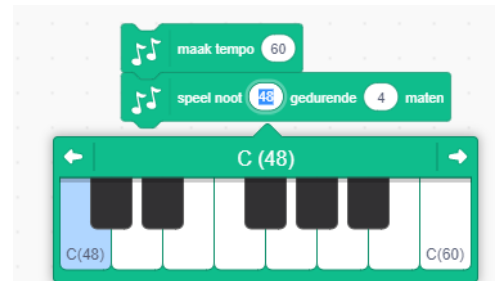


Een muzieknoot heeft een toonhoogte en een toonlengte.

Toonhoogte:

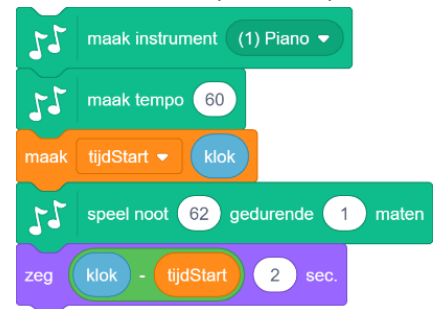
Wanneer op het veld noot (48) klikt verschijnt er een toetsenbord en kan je kiezen welke noot je wil laten horen. In de muziek zijn de namen van de noten de letters A t/m G. Dit zijn de witte toetsen op een piano. Na de letter G wordt weer gestart met de letter A. In Scratch zie je deze letters in combinatie met een nummer tussen haakjes. Dit nummer geeft eenduidig de toonhoogte. Hiernaast is noot C(48) geselecteerd. Noot C(60) is de middel C op een piano (4<sup>de</sup> C).



Toonlengte: In de muziek wordt een toonlengte uitgedrukt in het aantal tellen dat hij duurt. Hoe snel er wordt geteld noemen we het tempo. Het tempo is het aantal tellen dat in 1 minuut past. Men spreekt van BPM (Beats Per Minute). We moeten kiezen welke toonlengte bij de BPM past. We kiezen in deze oefening ervoor om met kwartnoten te tellen. Bij een BPM van 60 voor kwartnoten duurt een kwartnoot 1 tel (1 seconden). Dus een hele noot duurt dan 4 tellen. Een halve noot duurt dan 2 tellen en een achtste duurt 0.5 seconden. Dus hoe hoger de BPM, hoe korter een (kwart/halve/hele)noot duurt.

In het voorbeeld hierboven wordt de toonhoogte op C(48) gezet en het tempo op 60. De toon (noot) 4 duurt dan seconden. ("maten" is hier het Engelse "Beats" vertaald). Het nummer 48 is het unieke nummer op de computer voor deze C-toon.

Hiernaast staat een testje. De nootlengte wordt gemeten door de begin- en eindtijd te meten en het verschil te melden. Neem dit even over bij een willekeurige sprite. De variabele "startTijd" moet eerst even worden aangemaakt.

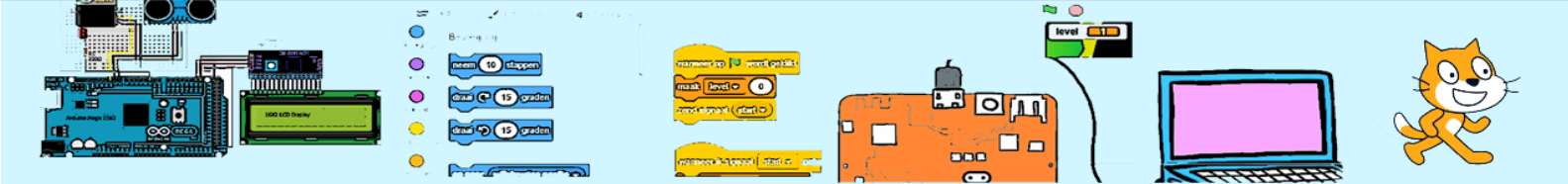


Nu kan wat je experimenteren met de muziek-blokken. Zet het tempo op verschillende waardes en kijk wat het effect is op de nootlengte. Kies eens een ander noot/instrument om te horen hoe dit de klank verandert.

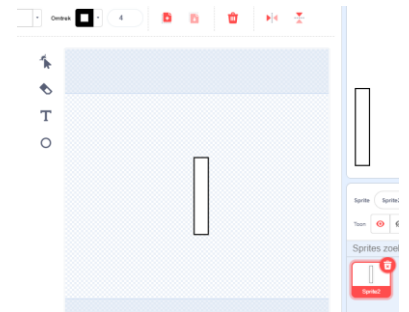
In deze oefening hieronder wordt in het begin de BPM op 100 gezet.

### 3. De witte toetsen

We gaan eerst het beginbestand (<https://coderclub.nl/dlx/scratchdigitalepianobegin.sb3>) downloaden en gebruiken. Ons testje gaat daarmee verloren. De muziekextensie wordt automatisch geladen. In het beginbestand staan bijna alle uiterlijkheden voor de sprites en alle globale variabelen (variabelen zichtbaar voor alle sprites). De lokale variabelen (alleen zichtbaar voor een één sprite) moeten zelf worden aangemaakt.



De witte toets is een witte rechthoek. Maak een sprite aan met de naam witteToets en teken een rechthoek. Kijk voor een idee voor de afmetingen naar de afbeelding hier rechts. De uiteindelijke afmetingen worden later bepaald.



```

    wanneer op deze sprite wordt geklikt
      maak tempo 60
      maak instrument (1) Piano
      speel noot 60 gedurende 0.25 maten
  
```

Voeg de code hier links toe aan de witteToets.

Een test: Klik op de sprite in het speelveld. Deze eerste toets zou al moeten werken.

Dupliceer deze sprite, plaats de 2<sup>de</sup> sprite direct naast de 1<sup>ste</sup> en verander de noot in 62. Zorg dat de 2 toetsen precies op dezelfde hoogte zijn geplaatst. Er zijn nu al 2 toetsen bespeelbaar. Zo zouden we alle 24 witte toetsen en 15 zwarte toetsen kunnen aanmaken. Maar wat als we iets willen wijzigen dan moet we dat 38 keer herhalen. Daarom gaan we het anders doen. Verwijder de 2<sup>de</sup> witteToets-sprite weer en gebruik onderstaande code voor de overgebleven witteToets-sprite. Hierin worden klonen gemaakt van de witteToets. Een kloon is een kopie van een sprite. De code en uiterlijkheden van klonen zijn hetzelfde. De lokale variabelen ("alleen voor deze sprite"), positie, richting, grootte et cetera kunnen van elkaar verschillen. We gaan gebruikmaken van de variabele toetsNummerWit om iedere kloon uniek te maken. Bij het maken van een kloon gebruiken we deze variabele om de positie van de kloon te bepalen. Na het maken van een kloon verhogen we deze variabele met 1 zodat de volgende kloon een hoger toetsNummerWit gebruikt. Hieronder de code.

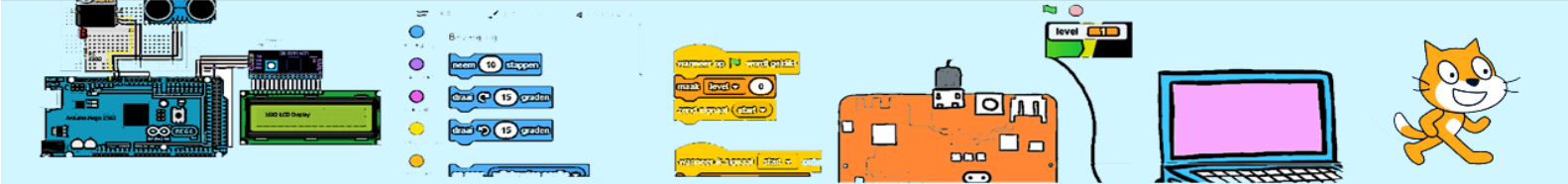
Wanneer we hierna op de groene vlag klikken verschijnen de witte toetsen.

```

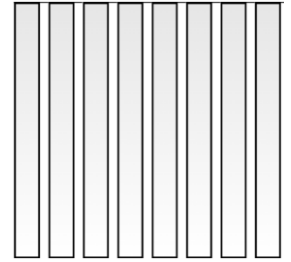
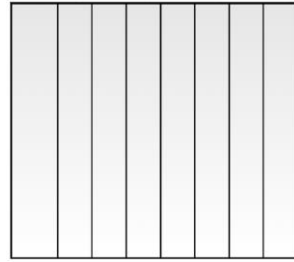
    wanneer op vlag wordt geklikt
      maak y -65
      maak x -220
      verdwijnt
      maak toetsNummerWit 1
      herhaal lengte van witteToetsen
        maak een kloon van mijzelf
        wacht 0.01 sec.
        verander toetsNummerWit met 1

    wanneer ik als kloon start
      maak toonHoogte item toetsNummerWit van witteToetsen
      verander x met 20 * toetsNummerWit + 0
      verander uiterlijk naar inrust
      ga naar laag achtergrond
      zet sleepbaar op niet sleepbaar
      verschijnt

    wanneer op deze sprite wordt geklikt
      maak tempo 60
      maak instrument (1) Piano
      speel noot 60 gedurende 1 maten
  
```

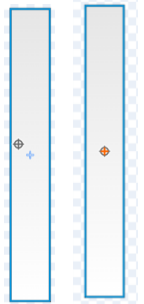


Maar één toets lijkt misschien breder dan de andere of de toetsen zijn te smal. Wanneer de rechter of linker toets breder lijkt komt dit doordat de toets te breed is en dat toetsen ervoor elkaar overlappen.

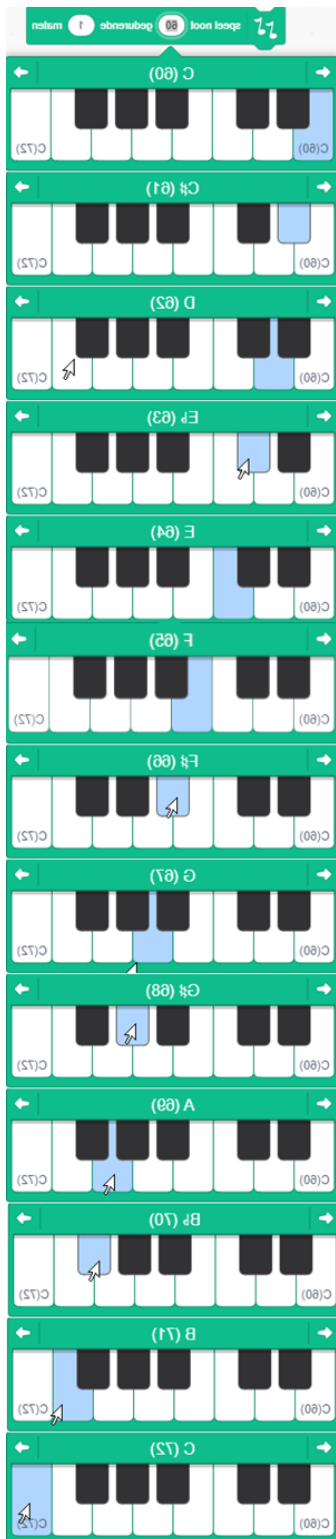
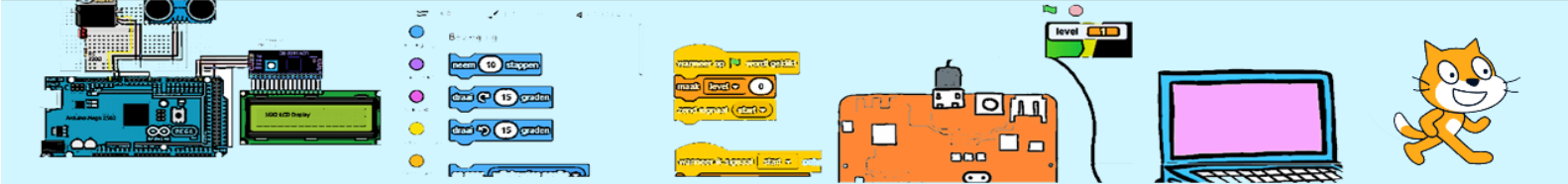


