

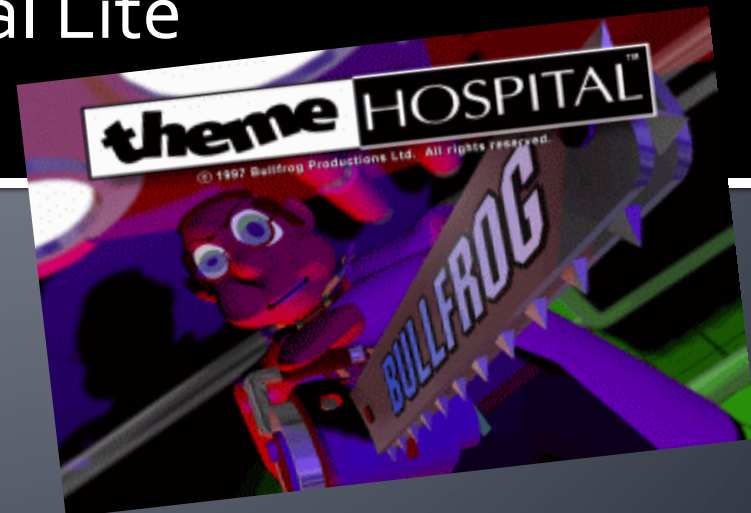
Faculty of Mathematics and Physics
Charles University in Prague
23th March 2015



C# Made Easy!

Programming II

Workshop 06 – Theme Hospital Lite
Part 2 – The Simulation



Workshop 06

Outline



1. No Test
2. Revisiting Workshop 05
3. Assignment 06
 - The Simulation



Test o6

No Test ;)



Find the test here (no-ads):

<http://goo.gl/JYKPXs>

Permanent link:

<https://docs.google.com/forms/d/1pukGlf-5WFISUhxaxiqTNCadZO8xuKDXWnjPb5xXg2k/viewform>

Time for the test:

5 min

Topic

Theme Hospital Lite



Have you played this during last week?



	173613	24 Feb				
						GP's Office Queue Size 23 Queue Expected 2

Topic

Navigation



- For the graph, rooms, etc. check slides from previous Workshop 05 !

- Now let us revisit some problems from Workshop 05
 1. Debugging + ToString()
 2. Dictionary + Composite keys (+ Caches!)
 3. WalkLink vs. LiftLink ... where to hold GetCapacity()
 4. Heap
 5. Dijkstra's Algorithm
 6. Questions?

Tools of Trade

Debugging + ToString()



- Default “String” representation of the object, e.g. Node

```
3 references
public override string ToString()
{
    return "Node[" + Enum.GetName(typeof(NodeType), type) + "-" + id + "]";
}
```

▷ this	{Workshop05.Graph}
▷ fromNode	{Workshop05.Node}
▷ toNode	{Workshop05.Node}
▷ person	{Workshop05.Patient}
▷ item	{Workshop05.SearchItem}
▷ heap	{Workshop05.Heap<Workshop05.SearchItem>}
▷ opened	Count = 0
▷ finished	Count = 1
▷ pathFound	false

▷ this	{Workshop05.Graph}
▷ fromNode	{Node[ENTRANCE-1]}
▷ toNode	{Node[INFODESK-1]}
▷ person	{Workshop05.Patient}
▷ item	{Workshop05.SearchItem}
▷ heap	{Workshop05.Heap<Workshop05.SearchItem>}
▷ opened	Count = 0
▷ finished	Count = 1
▷ pathFound	false

- To be used for DEBUGGING only! Do not misuse for “pretty printing that is handy for your billing application”!

Tools of Trade

Dictionary + CompositeKeys



```
Dictionary<NodeType, Dictionary<int, INode>> nodes = new Dictionary<NodeType, Dictionary<int, INode>>();
```

```
Dictionary<NodeKey, INode> nodeByKey = new Dictionary<NodeKey, INode>();
```

```
6 references
class NodeKey
{
    private int id;
    private NodeType type;
    private int hashCode;

    0 references
    public NodeKey(int id, NodeType type)
    {
        this.id = id;
        this.type = type;
        this.hashCode = 7 * id + 23 * typeof(NodeType).GetHashCode();
    }

    0 references
    public override bool Equals(object obj)
    {
        if (!(obj is NodeKey)) return false;
        NodeKey nodeKey = (NodeKey)obj;
        if (id == nodeKey.id && type == nodeKey.type) return true;
        return false;
    }

    1 reference
    public override int GetHashCode()
    {
        return hashCode;
    }
}
```


Theme Hospital Lite

Navigation - Time



- The link's cost is in "seconds"
- So if lift's cost is "10" it means it travels the link in 10 seconds.
- If person with speedMultiplier 2 is travelling through "walk" link of cost 20, then it means it will take them " $2 * 20 = 40$ " seconds

Theme Hospital Lite

Navigation - Lifts



- Now you will have to simulate LIFTs!
- This means that you have to know where lift “begins”

Lift link: [<lift-left-link> | <lift-right-link>]

lift-left-link: `L<--(lift:c' <capacity> `:t'<cost>
`)-->`

lift-right-link: `<<--(lift:c' <capacity> `:t'<cost>
`)-->L`

Theme Hospital Lite

Navigation - Lifts



- Person (patient or doctor) will always try to use the lift
- When the person arrives to the lift, following cases may occur
 1. Lift is there => Person will immediately use it
 2. Lift is not there & Waiting queue (of lift capacity length) is not full => Person will wait for the lift to arrive
 3. Lift is not there & Waiting queue is full => Person will take detour

