

## Konklusion

Vi har i dette projekt udarbejdet et spil i form af et kapløb, hvor to konkurrerende robotter bliver fjernstyret af to spillere og dystet mod hinanden om at komme først igennem en bane, hvilket også oprindeligt var hovedformålet med projektet. Overordnet set har vi opnået en væsentlig del af de mål for projektet, som vi satte os for i vores indledende diskussion af, hvad vores projekt kunne indeholde.

Hvis vi ser mere konkret på de forskellige elementer af spillet, så har vi blandt andet lagt op til, at der skal være checkpoints, som har farver, der er væsentligt forskellige fra den resterende bane, sådan at robotterne ved hjælp af lyssensorer ville være i stand til at detektere, når de havde nået et checkpoint, hvilket også er lykkedes. Her kan vi dog nævne den uheldige mangel der er i forbindelse med netop detektering af checkpoints, da vi ikke tager højde for det scenarie, hvor man blot kører sin robot tilbage på et checkpoint, hvor man allerede har været, hvilket klart er den nemmeste måde at vinde spillet på.

Udover anvendelse af lyssensorer, så har vi også anvendt ultralydssensorer til at kunne detektere, hvorvidt der er forhindringer foran robotten, som den skal forsøge at undvige, selvom den pågældende spiller forsøger at køre ind i den respektive forhindring.

I vores oprindelige projektbeskrivelse havde vi diskuteret, hvilke opgaver der skulle være i spillet, som man som spiller skulle forsøge at udføre. Her havde vi forskellige idéer, blandt andet at robotterne kunne samle objekter op, men vi diskuterede også på et senere tidspunkt i forløbet, om det ikke ville være sjovere at lave et racerløb i stedet, hvilket var den variant, som vi endte med at udføre. I forlængelse heraf, så har vi gjort det muligt for hver af robotterne at deaktivere dens modstander ved at køre ind i den bagfra og derved ramme TouchSensoren, som har et tilknyttet adfærdsmønster, der sørger for at suppressere fjernstyring af robotten.

Endelig har vi udarbejdet en GUI og anvendt Bluetooth til kommunikation mellem robotterne og PC'erne, hvilket også var hvad vi havde lagt op til i den oprindelige beskrivelse.

Undervejs i projektet løb vi ind i problemer med implementering af kommunikation imellem de to NXT-enheder, hvor vores oprindelige idé var, at lade de to enheder kommunikere direkte med hinanden ved hjælp af Bluetooth. Desværre havde NXT-enhederne den begrænsning, at de kun var i stand til at have en enkelt indadgående og tre udadgående forbindelser på samme tid. Samtidig understøttede LeJOS API'en ikke, at en PC kunne agere server for en NXT over Bluetooth, så her løb vi ind i problemer og måtte komme på en anden løsning, hvilket vi håndterede ved at oprette en socket forbindelse imellem de to PC'er, hvorpå GUI'erne kørte. Dette var bestemt ikke en særlig

elegant løsning, men trods dette så gav denne løsning ikke anledning til mærkbare forsinkelser, og derved opfyldte den trods alt sit formål.

I løbet af projektet havde vi som nævnt også forsøgt at udvikle en Android-baseret applikation, da dette i forhold til en PC ville være en langt mere oplagt enhed at anvende til fjernstyring af en robot. Desværre havde vi det problem, at vi ikke var i stand til at oprette en Bluetooth-forbindelse fra Android-telefonen, da vi her fik kastet en exception ved selve connect kaldet i applikationen. Vi brugte lang tid på at forsøge at finde grunden til denne fejl, men det var desværre forgæves.

Såfremt vi havde haft længere tid til projektet, så ville der være en oplagt forbedring af spillet, hvis vi kunne forhindre, at en spiller kunne misbruge et checkpoint. Her kunne man for eksempel anvende checkpoints i forskellige farver, så det ville være nemmere at detektere, om man havde passeret et bestemt checkpoint. Omvendt ville dette indføre en begrænsning i forhold til antallet af checkpoints, da vi på nuværende tidspunkt kan have et vilkårligt antal checkpoints ved kun at anvende checkpoints af en enkelt farve. En anden idé til forbedring af detektering af checkpoints kunne være at anvende mulighederne for navigering i et koordinatsystem, som er tilgængelige i LeJOS API'en. På denne måde ville vi måske være i stand til at definere lokationer for, hvor de forskellige checkpoints var placeret og derved kunne robotten huske på, at den havde været forbi et bestemt checkpoint og derved også være i stand til at ignorere dette, såfremt den vendte tilbage til det. En ulempe ved at anvende et koordinatsystem ville være, at banen ikke kunne konstrueres særlig dynamisk, da vi på forhånd ville være nødt til at vide, hvor de forskellige checkpoints var.