

# Att göra lärappar tillgängliga för barn med synnedsättning

Charlotte Magnusson, Certec, Institutionen för Designvetenskaper, Lunds universitet

---



# Innehåll

Bakgrund .....	3
Vad är problemet? .....	3
Det behöver inte vara svårt.....	6
Några exempel som inspiration .....	7
Exempel 1. En stor bild med bildbeskrivning .....	7
Exempel 2. Flera bilder med bildbeskrivning .....	8
Exempel 3. Information i en ljudfil .....	9
Exempel 4. Fler sätt att ange antal (utmaning).....	10
Exempel 5. Geometri.....	11
Exempel 6. Stavning.....	12
Exempel 7. Skaka telefonen .....	13
Exempel 8. Knacka på skärmen .....	14
Exempel 9. Dra objekt.....	15
Exempel 10. Peka och tippa .....	16
Exempel 11. Tippa och rör vid skärmen.....	17
Exempel 12. NFC-taggar – välj ett objekt.....	18
Exempel 13. NFC-taggar – namnge objekt .....	18
Sammanfattning.....	19

# Bakgrund

Det finns idag ett ofantligt utbud av så kallade lärappar, av varierande kvalitet. Nyttan med alla dessa appar kan dock ifrågasättas. Exempelvis säger Agneta Gulz, professor i kognitionsvetenskap:

*”Av de pedagogiska appar och digitala läromedel som används i Sverige är det kanske endast 2–3 procent som ger tillräckligt med feedback. (...) Det är inte bra om ett barn som lär sig matematik tror att två gånger tre blir sex, men att tre gånger två däremot inte blir sex. Det finns exempel på välspredda pedagogiska appar som har sådana här fel.” (Radio P4 Malmöhus)*

För att lära behöver eleven få direkt och tillräcklig återkoppling. Om hen gör fel, måste det vara tydligt varför svaret var fel. Så mycket som 95 procent av de tillgängliga lärapparna är dock snarare testappar – det vill säga testverktyg för att testa redan erhållna kunskaper – men den upprepning som testandet innebär kan ändå vara värdefull. Och för den typ av kunskap som närmast måste nötas in – som exempelvis stavning och enkel matematik – ger dessa appar en möjlighet att göra den nödvändiga ”nötningen” roligare.

De flesta lärappar är mycket visuella, vilket gör dem attraktiva för seende barn men skapar problem för de barn som har synnedsättning. För att ge barn med synnedsättning likvärdiga möjligheter som seende barn att ta del av denna typ av inläring är det viktigt att apparna anpassas till och görs tillgängliga för barn med synnedsättning.

Eftersom utbudet av appar är enormt, och nya appar ständigt utvecklas, skulle en översikt över hur bra olika appar är nästan omedelbart blir inaktuell. Därför har vi i denna rapport valt att fokusera på råd, information och inspiration för den som beställer eller utvecklar nya lärappar.

## Vad är problemet?

Synnedsättning innebär svårigheter att även med glasögon och god belysning läsa text och ta del av bilder och annan grafisk information. En del personer med synnedsättning kan ändå – om informationen har en lämplig utformning – ta del av visuell information, medan andra behöver få informationen via andra sinnen som hörseln eller känseln.

Dagens smartphones och surfplattor har en så kallad skärmläsare inbyggd, vilket gör att också den som är blind i princip kan använda denna typ av teknik. Jag skriver ”i princip”, för hur bra detta fungerar beror helt på hur väl olika appar är utformade. Grundfunktionaliteten i telefonen eller plattan kommer man åt, men tyvärr är många appar utformade på ett sådant sätt att de fungerar dåligt ihop med skärmläsaren.

Ett problem med skärmläsaren för den som utvecklar appar är nämligen att den ändrar hur gränssnittet fungerar. Grundprincipen med skärmläsaren är att du letar reda på det element (knapp, länk etc.) som du vill aktivera, antingen genom att föra fingret över skärmen eller (vanligast) genom att svepa i sidled för att stega igenom de olika skärmelement som finns. När du hittat det element du söker ”dubbelknackar” du någonstans på skärmen för att aktivera. Detta gör till exempel att gester som innebär att man pekar eller drar med ett finger på skärmen fungerar olika med och utan skärmläsare – skärmläsaren lägger så att säga ”beslag på” många gester. Läs mer om iPhone’s skärmläsare VoiceOver på <http://www.apple.com/accessibility/ios/voiceover/> . Också Android har en inbyggd skärmläsare, TalkBack, som fungerar på liknande sätt.

