

6

KAPITEL SEKS

Hvor er du?

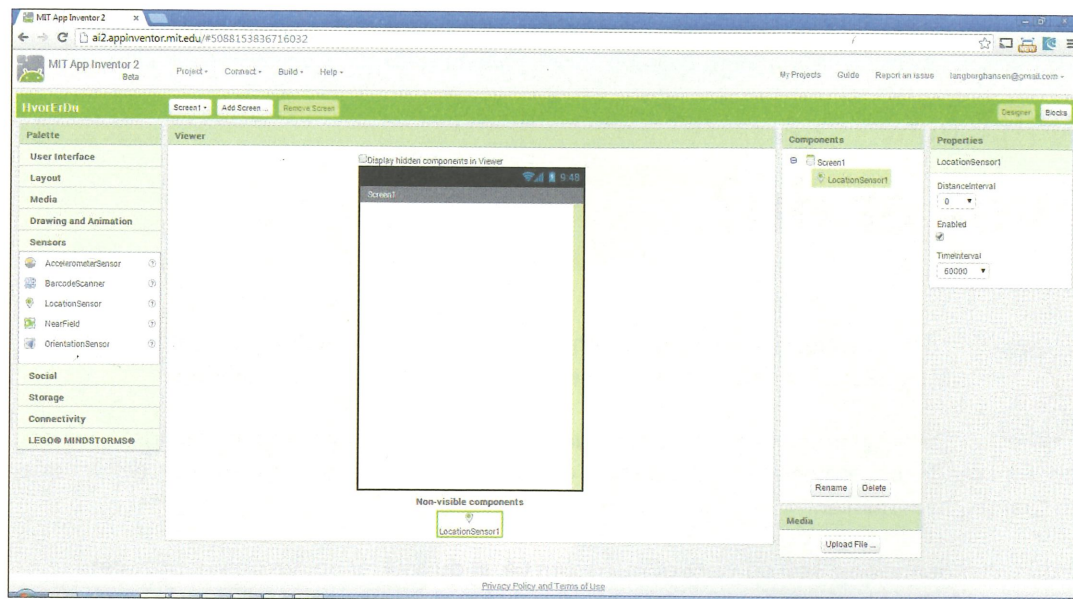


I dette kapitel laver vi en app, som udnytter telefonens mulighed for at bruge GPS. Med GPS kan vi få at vide, hvor vi er – i hvert fald med en vis nøjagtighed.

Det vil vi bruge til at vise den aktuelle placering på et kort. Appen skal også have en knap, som sender den nuværende adresse som en SMS. Så kan man hurtigt svare, hvis der er nogen, som spørger, hvor man er.

Når du tester denne app, skal du være opmærksom på, at du ikke får nogen oplysninger om placeringen, hvis du kører appen i emulatoren. Hvis du kører appen på din telefon, skal du vælge at installere den på telefonen, før du kan se din placering. Det er ikke nok at køre den gennem AI2 Companion-appen. Se mere om dette i kapitlet *I gang med App Inventor*.

Placering



Opret et nyt projekt. Gå til designeditoren.

Telefonens GPS-muligheder er repræsenteret ved komponenten **LocationSensor**. Den komponent ligger i gruppen **Sensors** i paletten.

Når du trækker komponenten ind på skærmen, bliver den placeret i sektionen **Non-visible components** under skærmen. GPS-mulighederne er ikke noget, brugeren kan se, men det giver os nogle spændende muligheder i blokeditoren.

I designeditoren har **LocationSensor** ikke ret mange egenskaber. Det primære er, om den er slået til. Det styres ved fluebenet i feltet **Enabled**. Det er en god ide, at starte GPS'en tidligt i appen, fordi det tager noget tid, før den får det første signal og kan fortælle, hvor man er. På den anden side bruger GPS'en også meget strøm, så man skal heller ikke starte den alt for tidligt.

