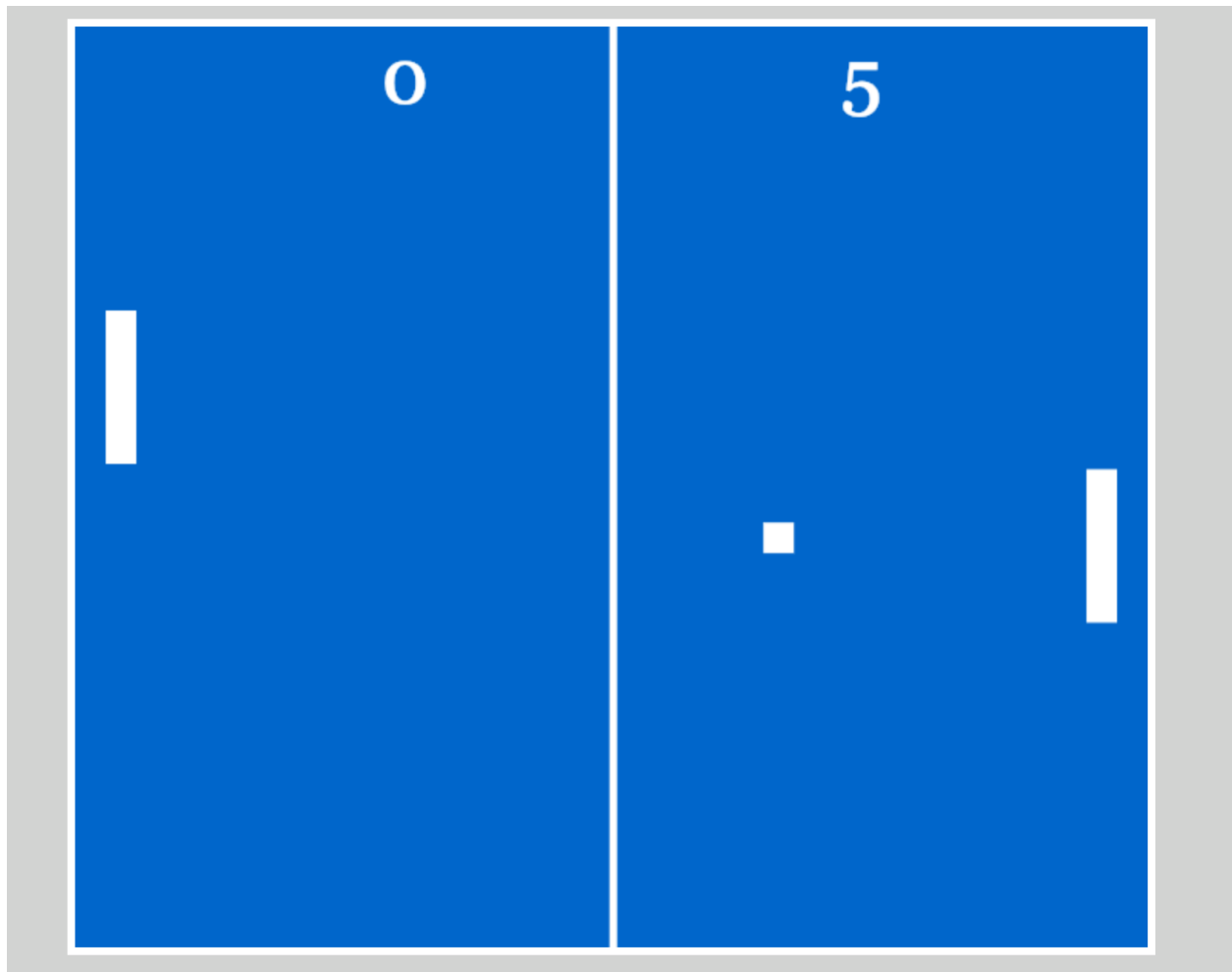
1. yarn
2. yarn start

Naredbom yarn se instaliraju node moduli koji su korišteni u projektu(ovdje je to samo dev server budo). Naredbom yarn start se otvara igra u web browseru. Nakon što se igra pokrenula pojavit će se izbornik u kojemu možemo odabrati težinu igre(slika 1).



*Slika 1.*

Nakon odabira težine započinjemo igru protiv računala. Reket možemo pomicati po ekranu strelicama gore i dolje. Da bismo pobijedili računalo trebamo postići 11 poena prije njega. Izgled igre možemo vidjeti na sljedećoj slici.(slika 2.)



*Slika 2.*

Igra je izrađena u programskom jeziku javascript. Za provjeru sudara između lopte i reketa koristio sam algoritam AABB(slika 3. linija 94). Kut pod kojim se loptica odbija od reketa ovisi o tome koliko je daleko udarila od središta reketa(slika 3). Ako lopticu pogodimo s rubom reketa ona će ubrzati(slika 3 linija 112).

```

92     }
93     const isIntersect = function(ax, ay, aw, ah, bx, by, bw, bh) {      You, 9 days ago • pong
94         return ax < bx + bw && ay < by + bh && bx < ax + aw && by < ay + ah
95     }
96     let racquet = this.speed.x < 0 ? player : ai
97     if (
98         isIntersect(
99             racquet.x,
100            racquet.y,
101            racquet.width,
102            racquet.height,
103            ball.x,
104            ball.y,
105            ball.side,
106            ball.side
107        )
108    ) {
109        this.x = racquet === player ? player.x + player.width : ai.x - this.side
110        const n = (this.y + this.side - racquet.y) / (racquet.height + this.side)
111        const phi = 0.25 * PI * (2 * n - 1) // pi/4 = 45
112        const smash = Math.abs(phi) > 0.2 * PI ? 1.5 : 1
113        this.speed.x =
114            smash * (racquet === player ? 1 : -1) * this.maxSpeed * Math.cos(phi)
115        this.speed.y = smash * this.maxSpeed * Math.sin(phi)
116    }

```

*slika 3.*