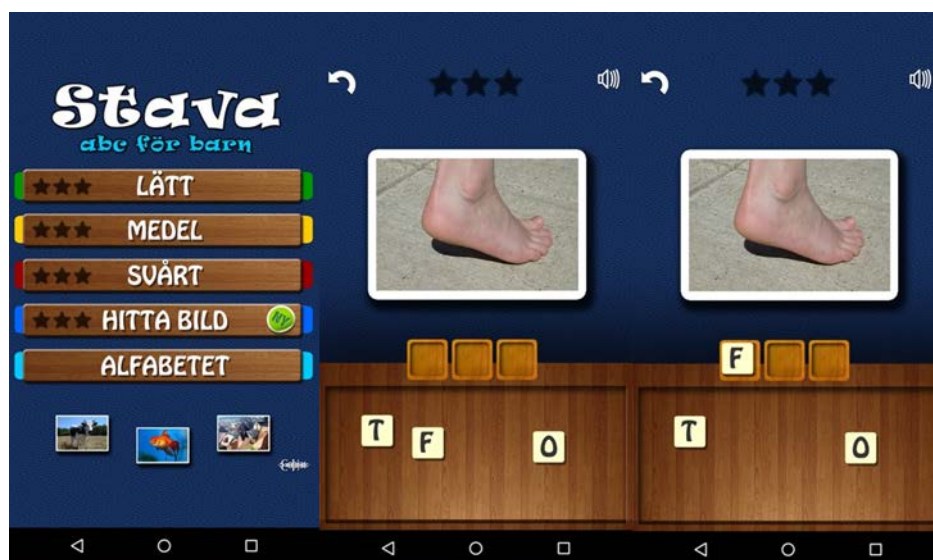
För att illustrera några vanliga fel visar vi två exempel. Det första kommer från appen ”King of Math (junior)”. Appen använder en hel del standardkontroller och på startsidan går det att stega omkring och få reda på vad det finns för information på sidan. Det går också att starta appen och välja vilken sorts problem man vill öva på. Appen har dessutom lite musik i bakgrunden som förhöjer upplevelsen. När man däremot startar en samling matteuppgifter så startar tyvärr också problemen. Den som ser får en nedräkning ”3, 2, 1 ...” med en trumvirvel – men den som inte ser siffrorna vet inte riktigt vad som händer. Detta kunde man kanske leva med, om det inte hade varit så att det dessutom är helt omöjligt att ta del av den information som krävs för att lösa uppgifterna.



Skärmdumpar från appen ”King of Math”. Startsidan (tv) och de sidor där man väljer typ av uppgift (”bok” och ”kapitel”) går bra att använda. I uppgifterna är tyvärr själva uppgiftsinformationen dock osynlig för skärmläsaren – bilden i mitten och bilden till höger

kommer direkt efter varandra när du stegar dig fram med skärmläsaren. Och klickar du på bilden händer ingenting.

Tyvärr ligger nämligen själva uppgiftsinformationen som en bild som helt saknar beskrivning och som skärmläsaren helt enkelt inte vet om att den finns. Som synes i exemplet ”King of Math” är det inte heller självklart hur bildinformationen ska utformas – skulle någon exempelvis lägga in ”tre bollar” som bildtext, så blir uppgiften meningslös (mer om detta senare).



Skärmdumpar från appen ”Stava”. Du drar bokstäverna på plats för att stava ordet.

Vårt andra exempel kommer från appen ”Stava”. Denna app fungerar på ett sätt som kan fungera bra också för barn med nedsatt syn, och som skulle vara ganska lätt att anpassa till en skärmläsare. Utan skärmläsare väljer du först nivå och får sedan en uppgift. När du rör vid en bokstav ljudas den, och när alla bokstäverna är på plats läses ordet upp.

Tyvärr är interaktionen dock implementerad på ett sätt som inte alls fungerar med skärmläsaren. På första sidan går det inte att välja nivå – men det går att ”dubbelknacka” på skärmen och då kommer man till en uppgift (verkar vara en på ”lätt” nivå). Sedan är det tyvärr stopp. Det finns ingenting som du kan komma åt med skärmläsaren – det enda du kan göra är att lämna appen.

Den som utvecklar appar för personer med synnedsättning måste alltså tänka både på hur appen kan göras visuellt tydlig (stor text, tydliga bilder, bra kontrast), hur ljud och musik kan användas *och* hur appen fungerar ihop med skärmläsaren.

## Det behöver inte vara svårt

Som utvecklare måste man bekanta sig med skärmläsaren. Sätt på den och prova. Och kom ihåg att ”dubbelknacka” för att aktivera. Tänk också på att vana användare oftast inte letar efter saker på skärmen utan istället sveper (”flick”) för att stega igenom vad som finns. Du kommer att märka att ganska mycket faktiskt fungerar och att standardkomponenter som knappar, texttrutor och liknande för det mesta fungerar rakt av. Vanliga problem du kommer att stöta på är att någon gjort bildknappar utan att ange en bildbeskrivning, eller att till exempel ”drag and drop” inte fungerar. Det kan också hända att du startar en app och där helt saknar saker att aktivera (trots att det finns grafik som du kan interagera med om skärmläsaren är av).

Det positiva med detta är att så länge du håller dig till standardkomponenter, och ser till att bilder som betyder något (t.ex. bildknappar) har en beskrivning, så kommer din app troligtvis att fungera mer eller mindre rakt av.

För det visuella gränssnittet bör du använda stor och tydlig grafik med god kontrast. Som vid all utformning bör du vara medveten om att inte alla ser färg likadant – förlita dig därför aldrig enbart på färg för att förmedla information (detta innebär inte att du ska undvika att använda färg – bara att färgen måste kompletteras med form, position eller något annat så att också den som inte kan urskilja färgen får en chans att ta till sig informationen). För att testa den visuella delen av gränssnittet kan verktyget ”**Cambridge simulation glasses**” vara till stor hjälp: <http://www.inclusivedesigntoolkit.com/betterdesign2/csg/csg.html> . Detta verktyg gör det möjligt att simulera olika grader av svårigheter med att se detaljer. Mer avancerad fingerad optik finns exempelvis att köpa från **Iris hjälpmedel**: <http://www.irishjalpmedel.se/Simulering-sv-1282/pm> .

En första snabb checklista för vad som behövs för att utveckla appar tillgängliga också för personer med synnedsättning blir:

- Lär dig använda skärmläsaren.
- Använd stor text och tydliga bilder med god kontrast (undvik små detaljer). Tänk på att alla inte ser färg likadant.
- Använd gärna standardkomponenter (gör du egna måste du kontrollera att de fungerar även med skärmläsare).
- Se till att alla bilder som betyder något har en bildbeskrivning (som inte löser uppgiften).
- Använd gärna ljud och musik för att göra appen roligare.
- Testa regelbundet din app med skärmläsare och simulerad optik under utvecklingen.
- Se till att också en del experter (personer med synnedsättning) testar din app.