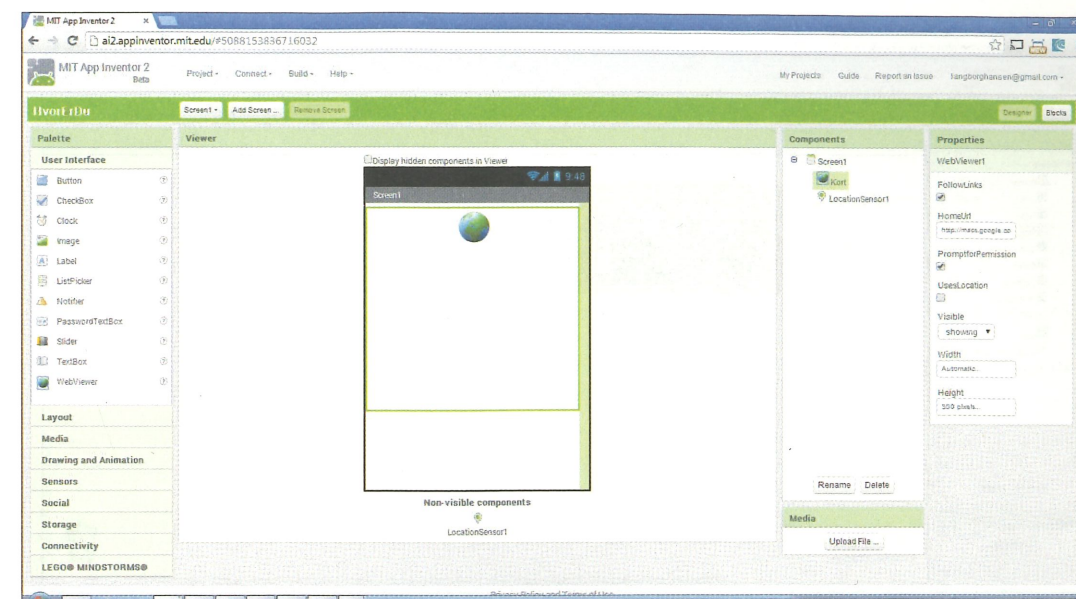
Hvis brugeren ikke har GPS på telefonen, eller der ikke kan findes et GPS-signal, sørger **LocationSensor**-komponenten automatisk for at bruge andre metoder til at finde placeringen – det kan være trådløse netværk eller mobilmaster.

Appen søger automatisk efter placering i det interval, som angives i **TimeInterval**. De 60.000, der står i feltet som udgangspunkt, betyder 60 sekunder og er som regel en passende værdi.



Du finder den færdige app ved at scanne QR-koden her eller ved at gå ind på denne side: <http://libr.is/ijqlc>. Hvis du ikke har en QR-scanner på din telefon, kan du fx hente QR Droid i Google Play Butik.

Google Maps



Google Maps er en webtjeneste, og det betyder, at vi kan vise den i vores app som en almindelig webside. Opgaven bliver at få det rigtige kortudsnit vist, men den del klarer vi i blokeditoren senere. I designeditoren behøver vi kun at lave en komponent, der kan vise en webside.

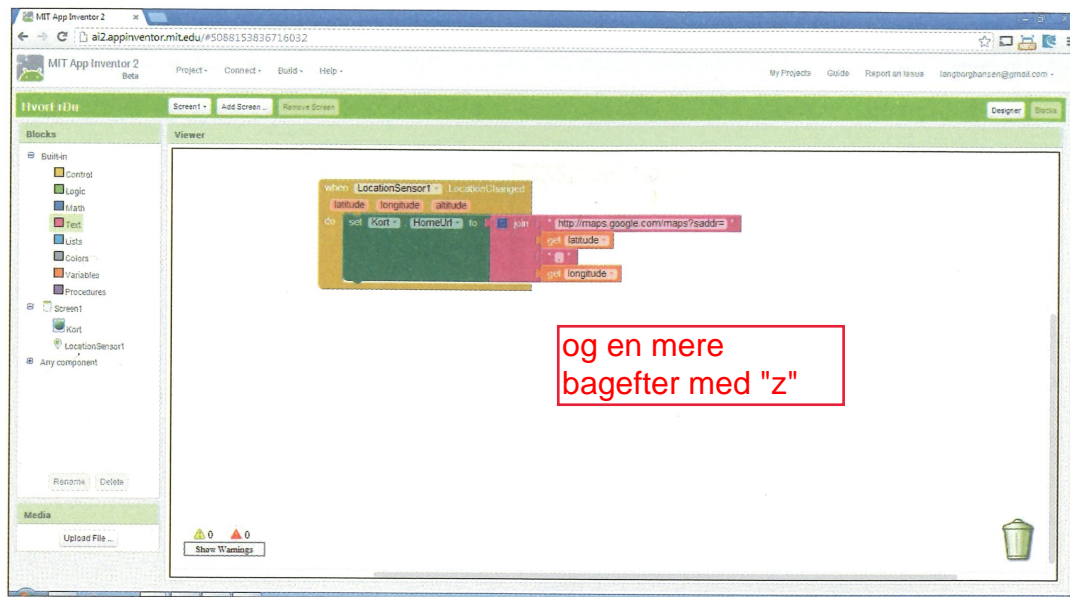
Den komponent hedder **WebViewer**, og den findes i **User Interface**-gruppen i paletten. Komponenten bruger automatisk hele skærmens bredde, men ikke så meget af højden. For at give plads til at vise kortet skal komponenten være noget højere – eksempelvis 300 pixels. Det ændres ved at klikke på feltet **Height** i **Properties**-listen, når **WebViewer**-komponenten er valgt.