Dit kunnen we aanpassen in het uiterlijk van de witteToets. Aanpassingen zien we direct in het speelveld. Kies nu een breedte waardoor de toetsen precies naast elkaar komen te liggen.

Zorg tegelijkertijd dat het zwaartepunt van de afbeelding precies in het midden valt (kruisje op kruisje zoals op het rechter plaatje).



Na het maken van een kloon wordt er 0.01 seconden gewacht. Waarom is dit? In Scratch kunnen meerdere script gelijktijdig actief zijn. Maar die zijn niet altijd echt gelijktijdig actief. De computer wisselt heel snel tussen de scripts waardoor het lijkt of ze alle actief zijn. Wanneer we heel snel achter elkaar klonen gaan maken kan het voorkomen dat het script “wanneer ik als kloon start” nog niet klaar is wanneer de volgende kloon wordt gestart. Beide zouden dan hetzelfde toetsNummerWit gebruiken. Door heel even te wachten met het maken van de volgende kloon is de kans nihil dat dit gebeurt.



We hebben nu een toetsenbord met 20 toetsen/sprites (en de “master” is aanwezig maar verborgen).

Echter gebruikt iedere sprite nog dezelfde toonhoogte. Als we kijken naar de toonhoogte die we kunnen ingeven, zien we dat het verschil tussen de toonhoogtes van 2 naast elkaar liggende witte toetsen niet altijd hetzelfde is. Wanneer er een zwarte toets is, is het verschil 2. Wanneer er geen zwarte toets tussen beide is, is dit verschil 1. Dit maakt dat het berekenen van de toonHoogte van een toets niet eenvoudig. In dit geval is het eenvoudiger om gebruik te maken van een tabel om de toonHoogte voor een toets op te zoeken. Hierbij is het toetsNummerWit het regelnummer in de tabel. In Scratch noemen we dit een “lijst”. We starten de lijst bij toon 48 en eindigen bij toon 83 (zie rechts). We noemen hem toonHoogteWitteToetsen. We maken een variabele toonHoogte aan in de witteToets-Sprite die **alleen** zichtbaar is voor die sprite (lokaal). Met de opdracht

```
maak toonHoogte van item toetsNummerWit van toonHoogteWitteToetsen
```

kunnen we de toonHoogte achterhalen voor iedere toets .

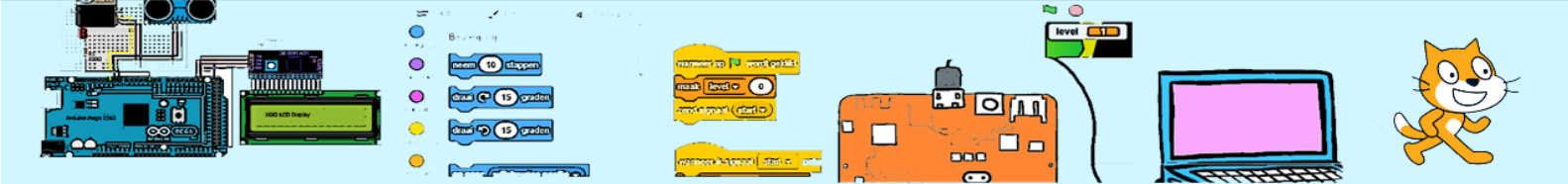
Deze lijst staat al in het begin-bestand. In plaats van “herhaal 20” gebruiken we voor het aantal klonen de lengte van de lijst. Mochten we de lijst later aanpassen dan verandert dit direct mee.

Hieronder de nieuwe code voor de witteToets. De bestaande code moet je dus even aanpassen.

Met de opdracht “zet sleepbaar op niet sleepbaar” voorkomen we dat tijdens het spelen de toetsen per ongelijk worden verplaatst. Dit werkt echter alleen in “full screen”-mode.

De opdracht “ga naar laag achtergrond” voorkomt dat wanneer we later de zwarte toetsen gaan maken de witte boven de zwarte worden getoond.

toetsNummerWit	toonHoogte
1	48
2	50
3	52
4	53
5	55
6	57
7	59
8	60
9	62
10	64
11	65
12	67
13	69
+ lengte 21 =	



```

    wanneer ik als kloon start
    verander x met 20 * toetsNummerWit + 0
    verander toetsNummerWit met 1
    verander uiterlijk naar inrust
    ga naar laag achtergrond
    zet sleepbaar op niet sleepbaar
    verschijn
    maak toonHoogte item toetsNummerWit van toonHoogteWitteToetsen
  
```

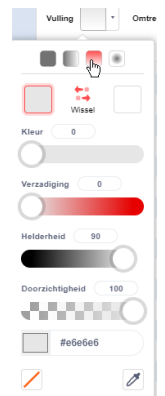
```

    wanneer op wordt geklikt
    maak y -65
    maak x -220
    verdwij
    maak toetsNummerWit 1
    herhaal lengte van toonHoogteWitteToetsen
    maak een kloon van mijzelf
    wacht 0.01 sec.
  
