TinyExcel: Tiny spreadsheet system

Code structure and step-by-step guide

Tomas Petricek, Charles University

tomas@tomasp.net
@tomasp.net
https://tomasp.net
https://d3s.mff.cuni.cz/teaching/nprg077

```
// In column, row format
// e.g. A1 becomes (1, 1)
type Address = int * int

// Note error is a value!
type Value =
  | Number of int
  | String of string
  | Error of string

// Operators are functions
type Expr =
  | Const of Value
  | Reference of Address
  | Function of string * Expr list

// Using immutable F# map
type Sheet = Map<Address, Expr>
```

Simple start

Standard ML-like expression language

References (instead of variables) are evaluated recursively

Sheet maps (filled) addresses to expressions

```
// Expression and value are
// mutable. Updated triggered
// when they change.
type CellNode =
  { mutable Value : Value
    mutable Expr : Expr
    Updated : Event<unit> }

// Immutable map
// of mutable cells
type LiveSheet =
  Map<Address, CellNode>
```

Version with the dependency graph

Value evaluated
on creation which
prevents circular refs

Expression stored
"drag down" expansion

Updated event
to notify of changes

Charles University

Advanced extensions

Ranges and array values

```
type Value = // (...)
  | Array of Value list

type Expr = // (...)
  | Range of Address * Address
```

	A	B	C	D	E	F
1	Continent	Population	Area	Pop (%)	Area (%)	Density
2	Asia	4753079	31033	52	21	153
3	Africa	1460481	29648	16	20	49
4	Europe	740433	22134	8	15	33
5	North America	604182	21330	6	14	28
6	South America	439719	17461	4	12	25
7	Australia/Oceania	46004	8486	0	5	5
8	Antarctica	1000000	13720	11	9	72
9	World	9043898	143812	100	100	62

Absolute addresses

```
type Index = Fixed of int | Normal of int
type RawAddress = int * int
type Address = Index * Index
```

Charles University

Lab overview

TinyExcel step-by-step

Charles University

TinyExcel - Basic tasks

1. Simple expression evaluator
   With grid references by cell address
2. Drag down formula expanding
   Relocating relative references in formula
3. Reactive event-based structure
   Refactoring code to use graph nodes
4. Reactive event-based computation
   Adding update event handling
5. Rendering sheets as HTML pages
   First step towards a user interface

Charles University

TinyExcel - Bonus and super tasks

1. Absolute and relative addresses
   Alongside with improved "drag down"
2. Adding range selection and array values
   Required for the SUM function
3. Adding change visualization
   Tracking and showing what has changed
4. Full support for live editing
   Updating dependencies in the dependency graph

Where can you use this...

Financial systems

- Live financial models
- Incremental computation with dependency graph

	A	B	C	D	E	F
1	Continent	Population	Area	Pop (%)	Area (%)	Density
2	Asia	4753079	31033	52	21	153
3	Africa	1460481	29648	16	20	49
4	Europe	740433	22134	8	15	33
5	North America	604182	21330	6	14	28
6	South America	439719	17461	4	12	25
7	Australia/Oceania	46004	8486	0	5	5
8	Antarctica	1000000	13720	11	9	72
9	World	9043898	143812	100	100	62

Interesting programming systems

- Live programming systems
- Future more usable programming tools!

Charles University

Lessons learned

A tiny spreadsheet system

 Computation as dependency graph

 Working with two-dimensional grid

 Abstracting from concrete computations

 Good old ML-like expression interpreter

Charles University