Theme Hospital Lite

The Simulation



- Patient's route:
 - Own entrance (you cannot choose this!)
 - > nearest INFODESK
 - > nearest GP that has a doctor inside
 - If no such exist, than just "nearest GP"
 - > nearest special diagnose room that has a doctor inside
 - If no such exist, than just "nearest one"
 - > nearest GP that has a doctor inside
 - If no such exist, than just "nearest GP"
 - > nearest TREATMENT
 - > nearest ENTRANCE

Theme Hospital Lite

The Simulation



- INFODESK / TREATMENT
 - Each info desk / treatment has a „service speed associated“, that is, how much time it needs to “tell the patient how to navigate around the hospital“, resp. “cure the patient”
 - This speed is fixed
 - There can be any number of patients waiting in the queue of an infodesk / treatment
 - Path is determined by the “start service time”

Theme Hospital Lite

The Simulation



- GPs / Specific diagnose room
 - Similar to INFODESK/TREATMENT, but this time, the speed of service is determined by the doctor who is in the room
 - There can be any number of patients waiting in the queue of this room as well

Theme Hospital Lite

The Simulation



- Doctors & GPs / Diagnoses
 - While there are patients in the queue of the room, the doctor won't leave his/her office
 - Whenever there is no queue, two cases may arise
 1. There is no other room that has a patient trying to "use" or navigating to in order to "use" it => doctor stays in his/her current room
 2. There is such a room and
 - 2.1 There is a doctor who is navigating there => doctor ignores it
 - 2.2 There is no doctor travelling there =>
 - 2.2.1 And this doctor is the nearest one => travel there
 - 2.2.2 Is not the nearest one => stays in his/her current room

Theme Hospital Lite

The Simulation



- Doctors & GPs / Diagnoses
 - While there are patients in the queue of the room, the doctor won't leave his/her office
 - Whenever there is no queue, two cases may arise
 1. There is no other room that has a patient trying to "use" or navigating to in order to "use" it => doctor stays in his/her current room
 2. There is such a room and
 - 2.1 There is a doctor who is navigating there => doctor ignores it
 - 2.2 There is no doctor travelling there =>
 - 2.2.1 And this doctor is the nearest one => travel there
 - 2.2.2 Is not the nearest one => stays in his/her current room

Assignment 6

Theme Hospital Lite



INPUT: <int> \n' [<node> '' <link> '' <node> \n']+ <int> \n' [<patient> \n']+ <int> \n' <int> [<infodesk/treatment> \n']+ \n' <int> [<doctor> \n']+ \n'

<node>: <node-type> '-' <id>

<node-type>: ['ENTRANCE' | 'INFODESK' | 'GP' | 'EEG' | 'SONO' | 'XRAY' | 'PSYCHO' | 'TREATMENT' | 'NODE']

<id>: <int>

<int>: [1-9][0-9]{0,1}

<link>: [<walk-link> | <lift-link>]

<walk-link>: [<non-oriented-walk-link> | <oriented-walk-link>]

<non-oriented-walk-link>: '<!--(walk:' <int> ')-->'

<oriented-walk-link>: '--(walk:' <cost> ')-->'

<lift-link>: [<lift-left-link> | <lift-right-link>]

<lift-left-link>: 'L<!--(lift:c' <capacity> ':t'<cost> ')-->'

<lift-right-link>: '<!--(lift:c' <capacity> ':t'<cost> ')-->L'

<cost>: <int>

<capacity>: <int>

Assignment 6

Theme Hospital Lite Navigation



INPUT: <int> '\n' [<node> ' ' <link> ' ' <node> '\n']+ <int> '\n'
[<patient> '\n']+ <int> '\n' <int> [<infodesk/treatment> '\n'
]+ '\n' <int> [<doctor> '\n']+ '\n'

<patient>: <name> ':' <speed-multiplier> ':' <health-
problem> ':' <node> ':' <time>

<name>: [A-Z][a-zA-Z]+

<speed-multiplier>: <int>

<health-problem>: ['CARDIAC' | 'PNEUMONIA' | 'HIP-PAIN' |
'NEUROTIC']

<time>: [0-2][0-9] ':' [0-2][0-9] ':' [0-2][0-9]

Assignment 6

Theme Hospital Lite Navigation



INPUT: <int> '\n' [<node> ' ' <link> ' ' <node> '\n']+
<int> '\n' [<patient> '\n']+ <int> '\n' <int> [
<infodesk/treatment> '\n']+ '\n' <int> [<doctor>
'\n']+ '\n'

<infodesk/treatment>: <node> ':' <service-time>

<service-time>: <int>

<doctor>: <name> ':' <speed-multiplier> ':'
 <service-time>

Assignment 6

Theme Hospital Lite



Output:

Which doctors are you going to use and in which rooms they should begin + when the last patient leaves the hospital (reaches his/her exit ENTRANCE node).

The hospital opens at 08:00:00.

The hospital closes at 18:00:00.

[<doctor-start> '\n']+ <finishing-time>

<doctor-start>: <name> ':' <node>

<finishing-time>: <time>

Assignment 6

Design time!



Assignment 6

Send me an email

- Email: jakub.gemrot@gmail.com
- Subject: **Programming II – 2015 – Assignment 06**
- Zip up the whole project and send it
- You WILL NOT find the assignment in CoDex!
- Deadline: **12.4.2015 23:59**

Questions?

I sense a soul in search of answers...

- Sadly, I do not own the patent for perfection (and will never do)
- In case of doubts about the assignment or some other problems don't hesitate to contact me!
 - Jakub Gemrot
 - gemrot@gamedev.cuni.cz