```

```

    wanneer op deze sprite wordt geklikt
    maak tempo 60
    maak instrument (1) Piano
    speel noot toonHoogte gedurende 1 maten
  
```

Dit werkt wel maar het ziet er vrij statisch uit. We gaan de toetsen laten “bewegen”. Ga naar het uiterlijk van de witteToets. Selecteer de rechthoek en klik op vulling-kleur-icoon. Kies voor tweekleuren onder en boven verschillend. Stel de bovenste kleur (linker) in op 0, 0, 90, 100 en de onderste (rechts) op 0, 0, 100, 100. Noem dit uiterlijk “inrust”.



Dupliceer het uiterlijk. Noem het boven uiterlijk “inrust” en noem het nieuwe uiterlijk “ingedrukt”. Selecteer de rechthoek en wissel de onder- en bovenkleuren.

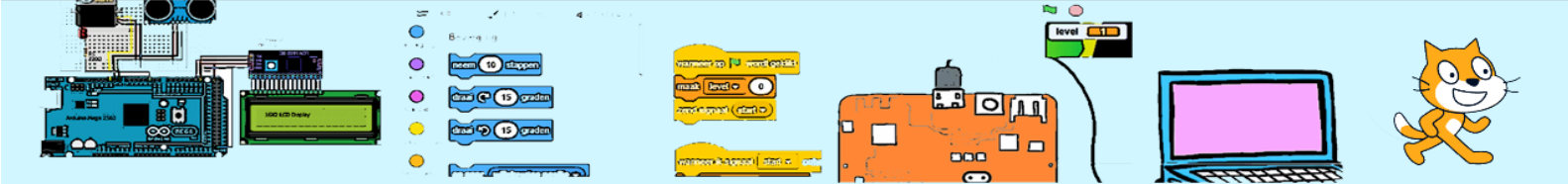
```

    wanneer op deze sprite wordt geklikt
    verander uiterlijk naar ingedrukt
    wacht tot niet raak ik muisaanwijzer
    verander uiterlijk naar inrust
  
```

Voeg de code hier links toe aan de witteToets.

Voeg `verander uiterlijk naar inrust` toe aan de code onder “wanneer ik als kloon start” zodat de toetsen starten in rust.

Probeer het uit. Klik op de groen vlag en klik op enkele toetsen.



## 4. De zwarte toetsen

De zwarte toetsen gaan we bijna hetzelfde doen als de witte toetsen. Kopieer de sprite witteToets en noem hem zwarteToets. Daarna moeten we enkele zaken gaan aanpassen. Het uiterlijk moet natuurlijk zwart zijn, minder breed en minder hoog. Laat het zwaartepunt wel weer in het midden. Nu de code. Er zijn minder zwarte toetsen dan witte. Om aan te geven dat er geen zwarte toets is, laten we in de lijst met toonHoogteZwarteToetsen op de plekken waar geen zwarte toets is, de regel leeg. We laten in de code “wanneer ik start als kloon” de kloon alleen zien wanneer de toonHoogte groter is dan 1. We verplaatsen de zwarteToetsen tenopzichte van 10 (x-eenheden) naar rechts (precies tussen 2 witte toetsen).

toonHoogteZwarteToetsen	
1	49
2	51
3	
4	54
5	56
6	58
7	
8	61

```

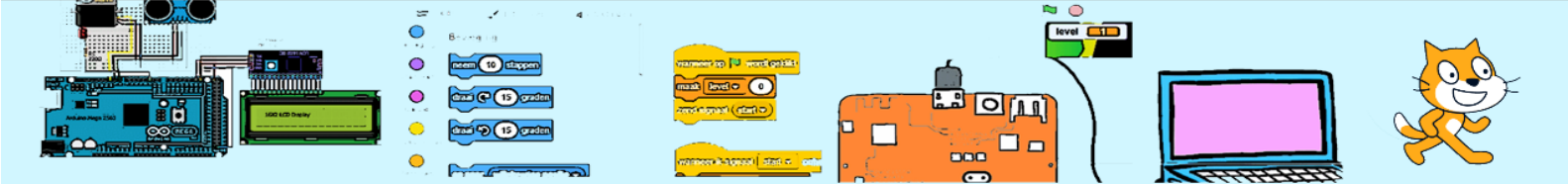
wanneer op vlag wordt geklikt
  maak y -65
  maak x -220
  verdwijf
  maak toetsNummerZwart 1
  herhaal lengte van zwarteToetsen
    maak een kloon van mijzelf
    wacht 0.01 sec.
    verander toetsNummerZwart met 1

wanneer ik als kloon start
  als item toetsNummerZwart van zwarteToetsen > 1 dan
    maak toonHoogte item toetsNummerZwart van zwarteToetsen
    verander x met 20 * toetsNummerWit + 10
    zet sleepbaar op niet sleepbaar
    verander uiterlijk naar inrust
    ga naar laag voorgrond
    verschijn

wanneer op deze sprite wordt geklikt
  maak zwartIngedrukt 1
  verander uiterlijk naar ingedrukt
  wacht tot niet raak ik muisaanwijzer
  verander uiterlijk naar inrust
  maak zwartIngedrukt 0

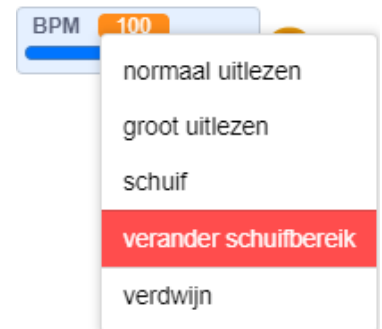
wanneer op deze sprite wordt geklikt
  maak tempo 60
  maak instrument (1) Piano
  speel noot 66 gedurende 1 maten
  
```

Tijd om weer eens te testen.