Vill du nå längre har Apple också en hel del dokumentation för utvecklare på <https://developer.apple.com/accessibility/>

Motsvarande länk för Android är

<http://developer.android.com/guide/topics/ui/accessibility/index.html>

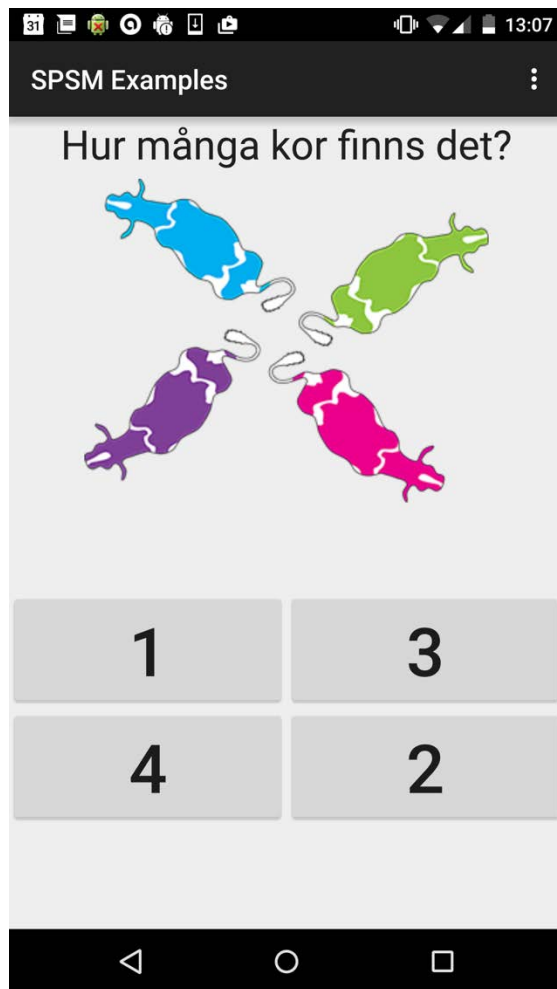
## Några exempel som inspiration

Eftersom det är vanligt med otillgänglig grafik och till exempel ”drag and drop” i många lärappar har vi valt att samla ett antal exempel avsedda att ge inspiration till hur man kan tänka när man skapar en lärapp som ska vara tillgänglig även för barn med synnedsättning. Exempelen finns implementerade på Android, men den interaktion de illustrerar går att implementera även på iOS. De inledande exemplen är sådana som utnyttjar standardkomponenter, men efter hand blir exemplen mer avancerade och utnyttjar olika typer av sensorer (därmed krävs givetvis att enheten har just dessa sensorer). Vi avslutar med två exempel som visar hur NFC-taggar kan användas för att knyta ihop fysiska föremål med en lärapp.

Observera att vi i exemplen har *fokuserat på interaktionen* – inte grafiken. Den grafik som implementerats är därför skissartad och är enbart gjord för att illustrera principen – den är alltså inte avsedd som exempel på bra grafisk utformning.

### Exempel 1. En stor bild med bildbeskrivning

Vårt första exempel visar hur man kan använda en stor bild med bildbeskrivning. Det som är viktigt att tänka på här är att man inte ger svaret på frågan i bildbeskrivningen. Om nedanstående bild hade blivit beskriven som ”fyra kor” så skulle uppgiften ha blivit ganska meningslös.



*Exempel 1. Ett bildelement med bildbeskrivningen "blå ko, grön ko, rosa ko, lila ko".*

Detta exempel använder enbart standardkomponenter (textelement, bildelement och knappar) och kräver inget extra arbete, förutom att bilden eventuellt måste anpassas så att den går bra att beskriva och bildbeskrivningen som sagt måste utformas så att den inte innehåller svaret. Det är dessutom bra om de olika elementen i bilden är tydligt identifierbara som olika – genom att ge korna olika färg gör vi det tydligt att det verkligen handlar om olika djur.

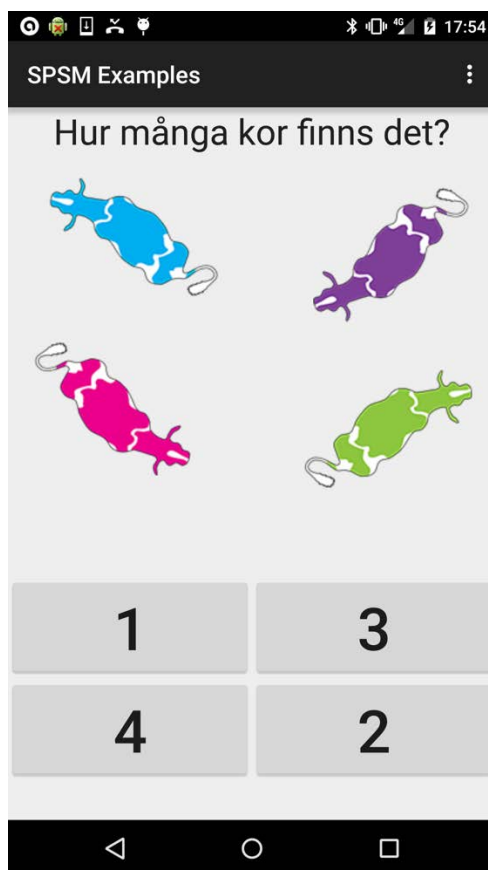
I exempelappen ges bara svaret "rätt" eller "fel", men i en riktig app kan det vara lämpligt att ge lite mer information; till exempel "rätt, det finns fyra kor" eller "tre kor är fel, rätt svar är fyra kor". Ljudeffekter som fanfar för rätt och plopp för fel är lätta att lägga till för att förhöja upplevelsen.

