

Exercici L1-A

El joc del pont

Fonaments de Programació II – Curs 2024-25

Objectiu

L'objectiu d'aquesta pràctica és que practiqueu amb els elements bàsics de C, tipus, variables, constants, operadors, i estructures de control. Per fer-ho, es demana que implementeu un codi que simuli "El Joc del Pont", que es detalla a continuació.

Descripció del joc

L'escenari d'aquest joc consisteix en un pont de vidre, elevat a una certa altura, format per dues plataformes fixes i una sèrie de graons de vidre que permeten passar de la plataforma inicial a la plataforma final.

Cadascun dels graons de vidre està dividit en dues parts, una és de vidre trempat i l'altre és vidre normal. El vidre trempat és suficientment resistent com per a què una persona adulta hi pugui saltar a sobre i no es trenqui. El vidre normal, en canvi, és fràgil i no aguanta el pes d'una persona.

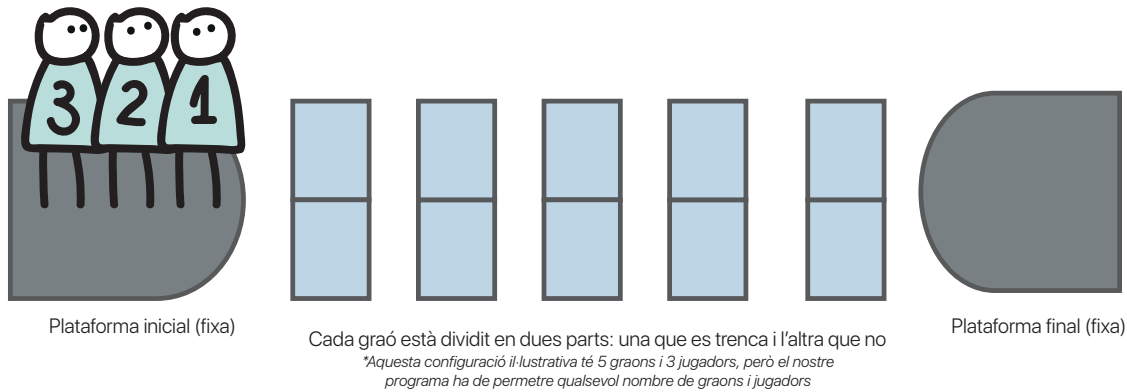
L'objectiu del joc és travessar el pont de vidre i arribar a la plataforma final.

Regles del joc

1. Si un jugador salta damunt del fragment de vidre trempat, sobreviu i pot seguir saltant el graó següent. El jugador pot saltar tants graons com pugui fins arribar a la plataforma final.
2. Si un jugador salta damunt del fragment de vidre normal, el vidre es trenca i el jugador es considera eliminat.
3. Si un jugador ha estat eliminat o ha arribat a la plataforma final, li toca travessar el pont al jugador següent.
4. El jugador següent farà servir els coneixements apresos del jugador anterior: és a dir, ja haurà vist quins fragments són segurs i quins no.
5. Els jugadors segueixen un ordre predeterminat i fix a l'hora de travessar el pont.

6. Els jugadors han de mantenir aquest ordre a l'hora de travessar el pont: no hi poden haver avançaments, ni tampoc hi pot haver més d'una persona al mateix graó.

Els jugadors travessen el pont per ordre



Il·lustració 1. Esquema de l'estructura del Joc del Pont amb una configuració de 3 jugadors i 5 graons.

Implementació del joc

En aquest exercici, implementarem un codi en C (compilable segons l'estàndard C99), que simularà aquest joc. Tindrem N participants que intentaran travessar un pont format per M graons de vidre. La simulació es farà per a una única partida, en la qual els jugadors intentaran travessar el pont un per un.