Den præcise adresse på den hjemmeside, der skal vises, afhænger som nævnt af, hvor brugeren befinder sig. Til at starte med viser vi derfor bare forsiden af Google Maps. Det gør vi ved at skrive adressen **http://maps.google.com** i feltet **HomeUrl**.

Da denne **WebViewer** skal bruges til at vise kortet, omdøber vi den til **Kort**.

I feltet **FollowLinks** kan du bestemme, om brugeren må skifte væk fra den side, du har valgt at vise ham. Hvis der er flueben i feltet, kan brugeren følge links på siden og komme til en anden side. Er der ikke flueben ved feltet, kan brugeren kun se de sider, du viser ham.

Vis placering



For at vise placeringen på kortet skal vi bruge telefonens aktuelle GPS-koordinater. Marker **LocationSensor1**. Vælg blokken **when LocationSensor.LocationChanged**. Indholdet af denne blok bliver udført, hver gang telefonen flyttes. Den bliver udført første gang, når telefonen får GPS-signal. Blokken kommer med tre parametre, **latitude**, **longitude** og **altitude**, som fortæller breddegrad, længdegrad og højde.

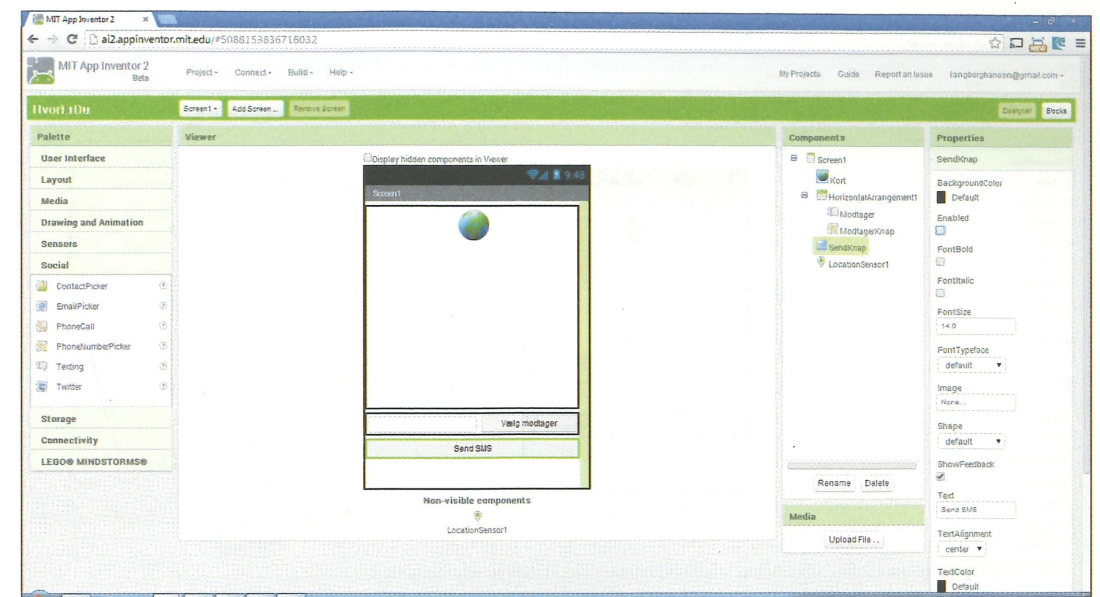
Tilføj blokken **set Kort.HomeUrl to**. Du finder denne blok ved at have **Kort**-komponenten markeret.