## Exempel 2. Flera bilder med bildbeskrivningar

Exempel 2 är visuellt förvillande likt exempel 1. Det som tekniskt skiljer är att varje ko är ett eget bildelement, vilket gör att en elev som använder skärmläsare själv får stega sig igenom vilka bilder som finns (alternativt söka igenom skärmen med fingret). Jämfört med det förra exemplet är det här ännu viktigare att korna inte bara heter "ko". Skulle du exempelvis söka



med fingret har du då ingen aning om ifall den ko du hittat är ny eller samma ko som du redan varit på. Med samma argument som i exempel 1 är det heller inte lämpligt att kalla korna ”ko 1”, ”ko 2” osv. – då ger ju bildbeskrivningen återigen svaret på uppgiften.



*Exempel 2. Fyra olika bildelement med bildbeskrivningarna ”blå ko”, ”lila ko”, ”grön ko” och ”rosa ko”.*

För övrigt gäller samma observationer som i exempel 1 – det är bra att fundera över hur informationen kring rätt och fel utformas samt att ljudeffekter kan förbättra upplevelsen.

### Exempel 3. Information i en ljudfil

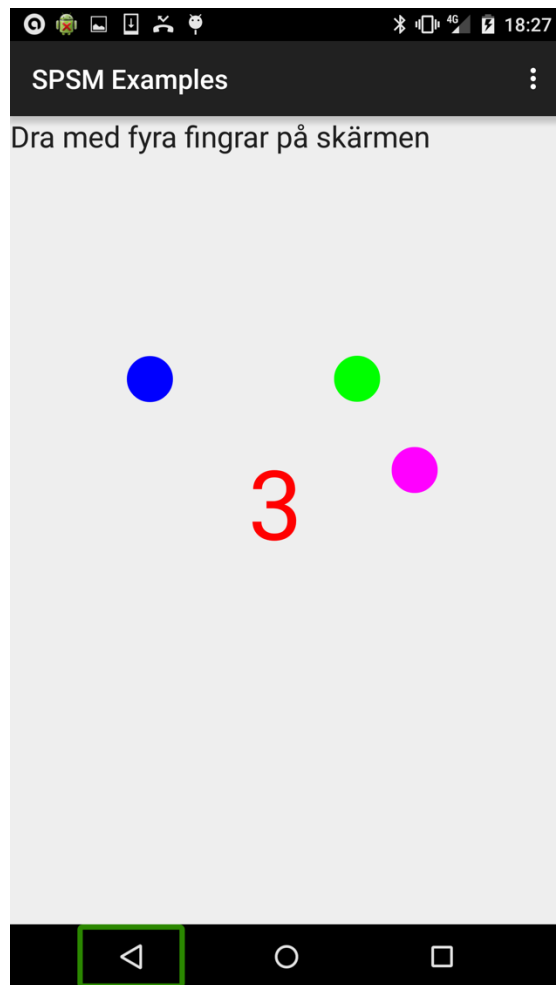
Exempel 1 och 2 utgår ifrån den visuella utformningen och ser till att det finns tillräckligt med information för att även den som inte ser ska kunna lösa uppgiften. Ett alternativ är att istället utgå ifrån ljudinformation så att både den som ser och den som inte ser använder samma information för att lösa uppgiften. Detta kan kräva lite mer fantasi, men ger också möjlighet att göra uppgiften roligare. I exempel 3 klickar du på ladan för att höra vilka djur som finns i den (höns, hund, gris och får).



*Exempel 3. När skärmläsaren kommer till bilden på ladan, som är en bildknapp, säger den "lada-knappen" (bildbeskrivningen är "lada"). En klickning på "lada-knappen" startar ljudfilen med de fyra djurlätena.*

## Exempel 4. Fler sätt att ange antal (utmaning)

Man skulle även kunna tänka sig att låta barnet ange antal genom att placera ett visst antal fingrar på skärmen. Vid test har detta dock visat sig vara svårt rent ergonomiskt om användaren inte placerar telefonen på ett bord. Dessutom kräver det mer programmering eftersom skärmläsaren "lägger beslag" på en del händelser. Medan vi kan fånga upp "enfingers"-gester (på Android via "hover") har vi ännu inte fått två fingrar att fungera – när skärmläsaren är på används nämligen två fingrar för att generera "enfingers"-händelser (används t.ex. vid scrollning). Andra möjligheter skulle kunna vara att exempelvis räkna antal "knackningar" eller skakningar. Eller (naturligtvis) att använda taligenkänning.

