Yarn Programming

Andy Price <welshbyte@sucs.org>
SUCS Lightning Talks
October 2017

Crochet

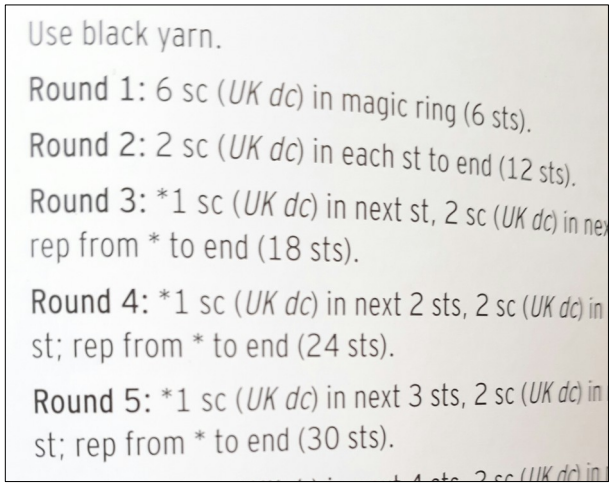
- **Domain-specific language**
- **Human-interpreted**
  - Stitch-in-time compilation strategy
- **Output stored on yarn media**
  - Various types: cotton, acrylic, (sheep) wool, silk, alpaca, etc.
- **Basic unit of data: stitch**
  - Many types: double crochet (basic), virus stitch, bobble stitch…
  - Lesser-used magnitudes: kilostitch, megastitch, liloandstitch
- **No floating point support**
  - Try knitting instead

Basic Requirements

- **Hook**

- **Yarn**

- **Source code**



Use black yarn.

Round 1: 6 sc (*UK dc*) in magic ring (6 sts).

Round 2: 2 sc (*UK dc*) in each st to end (12 sts).

Round 3: *1 sc (*UK dc*) in next st, 2 sc (*UK dc*) in ne
rep from * to end (18 sts).

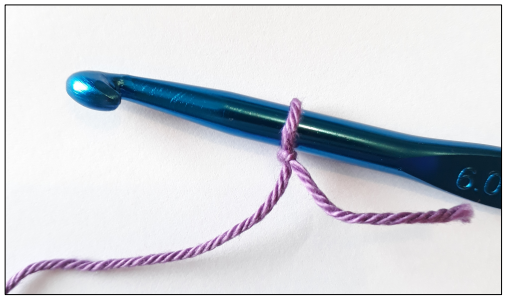
Round 4: *1 sc (*UK dc*) in next 2 sts, 2 sc (*UK dc*) in
st; rep from * to end (24 sts).

Round 5: *1 sc (*UK dc*) in next 3 sts, 2 sc (*UK dc*) in
st; rep from * to end (30 sts).

Getting Started: The “Main” Function

This is the entry point for your program

Slipknot (for rows)



“Magic ring” (for rounds)

Hello World!

Useless program that covers some basics:

```
CH 1 + TCH, SKIP TCH, 1 DC, TIE OFF W/ SL ST
```

1. Create 1 chain and a turning chain
2. Skip the turning chain
3. Do 1 “double crochet” stitch
4. Tie off with a slip stitch

Iteration

Yarn programming:

```
Rows 1 - 4: 2 TCH, DC to end
```

C-like:

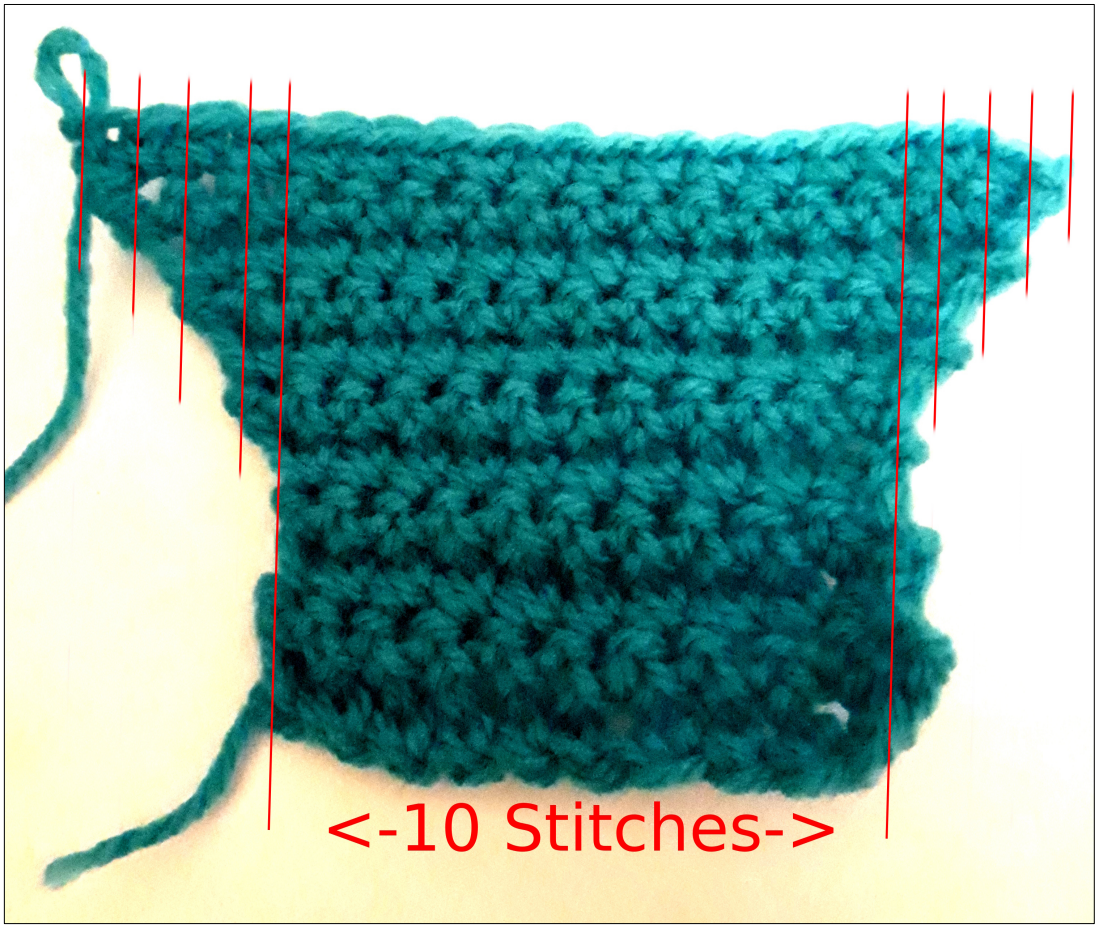
```
for (int i = 0; i < 4; i++) {
      crochet_chain(&work, 2);
      crochet_double(&work, WORK_STITCH_WIDTH);
      work_turn(&work);
}
```

Memory management

Row length increases

- **Memory allocations**
  - Row count
  - Stitch count
  - Stitches per hole
  - Time since last cuppa

- **Stitch markers help**



Modularity

Advantages:

- Reusable, replaceable units
- Easily testable
- Parallelism of work

Disadvantages:

- Sewing together is tedious
- Ends need weaving in!

Granny squares (statically linked)

Disassembly

x86_64:

```
(gdb) disas main
Dump of assembler code for function main:
331    {
   0x000000000001aa48 <+0>:push   rbp
   0x000000000001aa49 <+1>:mov    rbp,rsp
   0x000000000001aa4c <+4>:push   rbx
[...]
```

Yarn programming:

Pull the ends.

Security Considerations

Thanks!

https://twitter.com/andyprice

https://www.ravelry.com/people/andyprice0

Yarn Programming @ Swansea Hackspace “Crafternoons”

2nd & 4th Wednesdays, 7pm

http://swansea.hackspace.org.uk/