Tekster samles med blokken **Join**, som ligger i **Text**-gruppen. Forbind **Join**-blokken til blokken **set Kort.HomeUrl to**. Vi skal i alt sætte fire tekster sammen, så det er nødvendigt med flere end de to åbninger, **Join**-blokken normalt har. Ved at klikke på det blå kvadrat i **Join**-blokken kan du føje yderligere åbninger til blokken ved at trække **String**-blokke fra venstre side til højre side af vinduet.

For at vise den rigtige position på kortet skal vi have URL'en ændret på **WebView**-komponenten. Med Google Maps kan den rigtige position vises ved at føje bredde- og længdegrad til URL'en **http://maps.google.com/maps?saaddr=**. For at tilføje dette laver vi en ny tekst ved at tage en tom ""-blok fra **Text**-gruppen. I feltet i blokken skriver vi ovennævnte adresse. Dernæst skal du holde musen over den røde boks med teksten **latitude** og vælge **get latitude** og trække den hen til **Join**-blokken. Derefter henter du endnu en tekst-blok, skriver et komma og henter **longitude**-parameteren.

<https://maps.google.com/maps/@>

Brugergrænseflade til SMS



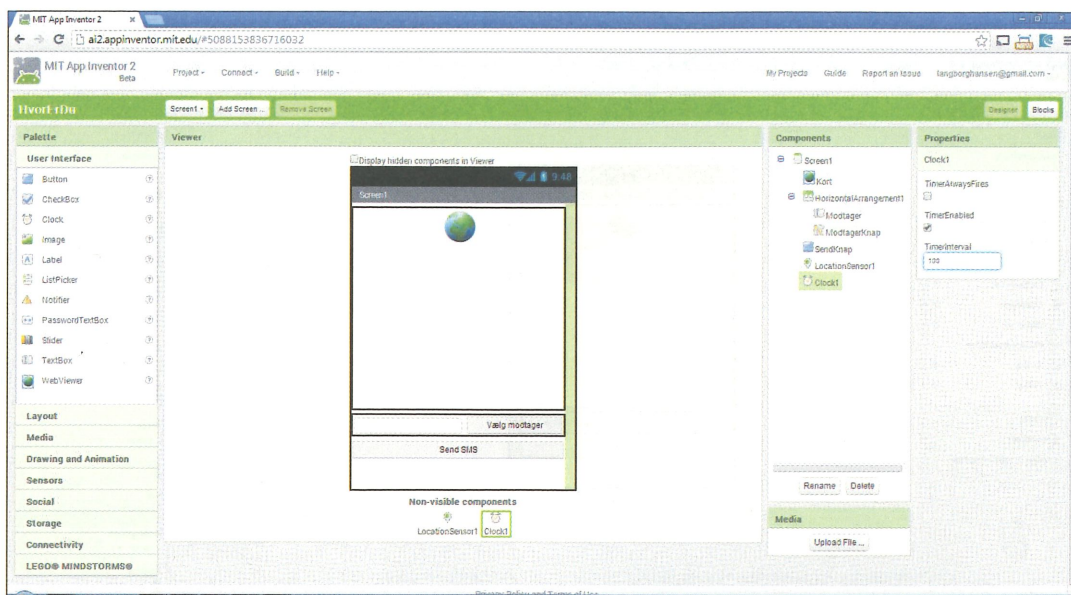
Appen skal også sende en SMS med den aktuelle placering. Derfor skal vi have tilføjet en knap til at sende SMS og mulighed for at vælge modtager på brugergrænsefladen. Modtageren skal kunne vælges fra kontaktbogen eller ved at indtaste telefonnummeret i et tekstfelt. Vi skal bruge følgende komponenter i designeditoren:

- En **PhoneNumberPicker**, som du finder under **Social**. Kald den **ModtagerKnap** under **Components** og ændr **Text** under **Properties** til "Vælg modtager".
- En **TextBox**, som du finder under **User Interface**. Kald den **Modtager**. Da telefonnummeret kun består af tal, skal du sætte flueben ved egenskaben **NumbersOnly** i højre side.
- En **Button**, som du finder under **User Interface**. Kald den **SendKnap** under **Components** og ændr **Text** under **Properties** til "Send SMS".
- En **HorizontalArrangement**, som du finder under **Layout**.