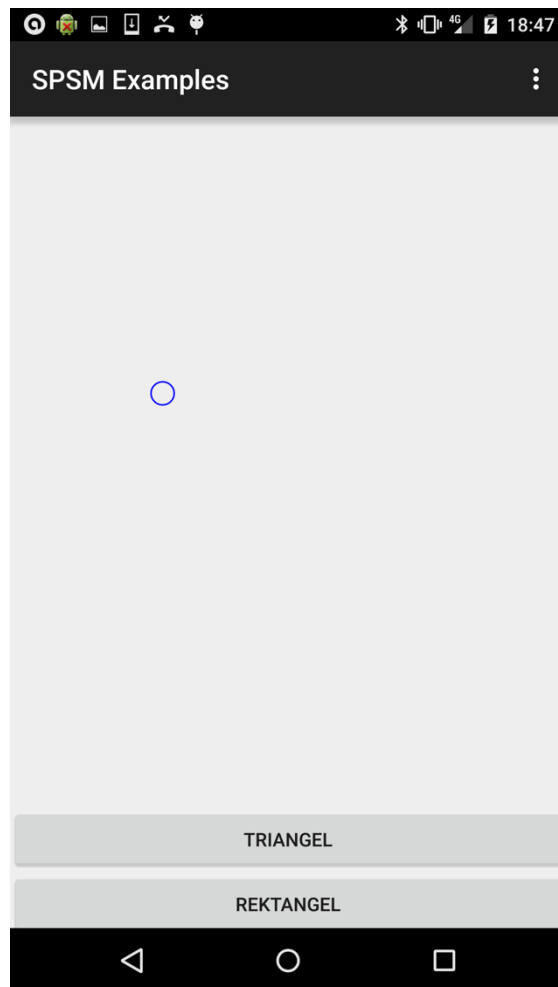


*Exempel 4. Varje färgad cirkel motsvarar ett finger. Just här är det alltså tre fingrar i kontakt med skärmen.*

Detta exempel visar inte en färdig lösning utan finns med här som inspiration och för att utmana läsaren att själv komma på fler och bättre alternativa lösningar!

## Exempel 5. Geometri

Det är svårt att ge någon som inte ser direkt tillgång till geometri via en skärm. En möjlighet för enklare figurer är att låta telefonen vibrera när användarens finger rör vid figurens kontur. Detta exempel fungerar i princip likadant både för den som ser och den som inte ser, men för den som ser var fingret befinner sig blir uppgiften lättare. Grundproblemet är att du ofta måste föra fingret fram och tillbaka över konturen, vilket gör det svårare att uppfatta den exakta formen.



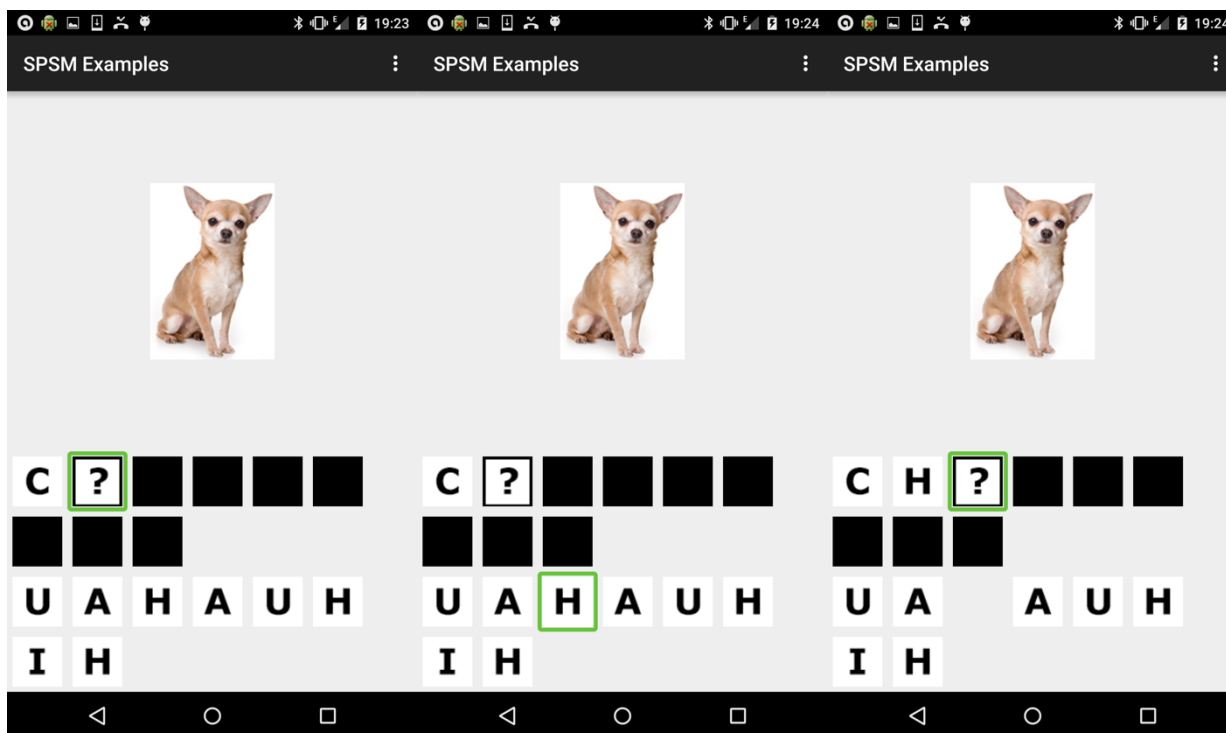
*Exempel 5. När du för fingret över skärmen vibrerar telefonen när fingret rör vid figurens kontur. Var på skärmen det vibrerar anges med en osynlig bakgrundsbild som lätt kan ändras.*

Detta exempel visar på en möjlighet att ändå förmedla enklare former via skärmen. Riktig känselåterkoppling via fysiska föremål och robotar som the PHANToM (eller framtida interaktiva material) ger dock betydligt bättre kvalitet på upplevelsen och är generellt att föredra.

## Exempel 6. Stavning

Det finns många trevliga stavningsappar. Tyvärr bygger många på att användaren ska dra bokstäver till rätt plats i ordet. Att dra något till en precis plats på skärmen utan att se är i princip omöjligt – och ”drag and drop” är generellt inte en interaktion som lämpar sig för skärmläsare (även om vi visar ett exempel som faktiskt fungerar senare). Ett alternativ är att utnyttja klickande istället. I detta exempel klickar du på den bokstav du tror kommer på nästa

plats i ordet. Om du klickar på rätt bokstav så får du höra bokstaven ljudas och den hoppar automatiskt in på rätt plats i ordet.