## 5. BPM

Tot nu toe duren de noten even lang. We gaan de gebruiker de mogelijkheid geven de BPM in te stellen tussen 10 en 200. Maak de variabele BPM zichtbaar op het speelveld en verander dit in een schuif als dat nog niet gedaan is. Stel het bereik in op tussen 10 en 200.



Nu moeten we BPM aan gebruiken bij het bespelen van de toetsen. Code moet én bij de witteToets én aan de zwarteToets worden aangepast.



## 6. Instrument

De computer kan diverse instrumenten na bootsen. Maak de variabele instrument zichtbaar op het speelveld en verander dit in een schuif. Stel het bereik in op tussen 1 en 100. We moeten de code voor de witteToets en de zwarteToets weer aanpassen. Zie rechts.



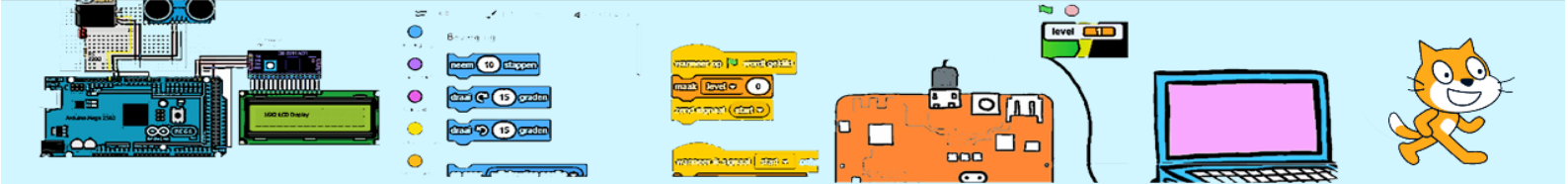
Stel je voor dat we niet met klonen hadden gewerkt. Dan hadden dergelijke wijzigingen 38 keer herhaald moeten worden in plaats van de 2 maal nu.

## 7. Nauwkeurige BPM en instrument selectie.

Het kiezen met een schuif gaat niet makkelijk. We gaan daarom knoppen maken om deze variabele met 1 te vermeerderen of te verminderen. Zo iets als hier rechts.







De sprites staan al in begin-bestand maar nog zonder code. Die gaan we nu toevoegen. Kies de juiste code bij de +/- sprites. Waarom staan er als-condities in de scripts?

wanneer op deze sprite wordt geklikt

als  $10 < \text{BPM}$  dan

verander BPM met -1

wanneer op deze sprite wordt geklikt

als  $1 < \text{instrument}$  dan

verander instrument met -1

wanneer op deze sprite wordt geklikt

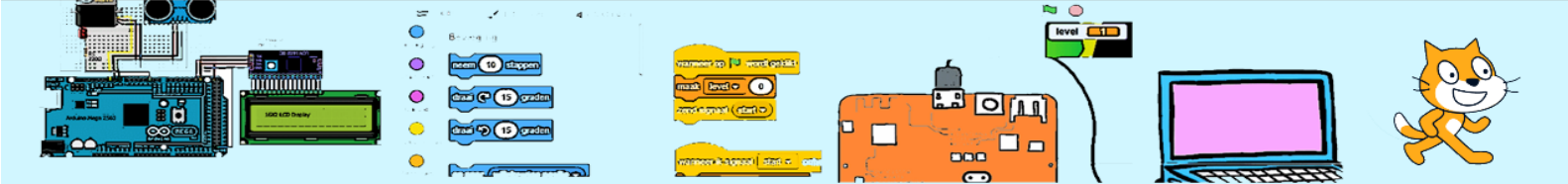
als  $\text{instrument} < 100$  dan

verander instrument met 1

wanneer op deze sprite wordt geklikt

als  $\text{BPM} < 200$  dan

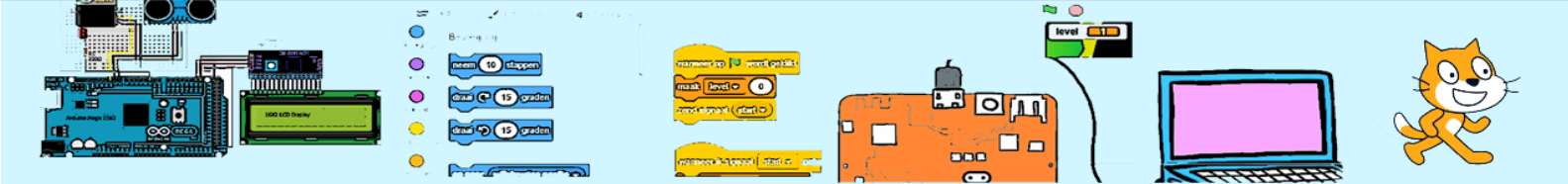
verander BPM met 1



In het beginbestand staan 4 sprites van instrumenten (piano, gitaar, saxofoon en vibrafoon). Het is de bedoeling dat wanneer de gebruiker hierop klikt dat het overeenkomstige muziekinstrument wordt geselecteerd. Het geselecteerde instrument wordt groter gemaakt (80%). Door daarna een signaal “nieuwInstrument” te versturen en aan dit signaal een script te hangen, kunnen we zorgen dat de **niet** geselecteerde instrumenten hun normale grootte (60%) (terug)krijgen.

