Realizarea unui control grafic de vizualizare a progresului a unui GIF.

Astfel eu am convertit giful in elemente de tip vectorial: eu am ales un Progreass spinner(adica o incarcare in forma de cerc)

Astfel pentru convertirea in element de tip vectorial a fost nevoie sa introduc biblioteca SVG(Scalable **Vector** Graphics) pentru transformarea in vector si lucrul cu el, si pentru lucru cu cercul

Astfel am creat o functie care returneaza raza cercului de 5mm, si pozitia sa pe vectorul cx si cy(spinner)

-Controlul are implimentate minim 3 evenimente: astfel primul Este START-atunci cand apasam butonul, ne pare notificarea, al doilea eveniment este Rpogresul , adica cand se incarca totul,

sI ultimul este Stopul- cand se temrina timpul si primim notificare ca evenimentul s-a incheiat.

Pentru ca sa incep tot ptocesul eu am creat o functie Delay Callbak care ne permite sa introducem delayul max pana la 10 s. Si am introdus ca sa puntem introduce valori si cu punct de ex, 2.5s pentru realizarea evenimentului.

Компонент **ProgressBar** (Индикатор прогресса) применяется в тех случаях, когда пользователю нужно показать, что программа не зависла, а выполняет продолжительную работу.

Metode de lucru cu **ProgressBar** :

* **setProgress ()** - setează valoarea specificată a indicatorului de progres;
* **getProgress ()** - returnează valoarea curentă a indicatorului de progres;
* **incrementProgressBy ()** - setează rata de eșantionare a incrementului valorii indicatorului;
* **setMax ()** - setează valoarea maximă a progresului.

XML- un limbaj de marcare extensibil,

XML a fost inițial inventat pentru stocarea și transmiterea mai convenabilă a datelor, inclusiv prin internet. Are o serie de avantaje care fac posibilă rezolvarea cu succes a acestei sarcini. În primul rând, este ușor de citit atât de oameni, cât și de computere. Cred că puteți înțelege cu ușurință ce descrie acest fișier XML:

De asemenea, computerul înțelege cu ușurință acest format.

(paraghaph, headline)

Clasa **AsyncTask** oferă un mecanism simplu și convenabil pentru realizarea operațiunilor care necesită mult timp. Ne oferă comoditatea de a sincroniza si gestiona evenimentele legate de elementele grafice, permițând să vă actualizați elementele de interfață pentru a raporta progresul unei sarcini sau pentru a afișa rezultatele când sarcina este finalizată.

Activitatea-contine tot ce vedem noi(bt,fere)

Fragmentul- Logica aplicatiei si interfata care corespunde unei anumite parti

Intentia- este ecanismul de comunicare dintre elementele unei aplicatii (activitati si servicii)

Serviciu- se refera la un proces ce trebuie sa ruleze in background adica o componenta Android care este utilizata pentru implimentarea de functionalitati complexe, ce implica un volum de procesare mare , de lunga durata, care nu ar elegatura cu utilizatorul

-