*Exempel 6. Hunden är en bildknapp som heter "stava mig". Trycker du på den hörs ett hundskall och sedan ordet "chihuahua". När du trycker på rätt bokstavknapp hoppar bokstaven upp i ordet (bokstaven försvinner samtidigt från de valbara bokstäverna). För att snabbare komma ner till bokstavknapparna hoppar skärmläsaren över de svarta rutorna.*

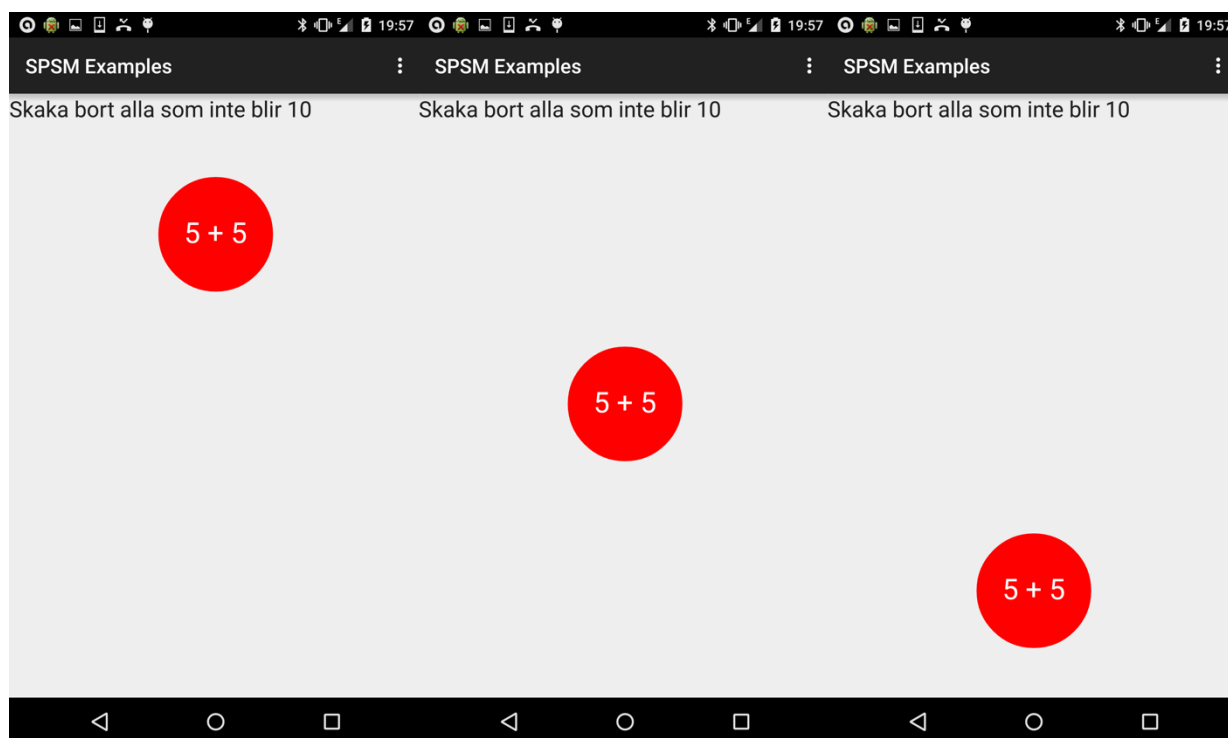
Detta exempel ger återkoppling på valda bokstäver och när ordet är färdigstavat sker återkoppling genom att hunden skäller. Bokstäverna läses upp när de väljs, och användaren kan gå tillbaka i ordet och lyssna på de delar av ordet hen redan lyckats få på plats. För att särskilja de knappar som hör till ordet heter de "bokstav 1", "bokstav 2" osv., medan de valbara knapparna som används för att stava heter "u", "a" osv.

Poängen här är att det viktiga vid stavning är att veta i vilken ordning bokstäverna kommer, inte att nödvändigtvis kunna dra dem på plats. Det fungerar lika bra att klicka på den bokstav som kommer näst i ordningen – en funktionalitet som fungerar bra också med skärmläsare. Om du gärna vill behålla "drag and drop" för dem som ser kan du alternativt låta programmet känna av om skärmläsaren är igång, och implementera "drag and drop" då skärmläsaren inte är igång.

## Exempel 7. Skaka telefonen

Detta exempel visar hur man kan använda telefonens sensorer – i detta fall accelerometern – till att skaka bort felaktiga svar. Bollar med slumpmässiga räkneuppgifter faller från toppen av

skärmen ner mot botten, och om summan inte blir 10 ska du skaka telefonen för att få bort bollen. Fallet åtföljs av en ljudillustration så att du även om du inte ser kan följa var bollen är. Felaktiga skak ger ett ”fel-ljud”, medan korrekta skak ger ett ”rätt-pling”. Bollar med korrekta summor som når botten genererar också ett rätt, medan felaktiga bollar som når botten ger ett fel.



*Exempel 7. När en ny boll dyker upp i ovankanten läser skärmläsaren upp den räkneuppgift som finns i bollen. Är summan inte 10 ska du skaka telefonen innan bollen når nedkanten.*

Naturligtvis kan man välja att istället fånga upp korrekta svar genom att låta användaren skaka telefonen. Vilket som fungerar bäst beror lite på hur problemet är utformat, men också på hur många rätta/felaktiga svar det finns. I detta exempel finns det många felaktiga svar och det är möjligt att det hade varit bättre att skaka för att fånga upp korrekta svar.