We hebben geen mogelijkheid om het signaal “nieuwInstrument” te koppelen aan de schijf-variabele waarmee je ook je ook instrumenten kan selecteren. Daarom laten we dit signaal ook regelmatig genereren door de metronoom die hieronder aanbod komt.

Hierboven de code voor de **gitaar**. Voeg de code toe aan gitaar en ook aan de andere instrumenten. Bedenkt dat verschillende instrumenten een ander instrument-nummer hebben. De code moet voor de andere instrumenten dus iets worden aangepast.



## 8. Metronoom

Wanneer je muziek maakt kan een metronoom helpen bij het tellen. Op de beat geeft de metronoom een zacht tikje.

```
when green flag clicked
  set volume to 3%
  make metronoomAan 0
  make BPM 100
  make instrument 1
  set sleepable to not sleepable
  appear
  loop
    send signal beat
    send signal nieuwInstrument
    wait 60 / BPM sec.

when clicked on this sprite
  if metronoomAan = 0 then
    make metronoomAan 1
  else
    make metronoomAan 0

when I receive signal beat
  if metronoomAan = 1 then
    play drum (1) Snarentrom gedurende 0.01 maten
```

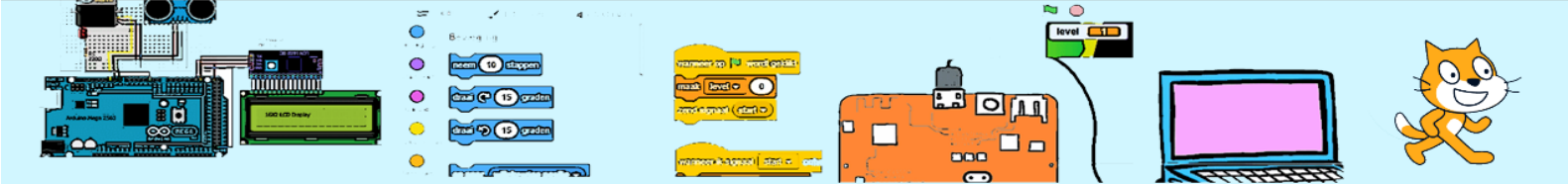
Ook levert onze metronoom een signaal "nieuwInstrument". Dit signaal zorgt ervoor dat wanneer een instrument wordt gekozen met de schuif het geselecteerde instrument groter wordt en de nadere nu normale grootte krijgen (zie voorgaande paragraaf).

## 9. Akkoorden

Een akkoord is een samenklank van meerdere toonhoogtes die in combinatie worden gespeeld. Het geluid klinkt daardoor voller. Er zijn tientallen verschillende akkoorden waarvan sommige vaak worden gebruikt en andere weinig. In deze oefening gaan we de veel gebruikte majeur- en mineur-akkoord nabootsen waarbij we zo eenvoudig mogelijk blijven. Zo kiezen we er o.a. voor de tonen binnen het akkoord gelijktijdig te laten horen. In het majeur-akkoord is er een grondtoon en 2 hogere tonen. De 2<sup>de</sup> toon ligt 4 toetsen ( $4 \cdot \frac{1}{2}$  toonafstanden) hoger dan de grondtoon. De 3<sup>de</sup> toon ligt 7 toetsen ( $7 \cdot \frac{1}{2}$  toonafstanden) hoger dan de grondtoon. Het majeur-akkoord wordt door mensen als vrolijk ervaren terwijl het mineur-akkoord juist als droevig wordt ervaren. In het mineur-akkoord is de 2<sup>de</sup> toon 3 toetsen hoger dan de grondtoon en de 3<sup>de</sup> toon is ook 7 toetsen hoger dan de grondtoon.

Hoe kunnen we meerdere tonen gelijktijdig laten horen en bepalen we of een akkoord moet worden gespeeld?

We gaan de variabele majeur en mineur bijhouden. En in de volgen paragraaf hebben we ook een variabele "glissando nodig". Daarvoor voegen we onderstaande code toe aan het speelveld.



```
wanneer a is ingedrukt
  maak majeur 1
  wacht tot niet toets a ingedrukt?
  maak majeur 0

wanneer i is ingedrukt
  maak mineur 1
  wacht tot niet toets i ingedrukt?
  maak mineur 0

wanneer g is ingedrukt
  maak glissando 1
  wacht tot niet toets g ingedrukt?
  maak glissando 0
```

Dus wanneer de variabele majeur gelijk is aan 1 weten we dat de gebruiker een majeur-akkoord wil horen wanneer de variabele mineur gelijk is aan 1 dan wil de gebruiker een mineur akkoord. We laten alle tonen van het akkoord gelijktijdig horen door ze parallel aan dezelfde gebeurtenis (wanneer op deze sprite wordt gedrukt) te hangen.

Deze code moet van de witteToets en de zwarteToets worden uitgebreid.

