Faculty of Mathematics and Physics
Charles University in Prague
19th May 2016

C# Made Easy!

Programming II

Workshop 12 – Graph Algorithms

Workshop 12
Outline

1. Test
2. Workshop Finals
3. Graph Algorithms
4. Homework

Questionnaire 1

No Test

Find the questionnaire here (no-ads):

https://goo.gl/JRGhFL

`0 vs. O vs. o, i vs. l vs. 1`

Permanent link:

https://docs.google.com/forms/d/1Yw_BBLhuB-_4-sEEUoPiajO9ijacv99vmaMX-ovmzUM/viewform

Time:

As much as you need

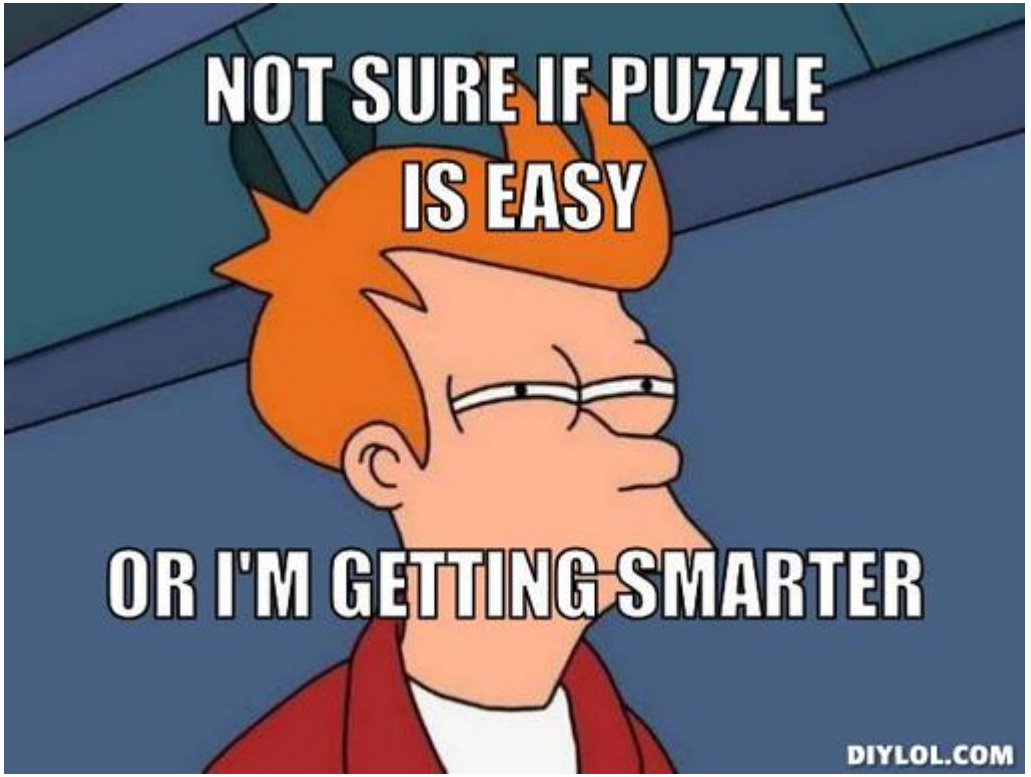
Workshop 12
Finals

You will code the **Final Workshop Test** next week during **Workshop 13, 26.5.2016**

You should revisit:

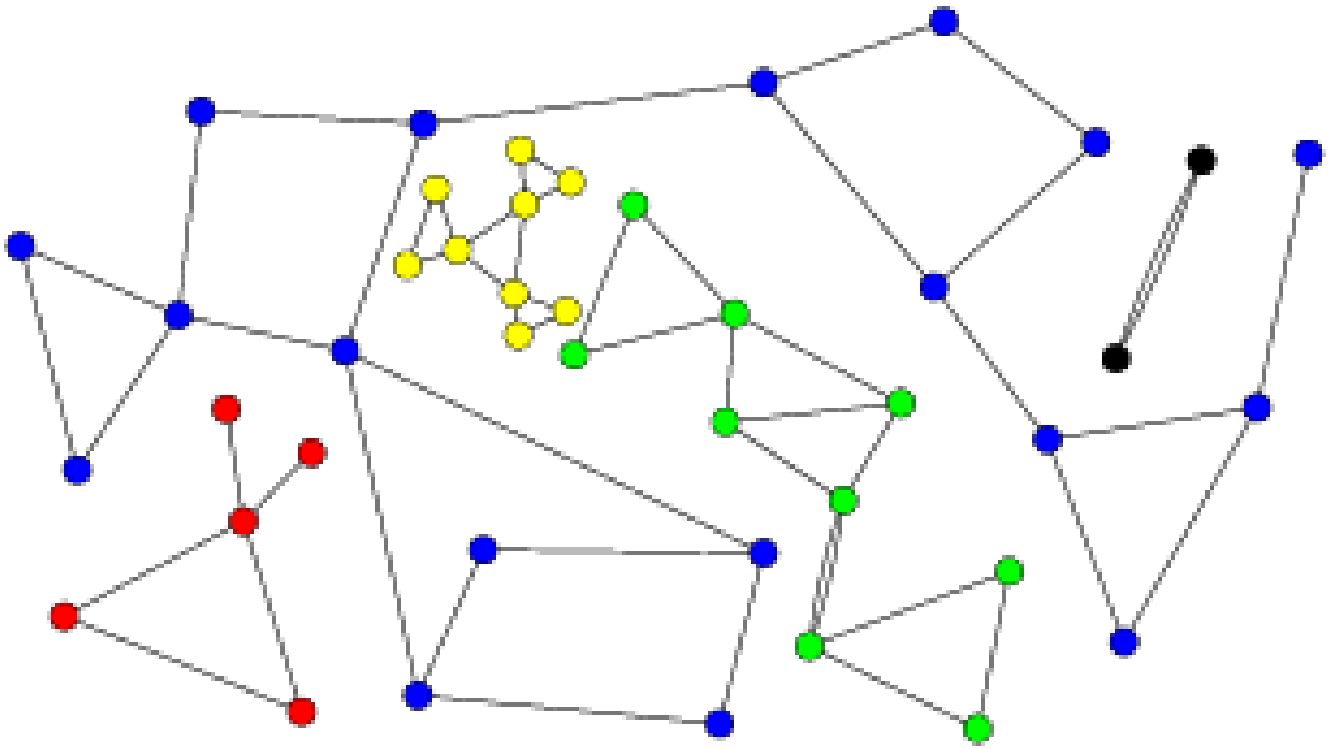
- Recursion
- Graph representation
- DFS, BFS
- Minimax for games

In order to feel like…



Graph Algorithms

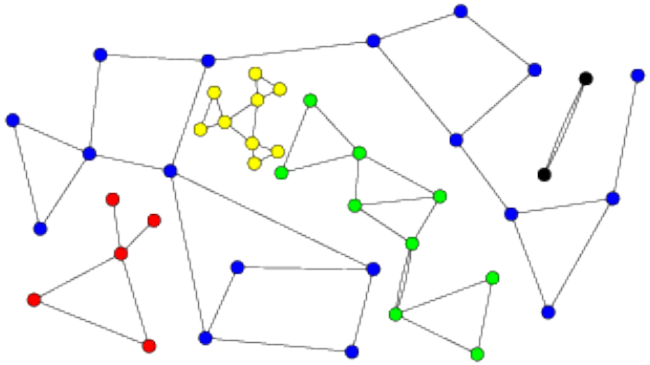
1. Components



Graph Algorithms

1. Components

Algorithm?



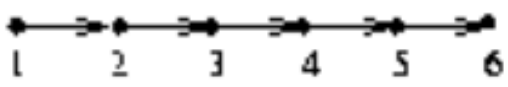
Use BFS or DFS to label nodes of single component, always start from unlabelled node.

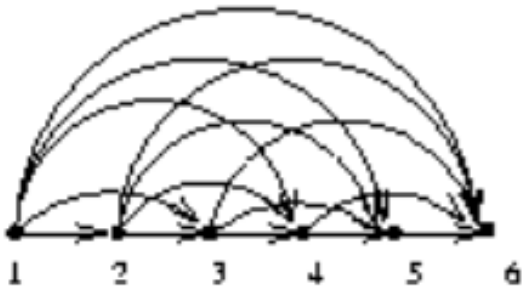
Repeat it as long as there are any unlabelled nodes in the graph.

Complexity?