Exemplet visar också hur man kan ge ljudåterkoppling som låter användaren följa hur långt ett objekt kommit. Det är nödvändigt för denna typ av uppgift att på något sätt ge den som inte ser grafiken en uppfattning om vad som händer.

## Exempel 8. Knacka på skärmen

Ett alternativ till att skaka är att knacka på skärmen. Uppgiften fungerar exakt likadant som den i exempel 7, förutom att du istället för att skaka knackar med ett finger på skärmen (utan skärmläsare) eller två fingrar på skärmen (med skärmläsare). För att den som använder

skärmläsare ska veta vad som ska göras skickar appen ett meddelande som skärmläsaren läser upp när appen startar.

## Exempel 9. Dra objekt

Även om ”drag and drop” generellt är en olämplig interaktion för barn med synnedsättning så går den att använda, under förutsättning att målområdet är stort nog och att barnet vet var på skärmen området finns.

I detta exempel finns två mål: området för vokaler finns i ovankant på skärmen, och området för konsonanter i skärmens nedkant. Din uppgift är att dra bokstäver som dyker upp (när de dyker upp läses bokstaven upp) till rätt område. När du är i något av målområdena får du ljudåterkoppling på vilket område du är i, och kan välja att släppa om du är i rätt område. Använder du skärmläsare drar du med två fingrar, annars drar du som vanligt med ett finger. Du kan börja dra varsomhelst på skärmen eftersom den blå bollen flyttas till ditt finger (du måste alltså inte träffa den blå bollen med fingret). Släpper du bollen rätt får du ett ”rätt-ljud” och släpper du den fel får du ett ”fel-ljud”.



*Exempel 9. ”Drag and drop” kan fungera, under förutsättning att målet är tydligt även för den användare som inte ser, att det sker en tydlig återkoppling när användaren når målet och att användaren inte behöver hitta någon speciell punkt på skärmen.*

Detta exempel visar att viss typ av ”drag and drop” kan fungera. Vi har undvikit att kräva att användaren träffar en specifik punkt på skärmen genom att flytta bollen till fingret oavsett var användaren klickar. Och genom att använda bara två riktningar och ge tydlig återkoppling när användaren når målet går ”drag and drop” att hantera även utan syn. Exemplet är dock inte hundra procentigt – bokstäverna läses ofta upp för snabbt, och i en riktig app hade man fått arbeta mer på att få bättre timing. Det hade också varit lämpligt om bokstäverna lästes upp på nytt när användaren börjar dra dem.

## Exempel 10. Peka och tippa

Ett sätt att slippa ”besvär” med skärmläsaren är som vi såg i exempel 7 att använda någon av telefonens sensorer. I exempel 7 använde vi accelerometern, medan vi i detta exempel istället använder gyro och kompass. När en siffra dyker upp i det gula fältet läses den upp, så att den som inte ser istället kan lyssna på vilken siffra som är markerad. För att välja en siffra tippas du telefonen uppåt.



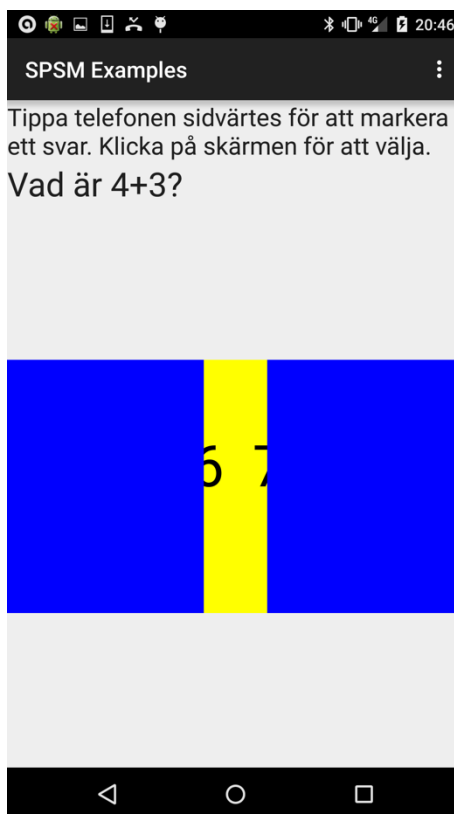
*Exempel 10. Peka telefonen i olika riktningar för att höra olika förslag på svar och markera rätt svar. Välj det markerade svaret genom att tippa telefonen uppåt.*



Återigen är återkopplingen viktig. Om inte siffrorna läses upp, blir appen omöjlig att använda för den som inte ser. En möjlig nackdel är att användaren behöver kunna peka 360 grader runt sig, vilket kan vara svårt för den som sitter ner.

## Exempel 11. Tippa och röra vid skärmen

Detta exempel liknar exempel 10, med den skillnaden att du istället tippa telefonen i sidled för att markera ett svar. För att välja ett markerat svar rör du vid skärmen (ett finger utan skärmläsare, två fingrar med skärmläsare). Precis som i exempel 10 läses siffrorna upp så att du vet vilka svarsalternativ som visas.



*Exempel 11. I detta exempel tippa du telefonen i sidled för att markera ett svar. Rör vid skärmen för att välja det markerade svaret (ett finger utan skärmläsare, två fingrar med skärmläsare).*

Till skillnad från exempel 10 fungerar denna interaktion bra även när du sitter ner. Återigen är återkopplingen viktig – utan den blir appen omöjlig att använda eftersom skärmläsaren inte vet vad som händer på skärmen när vi implementerar egen interaktion (om vi inte talar om det för den).

## Exempel 12. NFC-taggar – välj ett objekt

Med NFC-taggar blir det möjligt att knyta fysiska objekt till en app. I detta exempel har vi valt att skriva ut tre olika geometriska former på en 3D-skrivare och fästa NFC-taggar vid dem.