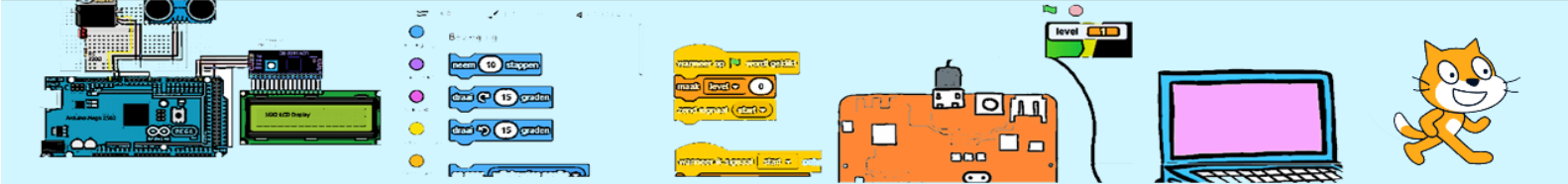
```
wanneer op deze sprite wordt geklikt
  maak tempo BPM
  maak instrument instrument
  speel noot toonHoogte gedurende 1 maten

wanneer op deze sprite wordt geklikt
  als majeur = 1 dan
    maak instrument instrument
    speel noot toonHoogte + 4 gedurende 1 maten

wanneer op deze sprite wordt geklikt
  als mineur = 1 dan
    maak instrument instrument
    speel noot toonHoogte + 3 gedurende 1 maten

wanneer op deze sprite wordt geklikt
  als mineur = 1 of majeur = 1 dan
    maak instrument instrument
    speel noot toonHoogte + 7 gedurende 1 maten
```

Wat doet onze piano wanneer de gebruiker gelijktijdig én de a-toet én de i-toet indrukt tijdens het bespelen?



## 10. Glissando

Wanneer je achtereenvolgens meerdere toetsen van laag naar hoog of andersom aanslaat noemen we dit een glissando. Een glissando is op onze digitale piano nog niet te doen daar je heel snel op iedere toets zou moeten klikken.

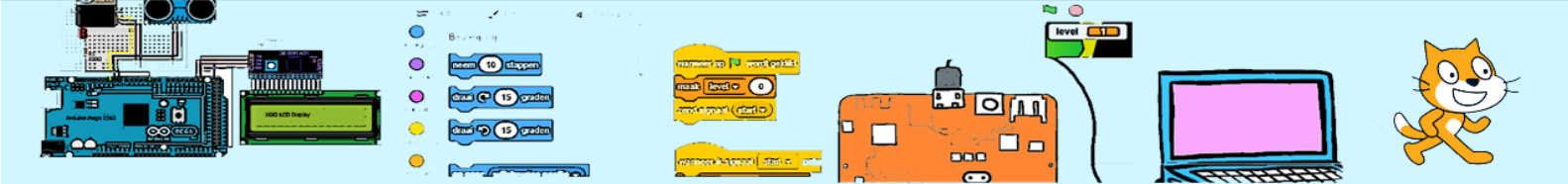
Maar we gaan code toevoegen die het mogelijk maakt met een ingedrukte muis over de toetsen te glijden.

Eerst doen we dit voor de **zwarteToetsen** en daarna voor de **witteToetsen** daar dit iets ingewikkelder is.

We voegen code toe aan het script onder “wanneer ik als kloon start”. In een herhaalloop gaan we continu testen of de gebruiker glissando wil, en de ingedrukte muis een toets raakt. Als dit waar is dan spelen we de noot. Dit kan parallel lopen met de acties onder “wanneer op deze sprite geklikt wordt”. We zetten ook al vast de variabele **zwartIngedrukt**. Deze hebben we later nodig bij de **witteToets**.

Het is niet logisch glissando te combineren met een akkoord. Deze digitale piano ondersteunt zoiets niet.

```
wanneer ik als kloon start
als <item toetsNummerZwart van zwarteToetsen > > 1 dan
  maak toonHoogte <item toetsNummerZwart van zwarteToetsen >
  verander x met 20 * toetsNummerWit + 10
  zet sleepbaar op niet sleepbaar
  verander uiterlijk naar inrust
  ga naar laag voorgrond
  verschijn
herhaal
  als glissando = 1 dan
    als raak ik muisaanwijzer ? dan
      als muis ingedrukt? dan
        verander uiterlijk naar ingedrukt
        maak instrument instrument
        speel noot toonHoogte gedurende toonLengte maten
        wacht tot niet raak ik muisaanwijzer ?
        verander uiterlijk naar inrust
      wacht 0.01 sec.
```



Wanneer in Scratch 2 sprites over elkaar liggen en de muis boven beide is gelokaliseerd dan zullen beide sprites positief testen op de conditie “raak ik muiswijzer?”

Dit betekent dat wanneer de muis over een zwarteToets beweegt ook de onderliggende witteToets actief wordt. Dat willen we niet.

Een oplossing is om bij de zwarteToets bij te houden of deze ingedrukt is en bij de witteToets hierop te controleren. Vlak voor deze controle wachten we 0.01 seconden om zeker te zijn dat de zwarteToets tijd heeft om de variabele te veranderen.

Hiernaast de aanpassing van de code voor de witteToets.

Dit werkt aardig maar er zijn enkele kleine probleempje/bependingen. Weet jij welke dit zijn?

```

herhaal
als glissando = 1 dan
als raak ik muisaanwijzer ? dan
wacht 0.01 sec.
als zwartIngedrukt = 0 dan
als muis ingedrukt? dan
verander uiterlijk naar ingedrukt
maak instrument instrument
speel noot toonHoogte gedurende toonLengte maten
wacht tot niet raak ik muisaanwijzer ?
verander uiterlijk naar inrust
wacht 0.01 sec.
  
```

## 11. Toonlengte

Tot nu toe hebben we gewerkt met kwartnoten die een tel duren. We gaan nu werken met hele, halve, kwart en achtste noten. Wanneer de gebruiker geen aanpassing vraagt blijven we halve noten gebruiken. Wanneer de toets “1” wordt ingedrukt wil de gebruiker een hele noot horen. Bij toets “2” wordt dit een halve noot. En bij een “8” wordt dit een achtste noot. Voeg onderstaande code toe aan het speelveld.