La mecànica del joc és la següent:

- Els jugadors intenten avançar graó a graó fins a arribar a l'altra banda del pont.
- Cada vegada que un jugador trepitja un nou graó, es determina de forma aleatòria si el graó suporta el pes del jugador o es trenca. Aquest procés es pot imaginar com "llançar una moneda" per cada graó: amb un 50% de probabilitat, el graó és segur i el jugador continua avançant; amb un altre 50%, el graó es trenca i el jugador cau, quedant eliminat. *Consulta l'Annex per saber com generar nombres aleatoris.*
- Un cop un jugador és eliminat o arriba a l'altra banda del pont, és el torn del següent jugador.
- **No s'han de fer servir taules per resoldre aquest exercici.** No és necessari guardar informació sobre l'estat de cada graó abans de començar, ni inicialitzar cap taula bidimensional que representi el pont. La decisió sobre si un graó és segur o no es pren de manera immediata i individual per a cada jugador i graó, utilitzant la funció aleatòria.

- Una de les assumpcions d'aquesta simulació, i que és important recordar, és que estem suposant que els jugadors tenen memòria perfecta. És a dir, que si el jugador 1 ha pogut travessar els dos primers graons i ha caigut al tercer; el jugador 2 ja sap per on ha de passar pels tres primers graons, i comença a decidir a partir del graó 4.

Sortida del joc

El programa imprimirà "Avanço el graó x" o "He caigut al graó x" depenent de la sort que tingui el jugador. Al final del joc, s'imprimirà quants supervivents hi ha hagut. Exemple de sortida per a una configuració de N=8 jugadors i N=5 graons:

```
Jugador 1 intenta travessar el pont...
  El jugador 1 ha caigut al grao 1!
Jugador 2 intenta travessar el pont...
  El jugador 2 avança pel grao 2.
  El jugador 2 avança pel grao 3.
  El jugador 2 ha caigut al grao 4!
Jugador 3 intenta travessar el pont...
  El jugador 3 avança pel grao 5.
Jugador 3 ha arribat a la plataforma final!
Jugador 4 intenta travessar el pont...
Jugador 4 ha arribat a la plataforma final!
Jugador 5 intenta travessar el pont...
Jugador 5 ha arribat a la plataforma final!
Jugador 6 intenta travessar el pont...
Jugador 6 ha arribat a la plataforma final!
Jugador 7 intenta travessar el pont...
Jugador 7 ha arribat a la plataforma final!
Jugador 8 intenta travessar el pont...
Jugador 8 ha arribat a la plataforma final!
```

```
Nombre de supervivents: 6
```

Nota: Aquesta simulació correspon a una única execució del programa. Cada vegada que es torna a executar, el resultat serà diferent, ja que el procés depèn de l'aleatorietat.

Annex: Recordatori sobre com generar nombres aleatoris

Per generar nombres aleatoris necessitem fer servir dues funcions. La primera, anomenada `srand()`, el que fa és establir quina és la llavor del generador de nombres aleatoris. Aquesta llavor és un valor numèric a partir del qual s'hi fan una sèrie d'operacions matemàtiques que finalment genera una seqüència de nombres pseudoaleatoris.

Perquè cada cop que executem el programa aquesta seqüència de nombres sigui diferent, el que es fa és inicialitzar aquesta llavor a la hora del sistema:

```
srand(time(NULL));
```

Perquè la comanda anterior funcioni, necessitem incloure la llibreria `<time.h>`.

Nota: si durant el desenvolupament de l'exercici voleu provocar que cada seqüència generada sigui igual entre execucions diferents, el que heu de fer és inicialitzar la llavor a un valor constant qualsevol, com per exemple `srand(123)`. Això genera que cada cop que executeu el programa la seqüència sigui la mateixa. Compte! No significa que tots els nombres aleatoris siguin els mateixos!

Un cop inicialitzada la llavor, cridarem a la funció `rand()` per obtenir un nombre de la seqüència de nombres pseudoaleatoris. Aquesta funció retorna un enter aleatori entre 0 i `RAND_MAX` (`RAND_MAX` és una constant definida a l'arxiu `stdlib.h` i el seu valor depèn de cada sistema).

```
int aleatori = rand();
```

Amb aquest nombre hi podem operar, de manera que podem fer que aquest nombre aleatori estigui entre certs valors, o transformar-lo a un nombre entre 0.0 i 1.0 (una probabilitat), etc. Per exemple, si `n=10` i fem `rand()%n` això ens donaria un nombre en el rang `[0,9]`. (Nota: com ho fem perquè sigui equivalent a llençar una moneda? Quin mòdul podríem fer?)

Per tal que es pugui cridar `rand()` cal incloure la llibreria `<stdlib.h>`.