






Computational Intelligence in Games

Michele Pirovano
Research Assistant at UNIMI, AisLAB
PhD Student at POLIMI
Game Design, Artificial Intelligence, and Games for Health



Milano, 18 Dicembre 2014 1

What is Computational Intelligence?

- «**Computational intelligence (CI)** is a set of nature-inspired computational methodologies and approaches to address complex real-world problems to which traditional approaches are ineffective or infeasible»
- In short terms: practical solutions!
- Examples:
 - Fuzzy Logic
 - Artificial Neural Networks
 - Evolutionary Computation

Milano, 18 Dicembre 2014 2



Schedule



- The REWIRE project and IGER
- Computational Intelligence for Rehab Games
 - Fuzzy Logic for Monitoring
 - Bayesian Adaptation
 - Procedural Content Generation (PCG)
- Computational Intelligence for Games
 - PCG for Games
 - Fuzzy Logic (Fuzzy Tactics)
 - Reinforcement Learning Agents
 - Evolutionary Computation
 - Etc.
- Bonus Topic: Kinect Fusion

Milano, 18 Dicembre 2014 3




Computational Intelligence in REWIRE



Milano, 18 Dicembre 2014 4

REWIRE  **The REWIRE Project** 

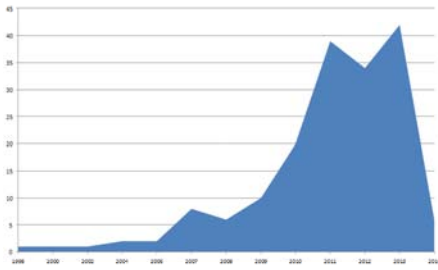
- **Goal**
 - Bring rehabilitation to the patient's home
- **Hierarchy**
 - Networking station
 - Hospital station
 - Patient station



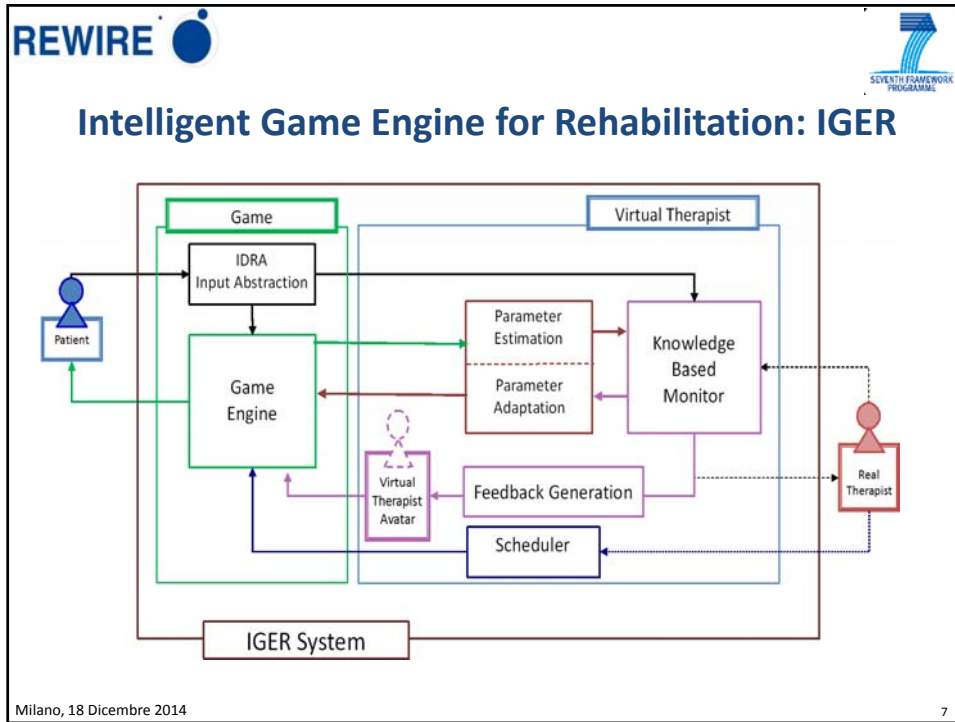
Milano, 18 Dicembre 2014 5



REWIRE  **The State of the Art** 

- **Commercial exergames**
 - Made for entertainment
 - Used at home: risks for safety and efficacy
 - Golomb 2009, Prosperini 2013
- **Ad-hoc rehabilitation exergames**
 - From VRR (not really fun)
 - Many different techniques, devices, goals... a lot of confusion!
- **No guidelines for design or development**
- **How can we design and develop exergames for autonomous rehabilitation?**