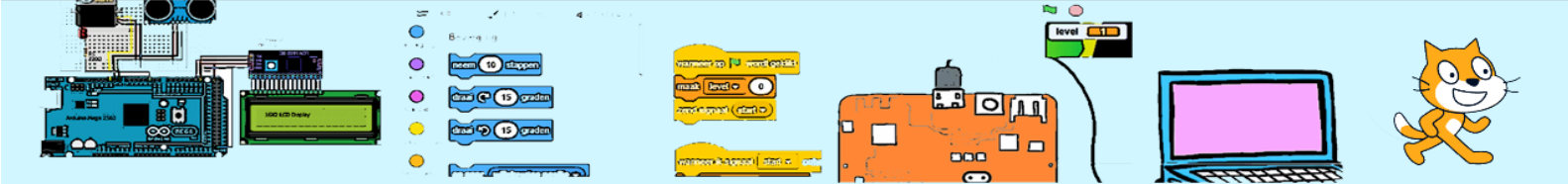
```

wanneer 1 is ingedrukt
maak toonLengte 4
wacht tot niet toets 1 ingedrukt?
maak toonLengte 1

wanneer 2 is ingedrukt
maak toonLengte 2
wacht tot niet toets 2 ingedrukt?
maak toonLengte 1

wanneer 8 is ingedrukt
maak toonLengte 0.5
wacht tot niet toets 8 ingedrukt?
maak toonLengte 1
  
```

Nu moeten we nog de code voor toetsen aanpassen. Dus beide toetsen moeten worden aangepast. Per toets moet 5 maal een “1” worden vervangen door nootLengte. Wanneer je het niet weet, vraag dan aan een mentor.



Wat gebeurt er wanneer je 2 van deze toetsen gelijktijdig indrukt?

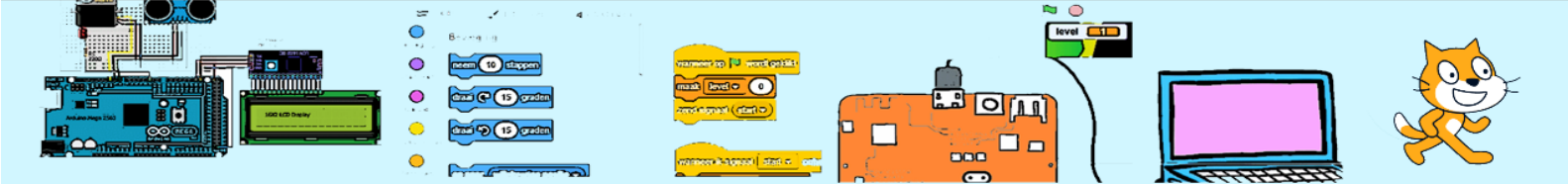
## 12. Opnemen en afspelen

De code hiernaast moet aan de spelen-sprite worden toegevoegd. Na het spelen van een noot wachten we even lang als de nootLengte is. Als de hele melodie is afgespeeld zetten we de toonLengte weer op 1 daar deze gewijzigd kan zijn door de laatste noot uit de melodie.

```
when clicked on this sprite  
  set volume to 1 %  
  play note 130 for 0.0001 measures  
  set volume to 100 %  
  change appearance to on  
  make index 1  
  repeat (length of melodyToneHeight)  
    make toneLength item index of melodyToneLength  
    make gewensteToon item index of melodyToneHeight  
    if gewensteToon > 0 then  
      send signal playTone  
      wait (60 / tempo * toneLength + 0.05) sec.  
      change index by 1  
      wait until pause = 0  
  make toneLength 1  
  change appearance to off  
when green flag clicked  
  change appearance to off  
  make pause 0
```

De code voor de beide toetsen moet ook uitgebreid met de code hieronder. Het “wanneer op de sprite wordt geklikt”-script slaat de aangeslagen toets op en de bijbehorende nootLengte in de daarvoor bedoelde lijsten als “opnemen”=1.

Het “wanneer ik signaal ‘speelToon’ ontvang”-script is na de “als gewensteToon – toonHoogte” is alleen actief in de sprite waar de gewensteNoot overeenkomt met de toonHoogte van de sprite zelf.



```

    wanneer op deze sprite wordt geklikt
    als <opnemen = 1> dan
        voeg toonHoogte toe aan melodieToonHoogte
        voeg toonLengte toe aan melodieToonLengte
        maak instrument instrument
        speel noot toonHoogte gedurende toonLengte maten

    wanneer ik signaal speelToon ontvang
    als <gewensteToon = toonHoogte> dan
        verander uiterlijk naar ingedrukt
        maak tempo BPM
        maak instrument instrument
        speel noot toonHoogte gedurende toonLengte maten
        verander uiterlijk naar inrust
  
```

Er staat al iets in de melodie-lijsten, dus je kan al op de speeltoets drukken.

We hebben nog wat nodig om de gebruiker het opnemen aan en uit te kunnen laten zetten. De code hiernaast moet aan het de opname-sprite worden toegevoegd. Bij het starten van een opname maken we eerst de melodielijsten leeg.

De opname registreert geen stiltes. Hoe zou dit wel kunnen?

NB: De opname registreert de toonLengte maar niet hoe snel gespeeld wordt.

Maak de melodie-lijsten zichtbaar en kijk wat er gebeurt wanneer je iets opneemt.

```

    wanneer op vlag wordt geklikt
    verander uiterlijk naar uit
    maak opnemen 0

    wanneer op deze sprite wordt geklikt
    als <opnemen = 0> dan
        verwijder alle van melodieToonHoogte
        verwijder alle van melodieToonLengte
        verander uiterlijk naar aan
        maak opnemen 1
    anders
        verander uiterlijk naar uit
        maak opnemen 0
  
```

Een DVD/MP3/Media-speler heeft meestal ook een pauze-toets. Verzin zelf een invulling.