Placer komponenterne **ModtagerKnap** og **Modtager** indeni **HorizontalArrangement** for at få dem ved siden af hinanden. For at få gruppen med **Modtager** og **Modtagerknap** til at være lige så bred som **SendKnap** er bredden sat til **Fill parent** på **SendKnap**, **HorizontalArrangement** og **ModtagerKnap**.

SendKnap skal som udgangspunkt ikke kunne vælges, fordi der ikke er indtastet eller valgt et telefonnummer. Derfor skal du fjerne fluebenet fra dens egenskab **Enabled**. I blokeditoren sørger vi for at gøre knappen aktiv på det rigtige tidspunkt.

Clock-komponenten



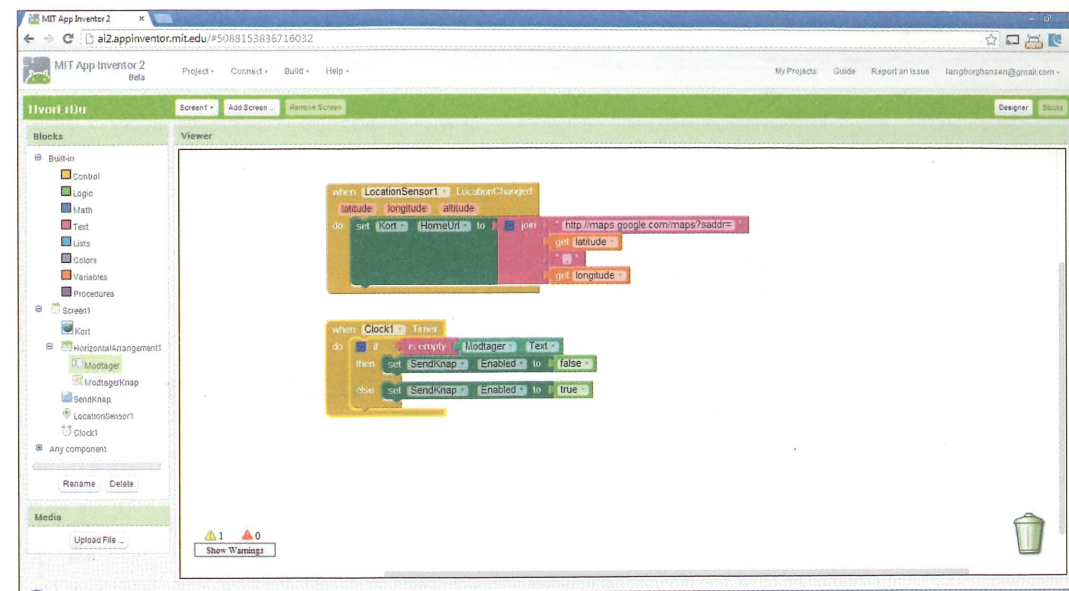
Som nævnt vil vi gerne have, at **SendKnap** kun er aktiv, når brugeren har indtastet eller valgt et telefonnummer. Til det har vi brug for hele tiden at holde øje med, om der er tekst i tekstfeltet. Når vi skal gentage noget med et fast interval, bruger vi en **Clock**-komponent, som findes i designeditoren i **User Interface**-sektionen.

Clock-komponenten lægges i gruppen **Non-visible components** under skærmen, når den føjes til brugergrænsefladen. Komponentens har to formål. Dels kan den fortælle det aktuelle tidspunkt, og dels kan den gentage opgaver med et givet interval. Her vil vi bruge den sidste mulighed til at kontrollere, om der er tekst i tekstfeltet, 10 gange i sekundet.

Det er værdien i feltet **TimerInterval**, der bestemmer, hvor lang tid der skal gå mellem hver udførelse af timeren. Værdien angives i millisekunder, så 10 gange i sekundet svarer til værdien 100. Med **TimerEnabled** kan man styre, om timeren skal være slået til fra start; det skal den i vores tilfælde.

Den sidste egenskab, **TimerAlwaysFires**, angiver, om timeren også skal udføres, selvom appen ikke bliver vist. I vores tilfælde kan der ikke komme tekst i **Modtager**-feltet, hvis appen ikke bliver vist, så vi slår den mulighed fra ved at fjerne fluebenet fra **TimerAlwaysFires**. På den måde sparer vi også lidt strøm på telefonen.

Aktivering af knappen



I blokeditoren findes blokken **when Clock1.Timer** under **Clock**-komponenten. Det er indholdet af denne blok, der bliver udført, hver gang det angivne interval er gået. Træk denne blok ind på hovedområdet.