Graph Algorithms

2. Graph transitive closure

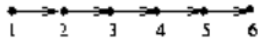


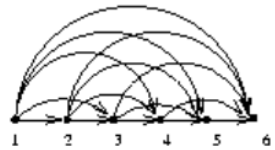


Graph Algorithms

2. Graph transitive closure

Algorithm?



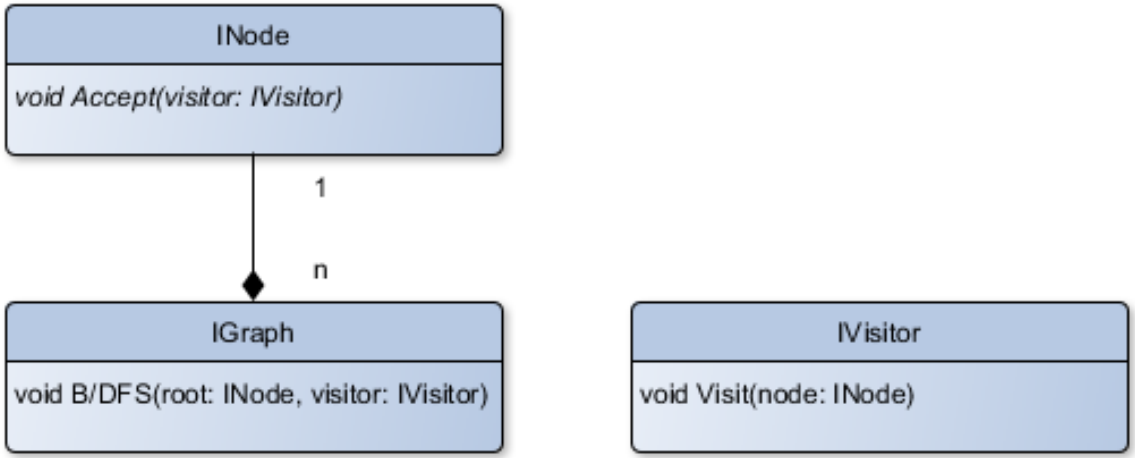


For every Vertex:
Launch DFS or BFS and introduce new edges
when new vertex is reached.

Complexity?

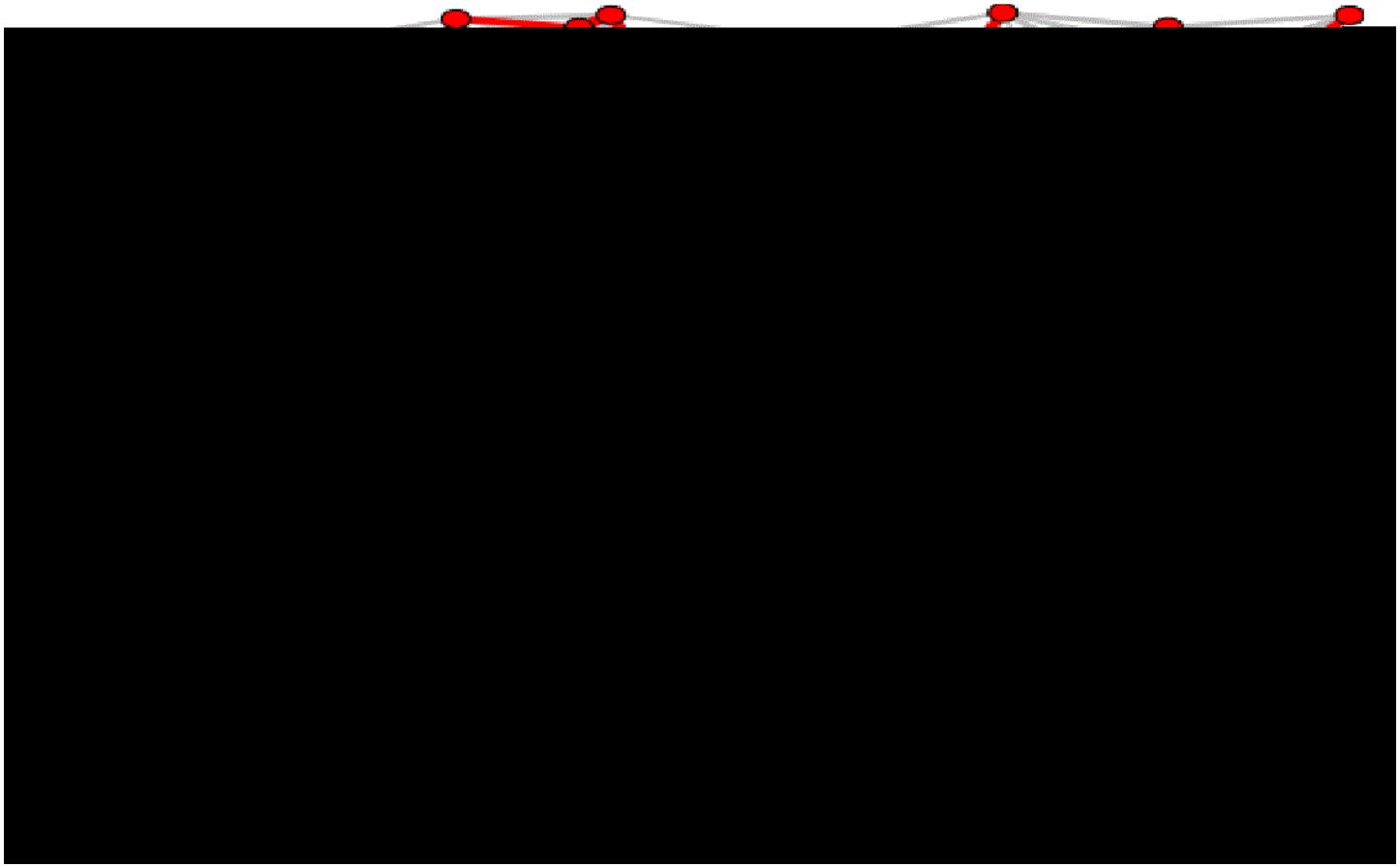
Graph Algorithms
1+2 Implementation?

