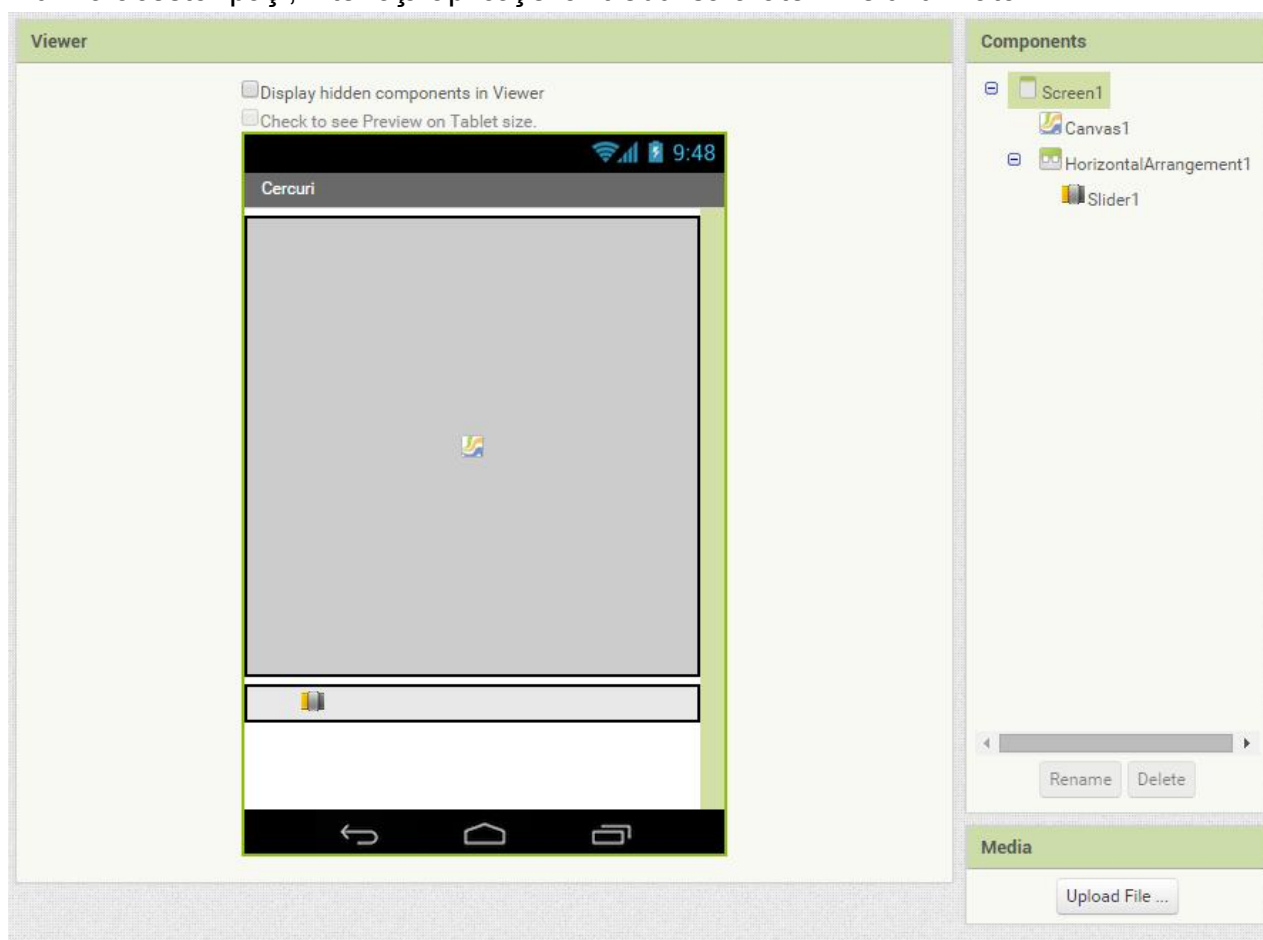


1. Deschideți App Inventor și creați o nouă aplicație cu numele "Cercuri".
2. Stabiliți următoarele proprietăți ale lui Screen1:
 - ScreenOrientation: **Portrait**
 - Title: **Cercuri**
3. Trageți Screen1 o componentă de tip **Canvas** și stabiliți-i următoarele proprietăți:
 - BackgroundColor: **Light Gray**
 - Height: **320pixels**
 - Width: **FillParent**
4. Trageți sub Screen1 o componentă de tip **HorizontalArrangement** căreia îi stabiliți următoarele proprietăți:
 - Width: **FillParent**
 - AlignHorizontal: **Center**
5. Plasați **ÎN** componenta de tip **HorizontalArrangement** o componentă de tip Slider, căreia îi stabiliți următoarele proprietăți:
 - Width: **75 percent**
 - MaxValue: **50**
 - MinValue: **3**
 - ThumbPosition: **6**

În urma acestor pași, interfața aplicației ar trebui să arate în felul următor:



Treceți în modulul de programare (designer) unde creați următoarele blocuri:

6. Definiți o variabilă globală **nc**, pe care o inițializați cu **3**. Această variabilă va reprezenta numărul de cercuri vizibile în desen:



7. Vom defini o funcție cu numele "**culoare**" care primește prin intermediul parametrului său "**i**" o valoare cuprinsă între **1** și **nc**. Funcția va întoarce (deci va fi una de tip "result") o nuanță de roșu de intensitate maximă (adică de valoare 255), atunci când **i** are valoarea **1**, nuanță care va descrește în intensitate cu cât **i** este mai mare, devenind negru atunci când **i** are valoarea **nc**. Creați pentru aceasta următoarele blocuri:

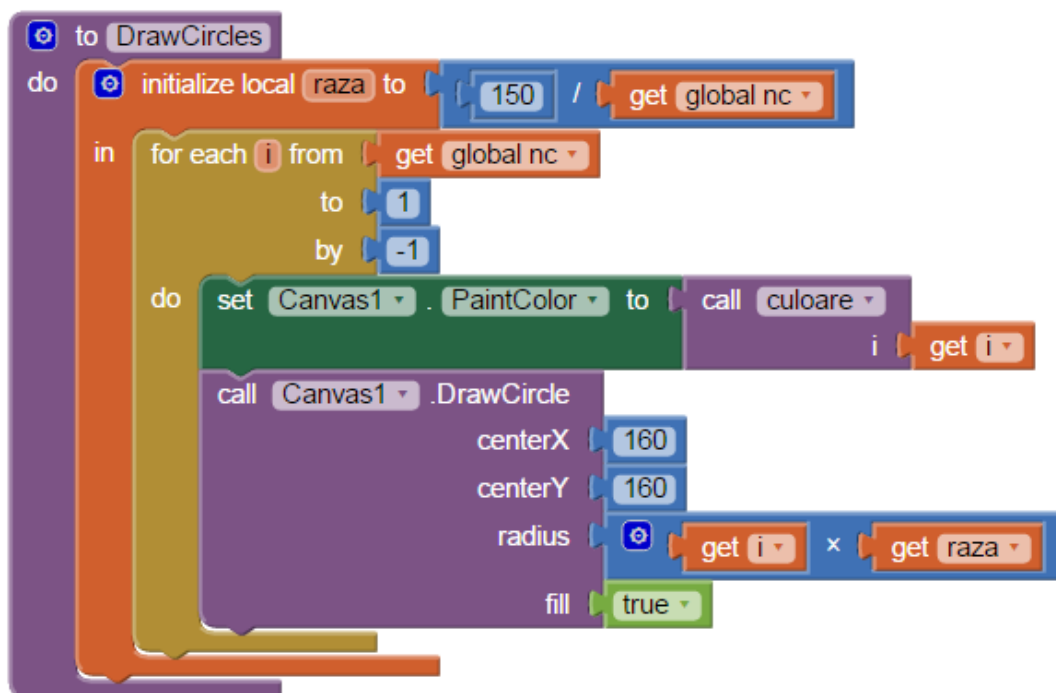


După cum se observă, această funcție definește nuanța dorită a roșului, folosind funcția predefinită **make color**, căreia i se dau prin intermediul listei cu cele 3 elemente componentele de **Red**, **Green**, **Blue**. Evident, ultimele două sunt **0**, prima având valoarea:

$$Red = (nc - i) \cdot \frac{255}{nc - 1}$$

Se observă că, dacă $i=1$, valoarea expresiei de mai sus este 255, iar dacă $i=nc$, valoarea sa este 0.

8. Definim funcția centrală a programului, pe care o numim DrawCircles. De data aceasta ne alegem o procedură de tip "**Do**", în care punem blocurile de mai jos:



Scopul său este de a desena cercuri concentrice umplute, din exterior către interior, ale căror culori să varieze de la negru le cercul cel mai exterior la roșul cel mai puternic pentru cercul din interior.

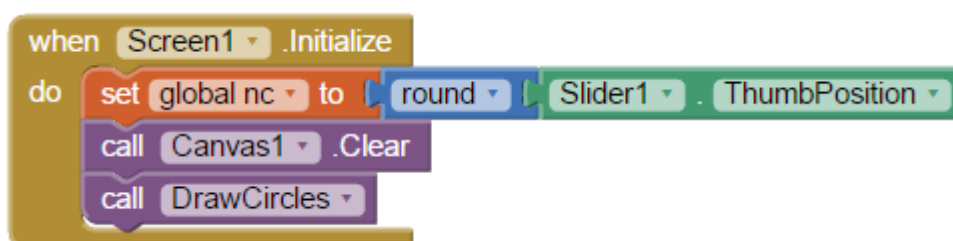
Numărul de cercuri va fi **nc**, deci începem prin a calcula diferența razelor dintre oricare două cercuri vecine, după cum se observă mai sus aceasta este **150/nc**.

Pentru rază utilizăm o variabilă locală **raza**, recunoscută doar în blocul său subordonat.

Continuăm prin utilizarea unei instrucțiuni **for**, care are menirea să numere descrescător, în variabila **i**, (observați pasul **-1** în al parametrului "by") între **nc** și **1**.

În cadrul acestui **for** schimbăm culoarea de desenare cu ajutorul funcției scrise la punctul 7, după care desenăm efectiv cercurile prin apelarea metodei **DrawCircle** a lui **Canvas**. Se observă coordonatele centrelor cercurilor având valori fixe, în mijlocul canvas-ului definit (160,160). Razele cercurilor au valoarea **i*raza**, expresie care ne asigură distanțele egale între ele.

9. A rămas să inițializăm variabila **nc** cu valoarea curentă a cursorului componentei de tip **Slider** și apelarea funcției definite la punctul precedent. Acest lucru se face prin crearea blocurilor următoare, în cadrul evenimentului **Screen1.Initialize**:



Am folosit funcția matematică "**round**" deoarece componenta de tip **Slider** ne întoarce valori cu zecimale, care, ca atare (adică netransformate la numere întregi) nu pot reprezenta un număr de cercuri.

10. În fine, pentru a definitiva aplicația, la fiecare mutare a slider-ului vom șterge desenul anterior, vom actualiza valoarea lui **nc** cu noua valoare a slider-ului și vom redesena. Pentru aceasta, vom crea blocurile următoare, în cadrul evenimentului **Slider1.PositionChanged**:

