

PRG (ETS d'Enginyeria Informàtica) - Curs 2019-2020
Pràctica 1. Resolució recursiva del dibuix d'una figura

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València



Índex

1	RSquare-A. Definició recursiva	1
2	Dibuix d'una RSquare-A amb la llibreria Graph2D	2
3	Activitats en el laboratori	4
3.1	Activitat 1: projecte prg. Instal·lació de la llibreria gràfica	4
3.2	Activitat 2: implementació del càlcul d'una RSquare-A	4
3.3	Activitat 3: implementació del càlcul d'una RSquare-B	6

1 RSquare-A. Definició recursiva

Una **RSquare** és una figura geomètrica plana que s'obté mitjançant la repetició d'un patró bàsic: un quadrat, en que en els seus quatre cantons es poden disposar altres quadrats de costat una fracció de l'original. Al seu torn, en els cantons d'aquests quadrats més xicotets es poden disposar altres quadrats encara més xicotets, i així successivament. El nombre de vegades $n \geq 1$ que es repeteix aquest patró en el dibuix s'anomena l'*ordre* de la figura.

Per senzillesa, aquestes figures es van a classificar per la forma en què aquests quadrats se superposen. De moment, es considerarà el tipus de figura anomenat **RSquare-A**, que es defineix recursivament de la forma següent:

- Una **RSquare-A** d'*ordre* 1, costats de *longitud* l i *centre* en el punt (x, y) , és un quadrat, amb la vora ressaltada¹, de costat l i *centre* en (x, y) .
- Una **RSquare-A** d'*ordre* $n > 1$ i *longitud* l , amb *centre* en el punt (x, y) , és una figura similar a la d'*ordre* 1 però que té, per sota, en cadascun dels seus quatre cantons, una figura **RSquare-A** d'*ordre* $n - 1$, *longitud* $l/2$ i *centre* en aquests cantons.

El quadrat central es dibuixa superposat a les subfigures dels cantons.

Per exemple, en la figura 1 es poden veure unes figures **RSquare-A** d'*ordre* 1, 2 i 3, totes elles de la mateixa longitud l .

¹Açò és, acolorida de forma diferent a la de la resta del quadrat.

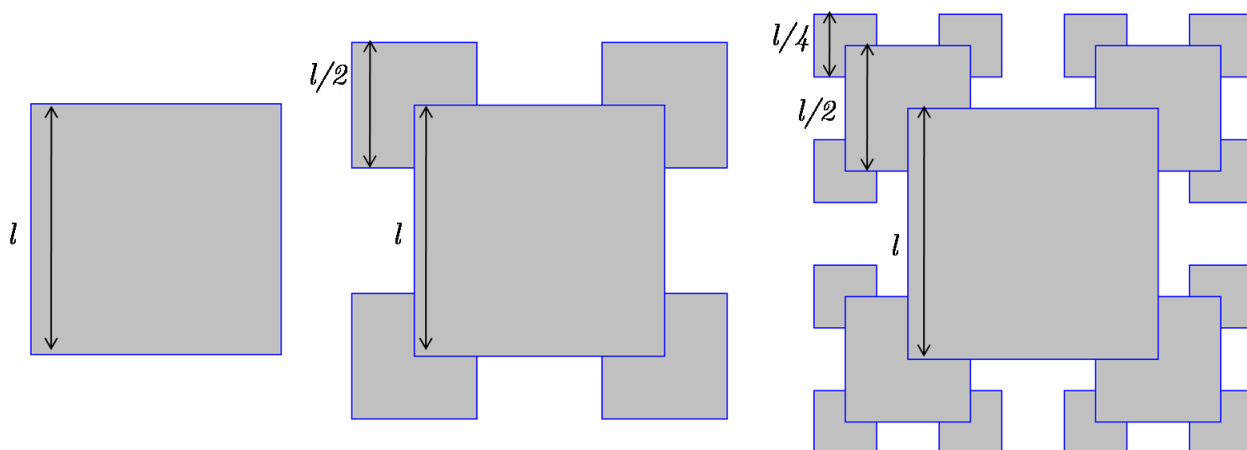


Figura 1: Figures RSquare-A de longitud l i ordre 1, 2 i 3, respectivament.

Noteu que, seguint la definició anterior, el dibuix d'una figura d'aquest tipus d'ordre 4 (per exemple) suposa el dibuix de quatre figures d'ordre 3 (i longitud l'original dividida per 2), situades cadascuna d'elles en els extrems de la figura original i, al seu torn, per aplicació recursiva, el dibuix de cadascuna de les d'ordre 3 implicarà el de quatre figures d'ordre 2 (de longitud dividida) i així, successivament, fins al cas base (figura d'ordre 1).

2 Dibuix d'una RSquare-A amb la llibreria Graph2D

En aquesta pràctica s'ha d'escriure una classe Java que permeti dibuixar figures RSquare-A d'ordre 1 en endavant. Aquesta tasca se simplifica amb l'ús de la llibreria gràfica Graph2D (usada ja en IIP), atès que permet definir finestres gràfiques, de la grandària que es requereixi, i dibuixar en ella rectangles i altres figures geomètriques en la ubicació i amb les dimensions que s'especifiquen.

A l'hora d'escriure el codi es tindran en compte les següents consideracions:

- Per les característiques de la definició de RSquare-A, una figura d'ordre n (per elevat que siga aquest) i de costat l , s'inscriu en un quadrat de grandària fitada de costat

$$l + \frac{l}{2} + \frac{l}{2^2} + \cdots + \frac{l}{2^{n-1}} = 2l - \frac{l}{2^{n-1}}$$

i que en el límit val $2l$.²

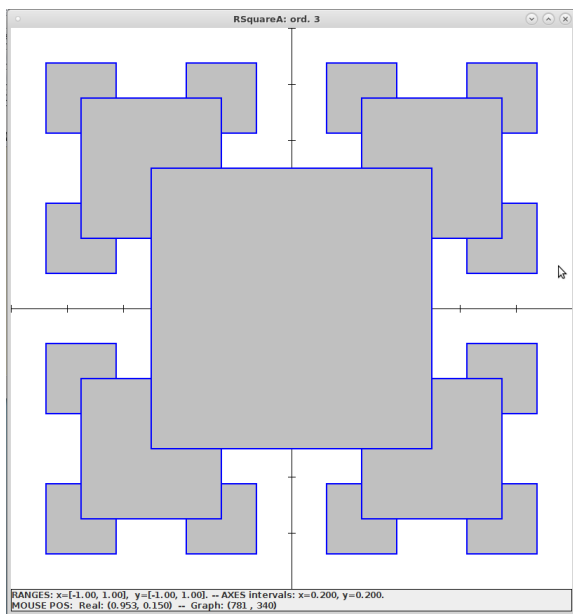
Per simplificar, encara que no és estrictament necessari, es considerarà que la figura de major ordre que el programa vaja a dibuixar, estarà centrada en el punt (0,0) de coordenades de la finestra, i que el seu quadrat central tindrà longitud unitària. Aquest és el criteri seguit en els dibuixos de la figura 2.

²El sumatori $l + \frac{l}{2} + \frac{l}{2^2} + \cdots + \frac{l}{2^{n-1}}$ és la suma dels n primers termes d'una sèrie geomètrica on l és el primer terme i $\frac{1}{2}$ és la raó comuna. Així, $\sum_{k=0}^{n-1} l \frac{1}{2^k} = l \frac{1 - \frac{1}{2^n}}{1 - \frac{1}{2}} = l \frac{2(2^n - 1)}{2^n} = 2l - \frac{l}{2^{n-1}}$

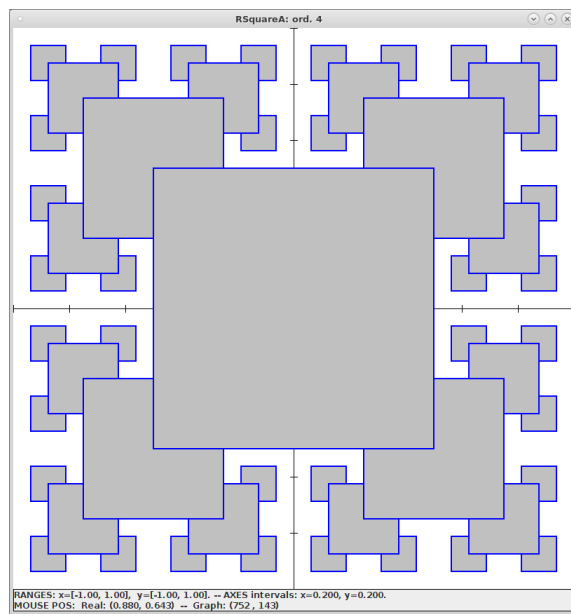
Segons el resultat anterior, açò implica que si es va a dibuixar una figura de costat unitari, n'hi ha prou que la finestra de dibuix es cree amb unes dimensions que estiguen incloses tant en abscisses com en ordenades entre -1 i $+1$ (un quadrat de costat 2). Un exemple del resultat es mostra en la figura 3.

- El dibuix d'un quadrat amb vorell es realitza dibuixant primer el quadrat farcit (mètode `fillRect` de la llibreria gràfica) i després el mateix sense farcit (mètode `drawRect`). Noteu que, tal com indica la seua documentació, aquests mètodes han de rebre les coordenades del cantó superior esquerre del rectangle.
- Perquè totes les parts de la figura es dibuixen recursivament en la mateixa finestra, s'implementarà un mètode que reba com a paràmetre la finestra de dibuix, a més de l'ordre, la dimensió i la ubicació.

Adicionalment, en la mateixa classe s'escriurà un altre mètode que tinga com a paràmetres únicament l'ordre de la figura, i que s'encarregue de crear la finestra de grandària suficient, i d'invocar al mètode anterior per a desencadenar el càlcul del dibuix de la figura amb les dades inicials preestablertes: costat de longitud 1 i centre en $(0,0)$.



(a) RSquare-A d'ordre 3.



(b) RSquare-A d'ordre 4.

Figura 2: Figures RSquare-A d'ordre 3 i 4, respectivament, ambdues de longitud 1, centrades en una finestra de grandària 2×2 .

3 Activitats en el laboratori

3.1 Activitat 1: projecte prg. Instal·lació de la llibreria gràfica

- Crea un projecte *BlueJ* de nom **prg** en `$HOME/DiscoW`. D'ara endavant, en aquest projecte es desenvoluparà el codi de les pràctiques de l'assignatura.
- Crea dins del projecte un paquet **pract1** per a la primera pràctica.
- Descarrega, de la carpeta *PRG:recursos/Laboratorio/Práctica 1* de *PoliformaT*, la classe **RSquare** i afegeix-la al paquet **pract1**.
- La llibreria gràfica (classe **Graph2D** del paquet **graph2D**) es facilita com una llibreria en el fitxer **graphLib.jar** i la seua documentació es proporciona en el fitxer **docGraph2D.zip** (en la carpeta *PRG:recursos/Laboratorio/Librería gráfica* de *PoliformaT*).

Aquesta llibreria s'ha de carregar de la manera adequada (indicant on es troba el fitxer **jar**, que s'haurà descarregat prèviament). Si vas cursar l'assignatura IIP en el primer semestre d'aquest curs, ja la tindràs carregada i, per tant, pots obviar els passos d.1) i d.2) descrits a continuació. Si no la vas cursar:

- Situa el fitxer **graphLib.jar** en el projecte de pràctiques **prg** i, des del menú *Preferències/Llibreries* de *BlueJ*, opció *Add File*, afegeix el fitxer **graphLib.jar**. Hauràs d'engegar novament *BlueJ* perquè la llibreria es carregue.
 - Descomprimeix el fitxer **docGraph2D.zip** en el projecte **prg** per tal de consultar la documentació de la classe **Graph2D** (fitxer **Graph2D.html**) quan siga necessari.
- Descarrega, de la carpeta *PRG:recursos/Laboratorio/Práctica 1* de *PoliformaT*, la classe **FigRecSimple** i afegeix-la al paquet **pract1**. Aquesta classe és un exemple d'ús de la llibreria gràfica en el dibuix d'una figura recursiva.

3.2 Activitat 2: implementació del càlcul d'una RSquare-A

- Completa el codi del següent mètode auxiliar en la classe **RSquare**:

```
/** Dibuixa en la finestra gd un quadrat sòlid de color gris i enmarcat
 * en blau, amb centre en (cX, cY) i costat l.
 */
public static void drawCentSquare(Graph2D gd, double cX, double cY, double l)
```

Una volta implementat, per a provar-lo, crea en el *code pad* un **Graph2D** amb el constructor per defecte (rang d'abscisses $[-1.0, 1.0]$, ídem d'ordenades) i executa'l per a que dibuixe en la finestra creada un quadrat amb centre en $(0, 0)$ i costat de longitud 1. Prova'l amb altres ubicacions i dimensions del quadrat.

Nota que aquest mètode invoca a un mètode privat auxiliar **delay()** ja implementat en la classe, que s'encarrega de retrassar suficientment el dibuix del quadrat. El propòsit d'açò és que, en execució, siga possible observar a simple vista l'ordre en que es van superposant els quadrats que conformen la figura.

- Escriu un mètode recursiu amb perfil:

```

/** Dibuixa en la finestra gd una RSquare-A d'ordre n >= 1, amb centre
 * en (cX, cY) i quadrat central de costat l.
 */
public static void rSquareA(Graph2D gd, int n, double cX, double cY, double l)

```

c) Escriu un mètode guia o llançadora, amb perfil:

```

/** Dibuixa una RSquare-A d'ordre n >= 1, longitud l i centrada en (0, 0). */
public static void rSquareA(int n)

```

que cree una finestra en el rang d'abscisses $[-1.0, 1.0]$ i d'ordenades $[-1.0, 1.0]$, i títol "RSquareA: ord. " + n , i que invocant al mètode anterior dibuixi en la finestra la figura d'ordre n (amb quadrat central de costat 1 i centre en $(0, 0)$).

d) Prova l'execució d'aquests mètodes amb ordres n igual a 3, 4 i 7 i observa que el resultat coincideix amb el de les figures 2(a), 2(b) i 3, respectivament.

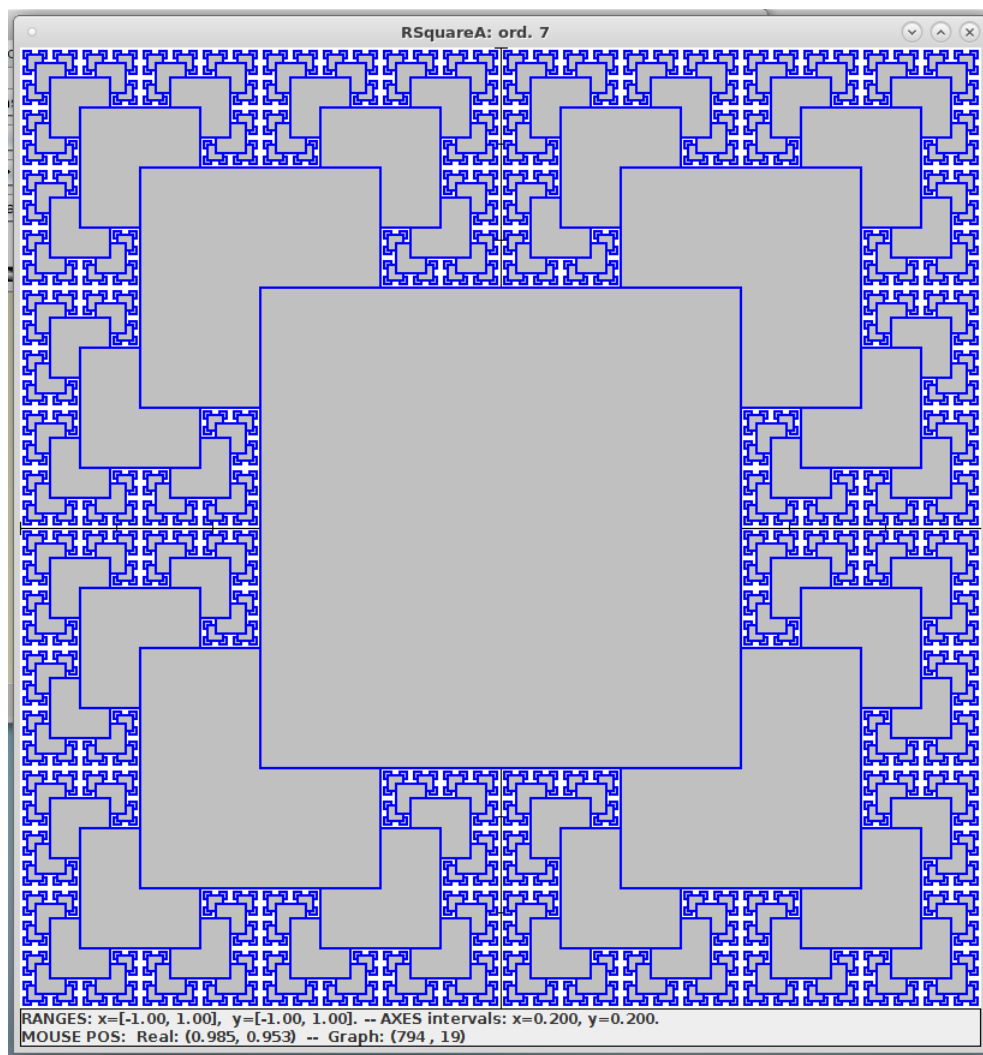
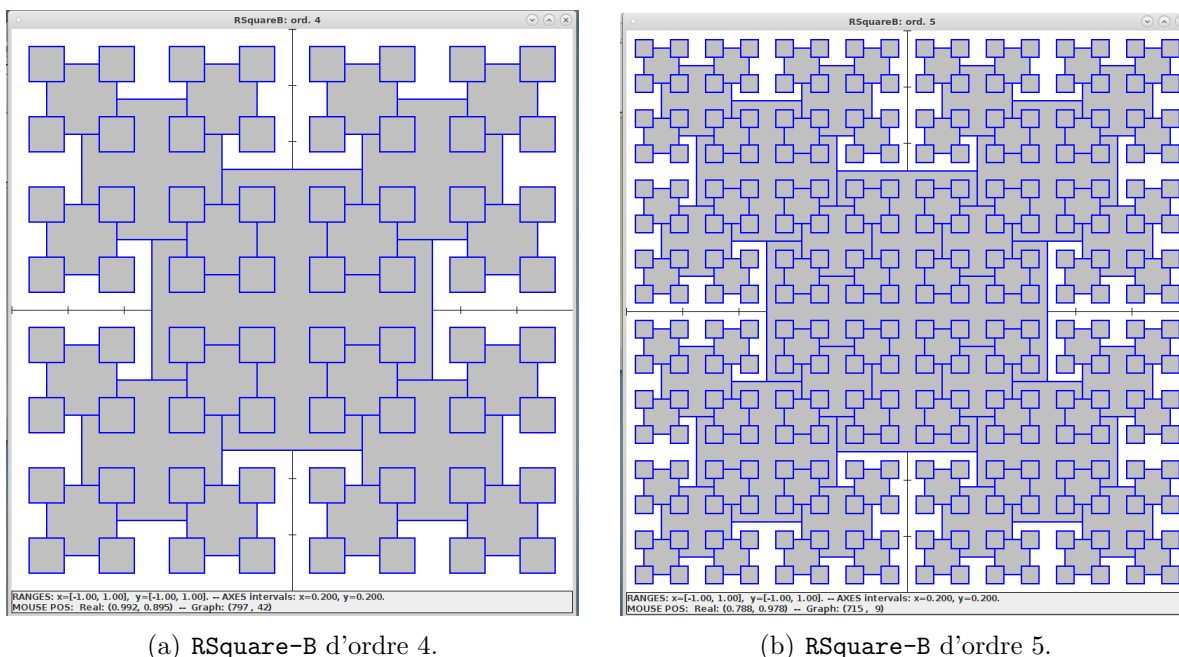