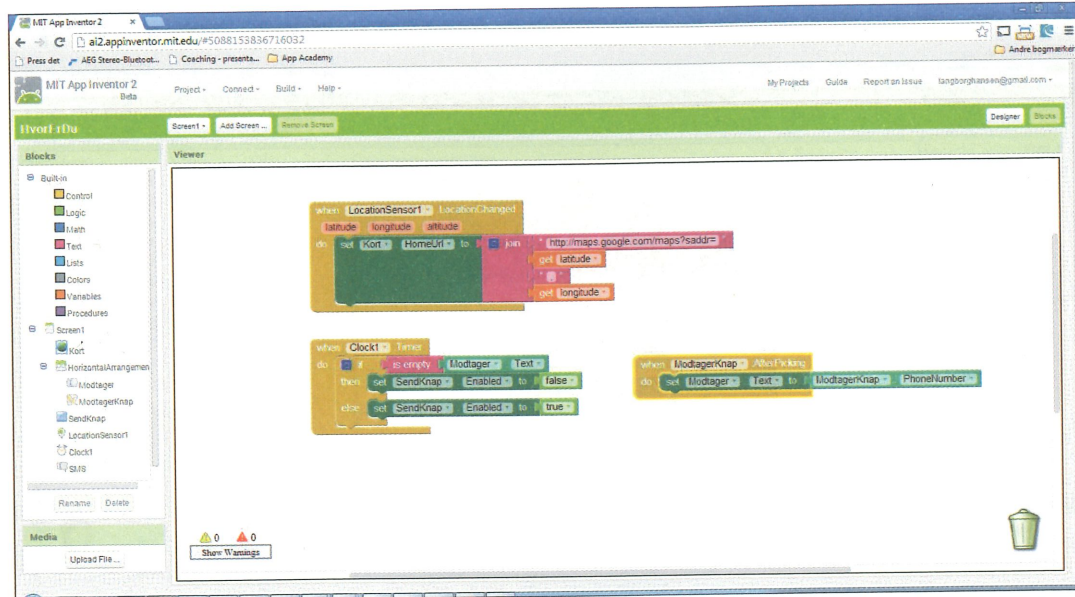
I blokken vil vi gerne gøre følgende: *Hvis feltet Modtager er tomt, så deaktivér SendKnap – ellers aktivér SendKnap.*

Da vi både har et *hvis*, et *så* og et *ellers*, laver vi en **If-Then-Else**-blok ved at tage **If-Then**-blokken fra **Control**-gruppen og tilføje en **Else**-sektion i dens konfigurator, som åbnes ved at klikke på det blå felt i blokken. Tilføj denne blok til **when Clock1.Timer**-blokken.

I **If**-delen af blokken skal vi undersøge, om feltet **Modtager** er tomt. Det er det, hvis dets tekst er tom. I **Text**-gruppen har vi blokken **is empty**, som netop undersøger, om en tekst er tom. Den tekst, der er indtastet i feltet, findes i blokken **Modtager.Text** når du markerer **Modtager**-komponenten. Ved at forbinde de to og sætte dem på **If**-åbningen af **If-Then-Else**-blokken har vi bygget vores betingelse.

Sektionerne **then** og **else** skal næsten være ens. Sektionen **then** skal gøre **SendKnap** inaktiv, **else** skal gøre knappen aktiv. Begge dele gøres med blokken **set SendKnap.Enabled to**. Denne blok skal forbindes med **false** fra **Logic**-gruppen for at gøre knappen inaktiv, og blokken **true** for at gøre den aktiv.

Valg af telefonnummer



Nu virker aktiveringen af **SendKnap**, når man indtaster telefonnummeret i tekstfeltet, men hvad hvis brugeren i stedet vælger modtageren fra telefonbogen?

I det tilfælde er vi nødt til at kopiere det valgte telefonnummer over i **Modtager**-feltet, så vi får knappen aktiveret. Det har også den fordel, at brugeren kan se, hvilket nummer beskeden bliver sendt til.

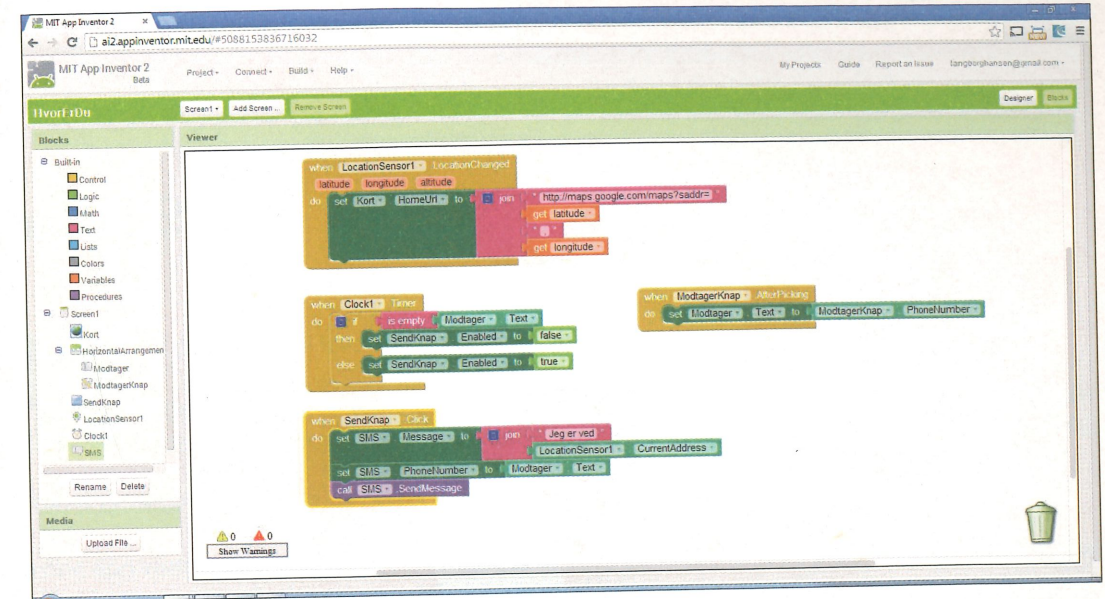
Når brugeren har valgt et telefonnummer med **PhoneNumberPicker**-komponenten, bliver blokken **AfterPicking** udført. I denne blok kan vi altså sætte de blokke, som kopierer det valgte telefonnummer over i **Modtager**-feltet.

Marker **ModtagerKnap** og træk **when ModtagerKnap.AfterPicking do**-blokken ind i hovedområdet.

Teksten i **Modtager**-feltet ændres med blokken **set Modtager.Text to**, som findes ved at markere **Modtager**-komponenten.

Det valgte telefonnummer fra **ModtagerKnap**-komponenten ligger i blokken **ModtagerKnap.PhoneNumber**. Ved at forbinde den til **set Modtager.Text to** og lægge de forbundne blokke ind i **when ModtagerKnap.AfterPicking**, bliver telefonnummeret kopieret, når brugeren har valgt det.

Send SMS



Som tidligere skal vi bruge en **Texting**-komponent i designeditoren for at kunne sende SMS-beskeder fra appen. Kald den SMS. I denne app skal vi ikke modtage beskeder – derfor skal **ReceivingEnabled** sættes til **Off**.

Det spændende i denne app er ikke at sende SMS'en, men at finde ud af, hvad der skal stå i den. Når vi skal sende vores position til andre, er det ikke nok at sende længde- og breddegrad; det er der ikke mange mennesker, der kan aflæse. Heldigvis kan **LocationSensor**-komponenten oversætte koordinaterne til en adresse for os. Det sker med blokken **LocationSensor.CurrentAddress**. Opgaven lyder altså således: Når brugeren trykker på SMS-knappen, så lav en besked med teksten "Jeg er ved" efterfulgt af den aktuelle adresse, og send den til det telefonnummer, brugeren har valgt.

I blokeditoren tager vi fat i blokken **when SendKnap.Click**. I den lægger vi blokken, der angiver indholdet af SMS-beskeden, **set SMS.Message to**. Indholdet består af to tekster, indledningen og adressen, derfor bygger vi den ved hjælp af en **Join**-blok fra **Text**-gruppen. I **Join**-blokken sættes først en **Text**-blok med indledningen ("Jeg er ved"), og dernæst **LocationSensor1.CurrentAddress**-blokken.

Modtagerens telefonnummer skal sættes til indholdet af **Modtager**-feltet. Det gøres ved at forbinde blokkene **set SMS.PhoneNumber to** og **Modtager.Text**.

Til sidst sendes SMS'en med blokken **call SMS.SendMessage**.