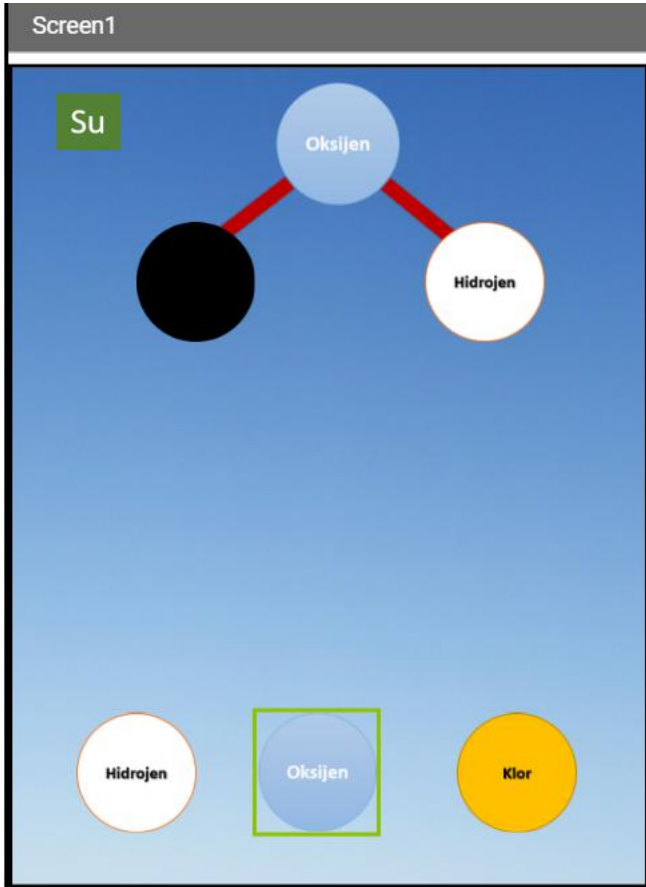


Molekölü tamamlama uygulaması

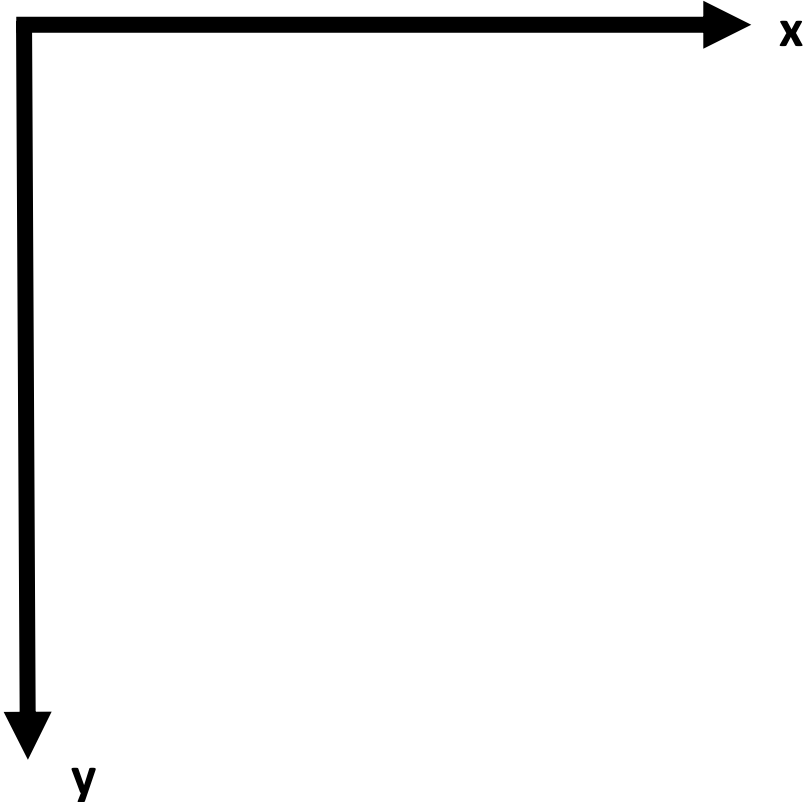
- Uygulamada öğrencinin eksik olan atomu su molekülünü tamamlaması için sürükleyip doğru yere bırakmasını isteyeceğiz.
- Bu örneğimizde de Canvas (Tuval) ve ImageSprite (Hareketli Resim) bileşenlerini kullanacağız.
- Önceki örnekten farklı olarak bu örnekte ImageSprite'ı sürükleyip taşıma, ve bu bileşenin koordinatlarını bloklar yardımıyla değiştirmeyi kullanacağız.

İçerik Bileşenler ve Medya Dosyaları



App Inventor'da Koordinat kavramı

- App Inventor'da kullanılan koordinat düzlemi matematikteki koordinat düzlemine oldukça benzemektedir. Tek fark orijin noktasının sol üst köşede olmasıdır.



Uygulama Blokları

- Uygulamanın blokları için ilk olarak atomlardan birini hareket ettirmek için aşağıdaki bloğu hazırlayacağız. Bu blok hareketli resmin koordinatlarını parmağımızın dokunduğu yer ile eşitlenmesi işlemi yapmaktadır.