Figura 3: La RSquare-A d'ordre 7 i costat de longitud l ocupa un poc menys que un quadrat de costat $2l$.

3.3 Activitat 3: implementació del càlcul d'una RSquare-B

Com s'ha comentat en el primer apartat, les figures RSquare es poden classificar segons la forma en la que els quadrats se superposen entre si. En la figura 4 es mostra un altre tipus de RSquare, anomenat RSquare-B, en la que els quadrats menors van sempre superposats sobre els majors, originant una distribució distinta.

Aquestes figures tenen una definició recursiva anàloga a la d'una RSquare-A, excepte que, en el cas general, primer cal dibuixar el quadrat més gran, i després superposar-li les subfigures de menor ordre. D'aquesta manera, en el dibuix resultant quedaran al front els quadrats més menuts: els dibuixats en les crides al cas base, com es mostra en els exemples de la figura 4.



(a) RSquare-B d'ordre 4.

(b) RSquare-B d'ordre 5.

Figura 4: Figures RSquare-B d'ordres 4 i 5, respectivament.

En aquesta activitat, hauràs d'afegir a la classe els mètodes recursius anàlegs als de l'Activitat 2, i fer les proves corresponents:

```
/** Dibuixa en la finestra gd una RSquare-B d'ordre n >= 1, amb centre
 * en (cX, cY) i quadrat central de costat l.
 */
public static void rSquareB(Graph2D gd, int n, double cX, double cY, double l)

/** Dibuixa una RSquare-B d'ordre n >= 1, longitud 1 i centrada en (0, 0). */
public static void rSquareB(int n)
```