- We have two graph algorithms using B/DFS … can we somehow split the implementation between "bare" B/DFS and "algorithm internals" ?

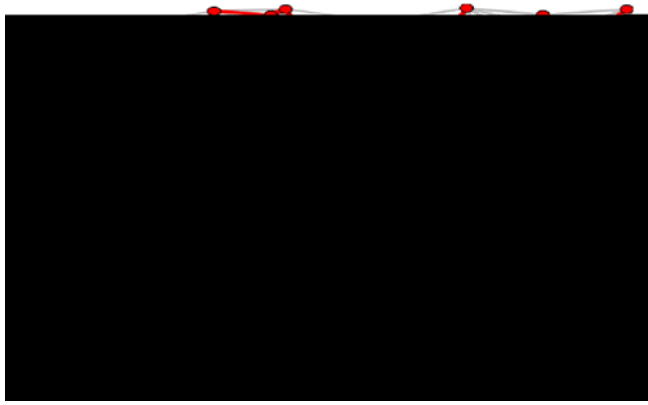


Graph Algorithms

3. Minimum spanning tree



Graph Algorithms
3. Minimum spanning tree



Algorithm?

Kruskal's hungry algorithm:

For every component:

1. Order edges according to their value
2. For each edge … add it to the result if it does not form the circle with already included edges

Complexity?

Assignment 12.1
Graph algorithms

GRAPH INPUT:

<int> '\n' [<node> ' ' <link> ' ' <node> '\n']+

<node> : [a-zA-Z]+

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

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

<oriented-link>: '--(' <int> ')-->'

Assignment 12.1
Graph algorithms

- Implement a GUI application that provides visualization of the graph via spring-algorithm
- Provide buttons for computing:
  - Component labeling
    - 5 points
  - Graph transitive closure of all components
    - 5 points
  - [BONUS] Minimum spanning tree of all components
    - 5 bonus points
- 10 points + 5 bonus points
- Bonus points deadline: 31.5.2016 23:59

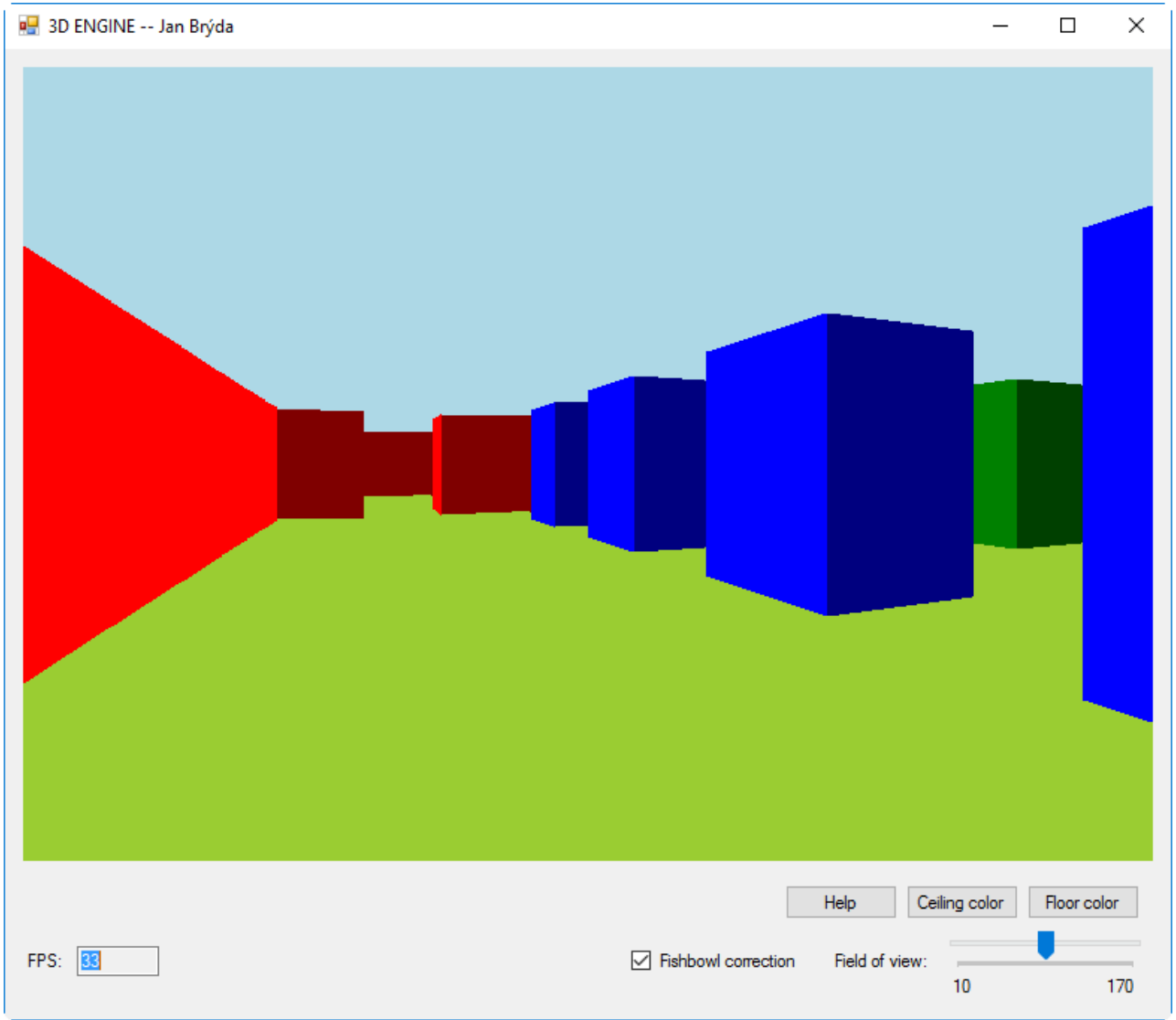
Assignment 12.2

Get ready…

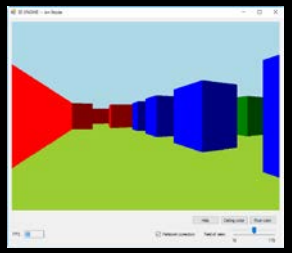
- This is going to be HUGE ☺

Assignment 12.2

3D Engine!



Assignment 12.2
3D Engine!



- Don't get scared!
- Can be written in about 300 lines!

