\subsection{Kodestil}

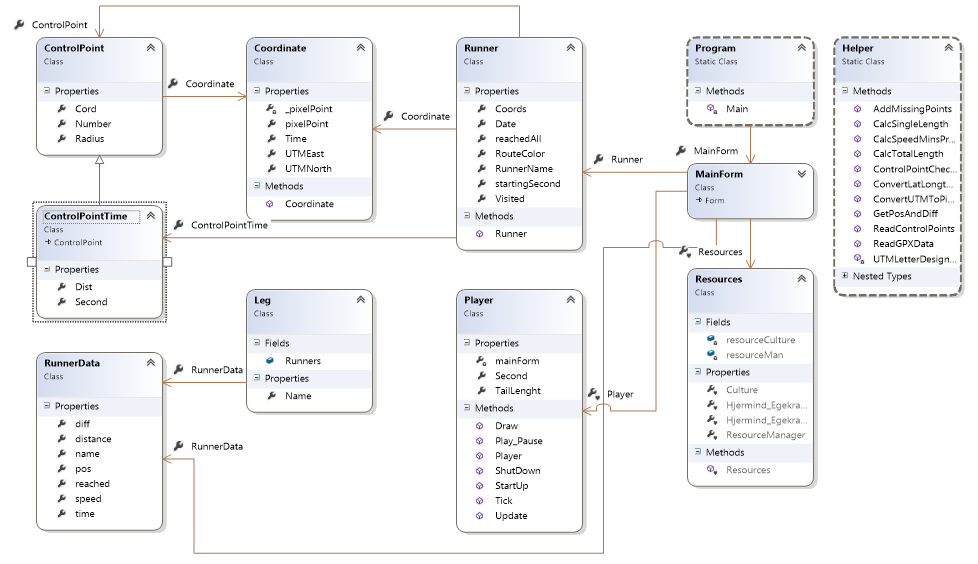
(Noget om C#)

(Noget om navngivning)

(Noget om nedarvning)

\subsection{Klasse beskrivelse}

Gruppen har vha. Visual Studio udarbejdet et klasse diagram, for at give et bedre overblik over programmets klasser. I firkanterne i klassediagrammet ses klassens navn, dens fields, properties og metoder.



\begin[itemize]

\item \textbf{Converter} MANGLER(Tidligere Helper class).

\item \textbf{Coordinate} indeholder koordinater for at kunne se hvor løberen er og hvor posterne er placeret, som både pixel- og UTM koordinater.

\item \textbf{ControlPoint} repræsentere posterne i løbet. Hvor den indeholde en posts radius, dens nummer og henter dens koordinater fra Coordinates.

\item \textbf{ControlPointTime} har nedarvet fra ControlPoint, hvor den så implementere Second og Dist som vil være tid i sekunder og distancen fra løberren til posten.

\item \textbf{Runner} indeholder informationer om en løber, dette er bl.a. løberens koordinater på ruten, om løberen har besøgt alle poster og løberens navn.

\item \textbf{RunnerData} gemmer på en mængde data om hver enkelt løber, som hastighed, distance løbet og position i løbet, hvor dette vil være de informationer som kan findes under tabben som gruppen har kaldt ”Data view”.

\item \textbf{Leg} er det som førnævnt kaldet et ”stræk”, dog nu oversat til engelsk. Leg indeholder en list af RunnerData samt et navn på strækket.

\item \textbf{Player} sørger for at tegne løberen på kortet, samt den hale som skal være efter løberen. Derudover viser, skjuler og opdatere den forskellige funktioner i programmet når det køres, dette er bruges i den tab som gruppen har kaldt ”Map view”.

\item \textbf{MainForm} MANGLER

\item \textbf{Resources} indeholder kortet som bruges I programmet, samt worldfilen der indeholder nogle koordinater. Dette bruges til at få koordinaterne fra kortet til at passe med koordinaterne fra løberen.

\item \textbf{Program} i denne klasse bliver programmet kørt fra.

\end[itemize]

\subsection{Brugervenlighed}

I gruppens problemformuleringen lød en af underpunkterne ” Hvordan kan løsningen gøres brugervenlig? ”. Gruppen har taget udgangspunkt i Rolf Molichs råd om brugrevenlighed, som har et speciale i software engineering: [http://www.dialogdesign.dk/Om\_brugervenlighed.htm].

\begin[itemize]

\item \textbf{Nyttigt} – Programmet skal løse de opgaver, som brugeren ønskes løst.

\item \textbf{Let at lære} – Programmet skal være let at lære at benytte, hvor indlæringstiden for at lære at løse bestemte opgaver skal være så kort som mulig.

\item \textbf{Let at huske} – Her menes det at genindlæringstiden skal være så lav som muligt, for de brugere der har været væk fra programmet i en længere periode eller ikke bruger det så ofte.

\item \textbf{Effektivt at bruge} – Programmet skal køre og løse opgaver så hurtigt muligt. Der skal være så få fejl som muligt, hvis det skulle ske der forekommer en fejl, så skal kvaliteten af fejl meldingen være god.

\item \textbf{Rart at bruge} – Dette er brugeren totaloplevelse af programmet, hvad deres mening er om det.

\end[itemize]

Rolf Molich nævner at alle disse egenskaber kan måles objektivt, hvor han giver eksempler som med et stopur eller spørgeskemaundersøgelser. Det program som gruppen har udviklet, er derfor prøvet at lave brugergrænsefladen så simpel og overskuelig som overhovedet mulig, samt løse opstående fejl. Da dette dog er det første store program som gruppen har lavet, har gruppen taget den beslutning at fokusere mere på at funktionerne i programmet til at virke, frem for at tænke på brugervenlighed. Dette gøres pga. manglende erfaring med større programmer og den lille mængde tid der er til projektet. Brugervenlighed er dog stadig vigtigt i et godt program, så det vil gøres så simpelt som muligt.