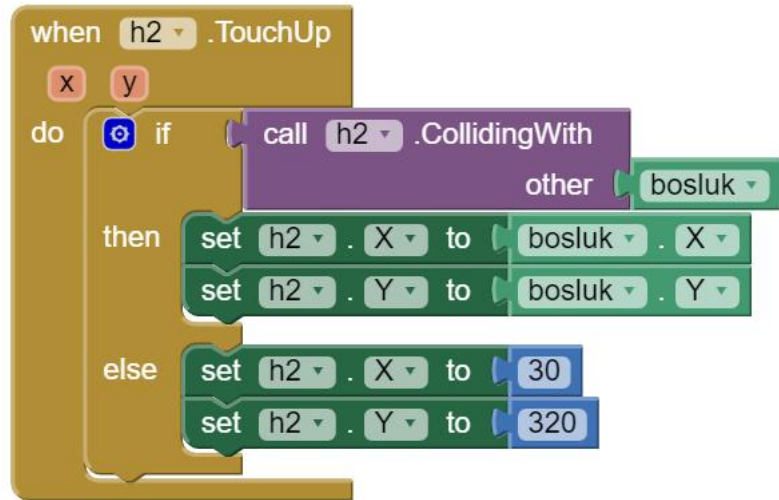
```
when h2 .Dragged
  startX startY prevX prevY currentX currentY
do
  set h2 . X to get currentX
  set h2 . Y to get currentY
```

The image shows a Scratch code block with a yellow background. At the top, it says "when h2 .Dragged". Below this, there are six variable input fields: "startX", "startY", "prevX", "prevY", "currentX", and "currentY". Underneath these fields is a "do" block containing two "set" blocks. The first "set" block is "set h2 . X to" followed by a "get currentX" block. The second "set" block is "set h2 . Y to" followed by a "get currentY" block.

- Bu uygulama için kullanacağımız en önemli kavram programlamanın en önemli noktalarından olan if...then...else kavramıdır.
- Bu kavram sayesinde programlar mantıksal sınama yaparak bir koşul doğru ise belli işlevleri yerine getirip, doğru değil ise farklı işlevleri yerine getirmektedir.

Uygulama Blokları

- Bu uygulamada if...then..else kavramını doğru atom yerine sürüklendiğinde yerine oturmasını yanlış ise yerine geri dönmesini sağlamak için kullanacağız.
- Bunun için bloğumuz aşağıdaki gibi olacak.



```
when h2 . TouchUp
do
  if
    call h2 . CollidingWith other bosluk
  then
    set h2 . X to bosluk . X
    set h2 . Y to bosluk . Y
  else
    set h2 . X to 30
    set h2 . Y to 320
```

- Yandaki blok parmağımızı hidrojen resminde kaldırdığımızda resmin boşluk isimli yuvaya dokunup dokunmadığını kontrol ediyor ve eğer dokunuyor ise ikisinin koordinatlarını birbirine eşitliyor.
- Eğer dokunmuyor ise hidrojen resmini ilk konumuna geri döndürüyor.

Uygulama Blokları

- Diğer iki atomun da sürüklenebilir olması için yandaki bloklara ihtiyacımız olacak.
- Diğer iki atom resmi yanlış cevap olacağı için sınamaya gerek olmadan parmağımızı çeker çekmez yerlerine dönmelerini yandaki bloklarla sağlayabiliriz.

```
when o2 .Dragged
  startX startY prevX prevY currentX currentY
do
  set o2 . X to get currentX
  set o2 . Y to get currentY
```

```
when cl .Dragged
  startX startY prevX prevY currentX currentY
do
  set startX to get startX
  set startY to get startY
  set prevX to get currentX
  set prevY to get currentY
```

```
when o2 .TouchUp
  x y
do
  set o2 . X to 120
  set o2 . Y to 320
```

```
when cl .TouchUp
  x y
do
  set cl . X to 220
  set cl . Y to 320
```

Uygulama blokları tamamlanmış hali

```
when h2 . TouchUp
  x y
  do
    if call h2 . CollidingWith other bosluk
    then
      set h2 . X to bosluk . X
      set h2 . Y to bosluk . Y
    else
      set h2 . X to 30
      set h2 . Y to 320
```

```
when o2 . TouchUp
  x y
  do
    set o2 . X to 120
    set o2 . Y to 320
```

```
when cl . TouchUp
  x y
  do
    set cl . X to 220
    set cl . Y to 320
```

```
when h2 . Dragged
  startX startY prevX prevY currentX currentY
  do
    set h2 . X to get currentX
    set h2 . Y to get currentY
```

```
when o2 . Dragged
  startX startY prevX prevY currentX currentY
  do
    set o2 . X to get currentX
    set o2 . Y to get currentY
```

```
when cl . Dragged
  startX startY prevX prevY currentX currentY
  do
    set cl . X to get currentX
    set cl . Y to get currentY
```