- http://goo.gl/QbcSto
  - Permalink:
  - http://www.playfuljs.com/a-first-person-engine-in-265-lines/

- Input file:

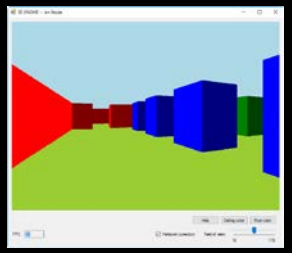
```
NxN
<maze>
```

- Example

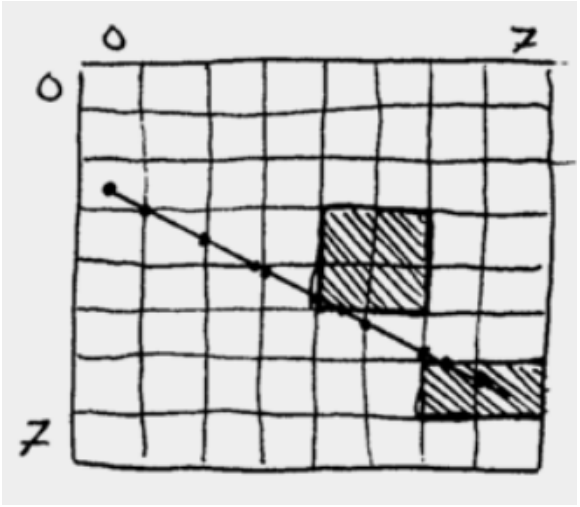
```
10x6
##########
#........#
####.###.#
#..#.#.#.#
#........#
##########
```

- **30+15 bonus points**

Assignment 12.2
3D Engine!



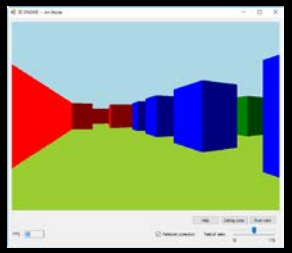
- On Event (button clicked / resize):
  - `//DrawSky()  // optional`
  - `//DrawFloor() // optional`
  - `DrawWalls()`



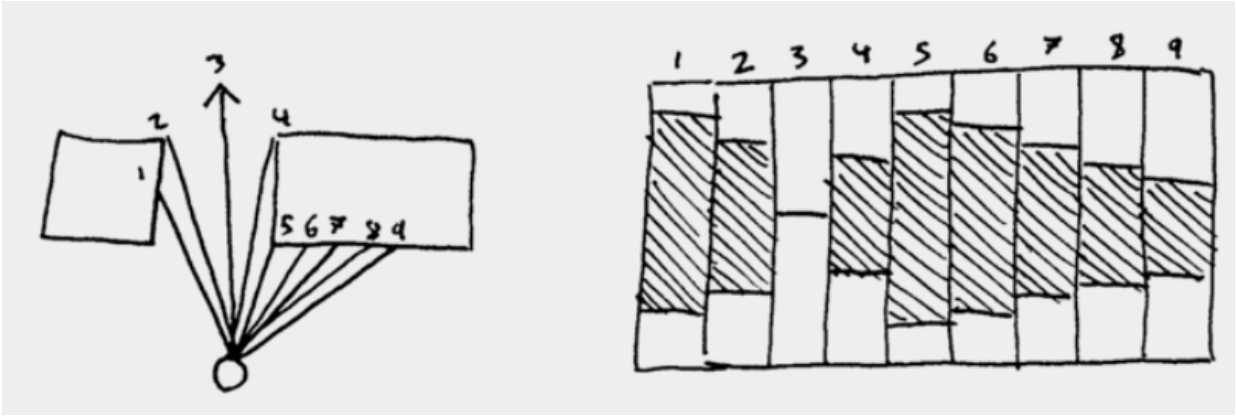
- Drawing walls will need
  - `double Raycast(x, y, angle, max)`
    - `[x;y]` where we're casting the ray from
    - `angle` (horizontal) we're throwing the ray in
    - `max` – limiting raycasting length
    - Returns distance to the wall (or `-1` if the wall not hit in "maxDistance")