Med en tag som hänger utanför objektet, som i bilden nedan, blir det tydligt var taggen är och enkelt att föra den nära NFC-läsaren på telefonens baksida. Ett alternativ hade varit att gömma taggarna inuti objekten, men eftersom läsavståndet är kort behöver taggarna i så fall finnas nära under ytan, och det skulle även krävas en tydlig markering som visar var på objektet taggen finns.



*Exempel 12. NFC-taggar fästa vid olika geometriska objekt. Uppgiften är att föra NFC-läsaren på telefonens baksida mot ett specificerat objekts tag.*

Denna uppgift går ut på att hitta en namngiven form – en pyramid. När du hittat den för du tillhörande tag mot baksidan på telefonen. Om du hittat rätt objekt spelar telefonen upp en liten snutt ”orientalisk” musik.

## Exempel 13. NFC-taggar – namnge objekt

Exempel 13 bygger på samma objekt som i exempel 12, men i denna uppgift ska du istället namnge alla tre objekten. Så som exemplet är utformat får du inte någon direkt återkoppling på om svaret är rätt eller fel (du kan använda exempel 12 för att kolla svaren) – istället är detta en utformning som skulle kunna användas för exempelvis ett prov.



*Exempel 13. Namnge de tre olika objekten genom att i rätt ordning föra NFC-läsaren på telefonens baksida mot objektens taggar.*

## Sammanfattning

Att utveckla lärappar som är tillgängliga också för barn med synnedsättning behöver inte vara svårt. Med tydlig grafik, bra kontrast samt standardkomponenter och bildbeskrivningar kombinerat med olika ljudfiler kommer man långt.

Vill du vara mer avancerad går det bra att utnyttja telefonens sensorer till att peka, tippa och skaka. Även ”drag and drop” kan fungera under förutsättning att den är väl utformad. Så fort du går ifrån standardkomponenterna för att implementera din egen interaktion måste du dock se till att den som inte ser får tillräcklig information och återkoppling. Detta kan du göra på två sätt – antingen spelar du upp inspelat ljud eller så skickar du ett meddelande till skärmläsaren och låter den tala om vad som händer på syntetisk väg. Det finns för- och nackdelar med båda sätten. Ljudfiler kan ofta bli roligare och låter vanligtvis bättre. Samtidigt tar ljudfiler plats och det kan därför ibland vara smidigt att istället låta skärmläsaren läsa upp information. Ljudfiler är lätta att spela upp, men måste spelas in. Syntetiskt ljud kräver viss implementering och att meddelanden skickas till skärmläsaren.

Vi har i dessa exempel inte alls använt telefonens mikrofon för input – men självklart går det att använda taligenkänning för att ge svar på olika uppgifter. Det som man dock behöver se till i det fallet är att inte talsyntesen från skärmläsaren (eller andra ljud från appen) går in och stör.

Handlar det om geometri är det fortfarande svårt att göra något bra med enbart en telefon. För denna typ av appar rekommenderar vi därför att man kombinerar appen med någon form av taktilt material – gärna 3D-utskriften med tillhörande NFC-taggar.

En sammanfattning av vad vi lärt oss genom att arbeta med dessa exempel är följande lista med råd **för dig som utvecklar appar:**

- Lär dig använda skärmläsaren och testa regelbundet din app både med skärmläsare och med fingerad optik (åtminstone ”Cambridge simulation glasses”) under utvecklingen. Tänk på att även testa med skärmläsaren utan att se på skärmen (blunda eller ha förbundna ögon) – man kan bli lurad att tro att något fungerar bättre än det gör för att man ser vad som händer.
- Använd stor text och tydliga bilder med god kontrast (undvik små detaljer). Tänk på att alla inte ser färg likadant.
- Använd gärna standardkomponenter (gör du egna måste du kontrollera att de fungerar även med skärmläsare).
- Se till att ordningen på komponenterna blir rätt när man stegar igenom med skärmläsaren.
- Se till att alla bilder som betyder något har en bildbeskrivning som inte i sig självt löser uppgiften och direkt ger svaret.
- Använd gärna ljud och musik för att göra appen roligare och för att ge återkoppling.
- Fundera på om användning av gester (peka, skaka, tippa etc.) kan göra appen roligare.
- Var noga med att implementera återkoppling för allt som händer i dina komponenter.
- Fundera på möjligheten att använda 3D-objekt och NFC-taggar.
- Se till att experter (personer med synnedsättning) testat din app.

För **dig som utvärderar appar** gäller en liknande lista:

- Lär dig använda skärmläsaren och testa appen både med skärmläsare och med fingerad optik (åtminstone ”Cambridge simulation glasses”). Tänk på att även testa med skärmläsaren utan att se på skärmen (blunda eller ha förbundna ögon) – man kan bli lurad att tro att något fungerar bättre än det gör för att man ser vad som händer.
- Kontrollera att appen använder stor text och tydliga bilder med god kontrast (inga små detaljer). Kontrollera att det inte krävs färgseende för att lösa uppgifterna.
- Kontrollera att ordningen på komponenterna blir rätt när man stegar igenom med skärmläsaren.
- Kontrollera att alla bilder som betyder något har en bildbeskrivning som inte i sig självt löser uppgiften och direkt ger svaret.
- Kontrollera att appen ger en bra återkoppling till användaren i all interaktion.
- Låt experter (personer med synnedsättning) testa appen.

De absolut viktigaste råden i båda listorna är det första och det sista – testa med skärmläsare och fingerad optik och ta alltid hjälp av personer med synnedsättning för att testa apparna.

*Lycka till!*