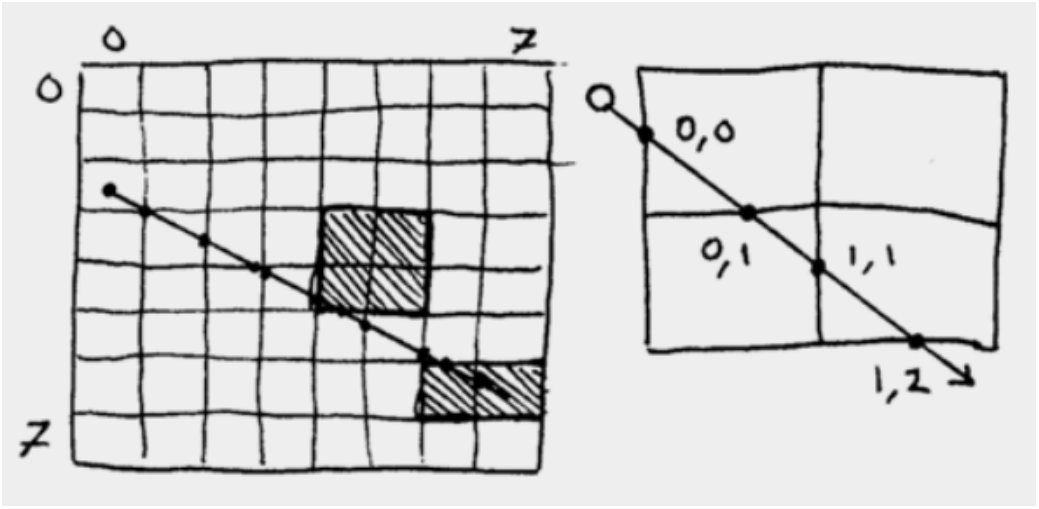
Assignment 12.2

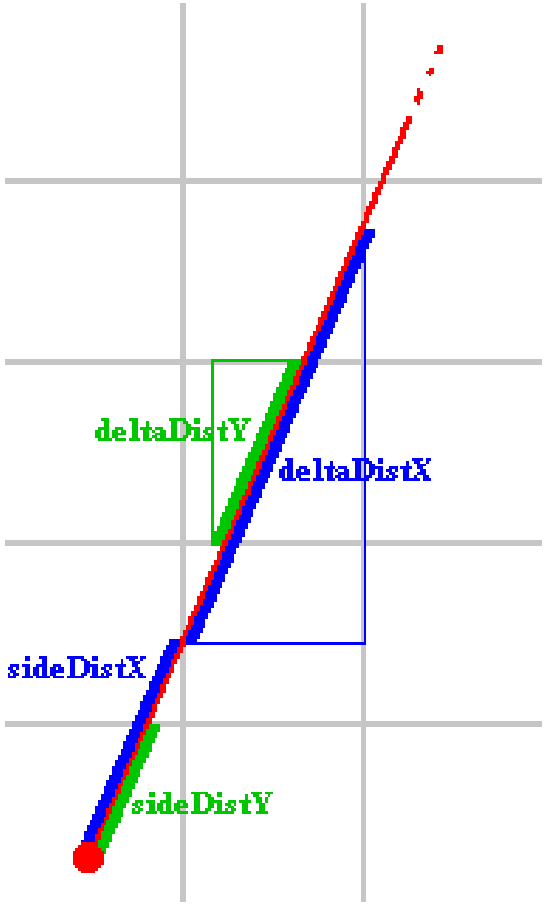
3D Engine!



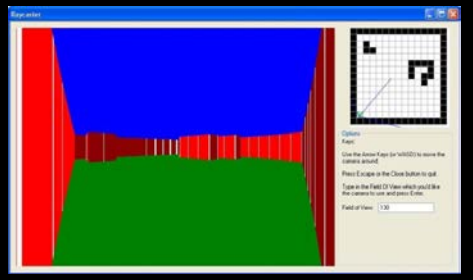
http://lodev.org/cgtutor/raycasting.html







Assignment 12.2
3D Engine



- Different colors for different wall sides
- Redraw on screen resize
- Provide buttons for Movement (turn left/right, move forward/backward)
- Provide color picker for walls
- Provide a way for changing FOV
- Provide a way for disabling "fish-eye" correction

- 30 points + 15 bonus points

- Bonus section
  - Provide (continuous) movement through keyboard (WSAD)
    - 5 bonus points
  - Do not cross walls
    - 5 bonus points
  - Draw sky + floor + provide color pickers for them
    - 5 bonus points

Assignment 12.1 + 12.2
Send me an email

- Email: jakub.gemrot@gmail.com
- Subject: **Programming II – 2016 – Assignment 12.1 / 12.2**
- Zip up the whole solution and send it
- You WILL NOT find these assignments in CoDex!
- Deadline:
  - **30.9.2016 23:59**
- Points:
  - Graph algorithms: 10 + 5
  - 3D Engine: 30 + 15

Questions?

I sense a soul in search of answers…

- In case of doubts about the assignment or some other problems don't hesitate to contact me!
  - Jakub Gemrot
    - gemrot@gamedev.cuni.cz