Milano, 18 Dicembre 2014 6



REWIRE  **Our Exergames** 



SCARECROW | FRUIT CATCHER | HAY COLLECTOR | HURDLER

BUBBLES | FIRE FIGHTER | BUG CATCHER | HORSE RIDER

BALLOON | WHEEL PUMP




Neglect Games

Milano, 18 Dicembre 2014 8



REWIRE  **Defining guidelines for Exergames** 

- **Guidelines for Autonomous Rehabilitation Exercises**
 - **Efficacy**
 - **Asynchronous Control**
 - Configuration
 - Assessment
 - **On-line Supervision**
 - Adaptation
 - Monitoring
 - Clear and Immediate Feedback
 - **Accessibility and Usability**
- **Guidelines for Motivational and Fun Games**
 - **Basic fun**
 - Meaningful Play
 - Clear and Immediate Feedback
 - Simple and Direct Interactions
 - **Intrinsic**
 - Challenge
 - Fantasy
 - Curiosity
 - Sensation
 - Social Play
 - **Extrinsic**
 - Praise
 - Virtual rewards & Scoring
 - Collection of Exergames



Milano, 18 Dicembre 2014 9

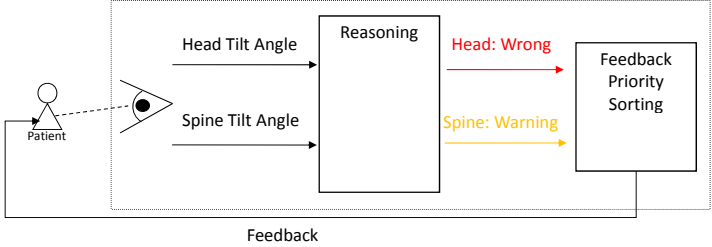
REWIRE  **Monitoring**  

- **Goal:** On-line monitoring and correction of the user's movements
 - Major feature that must be provided even in absence of the therapist



Milano, 18 Dicembre 2014 10

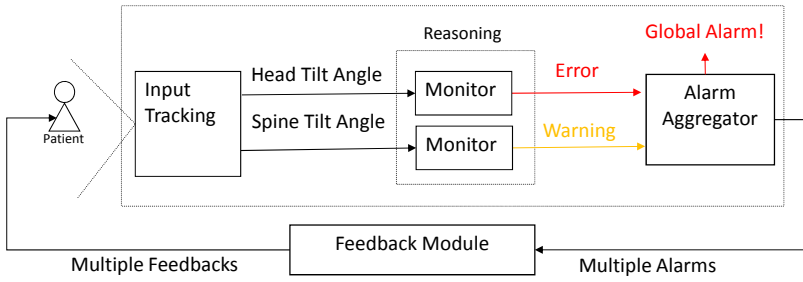
REWIRE  **How the therapist monitors...** 



- Feedback is either verbal or physical. But we cannot touch the patient!
- We can however separate different 'body parts' and treat them disjointly...

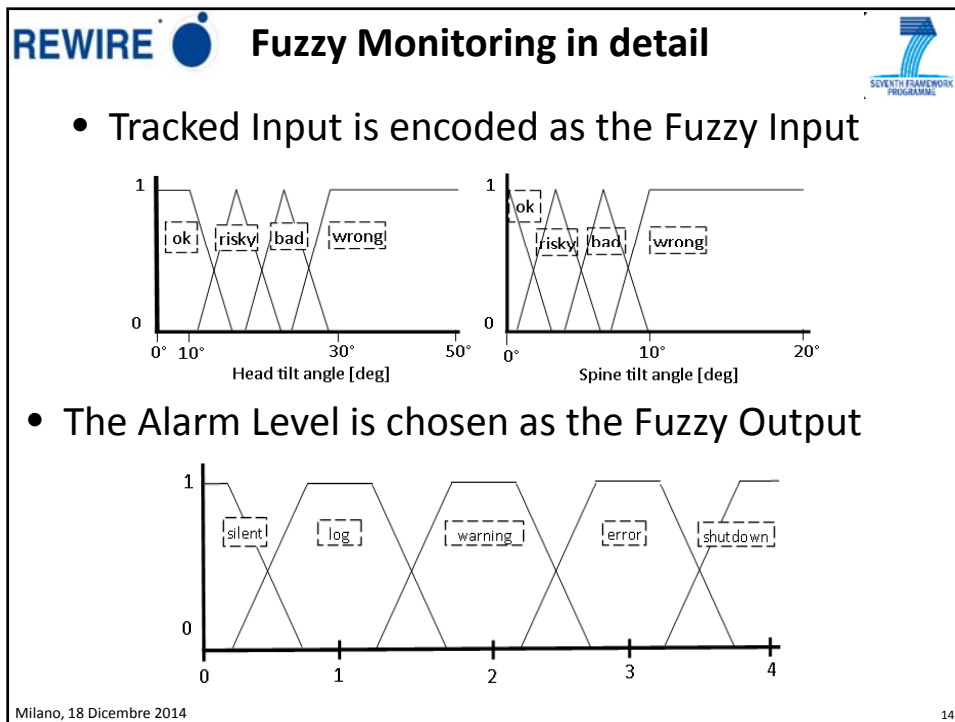
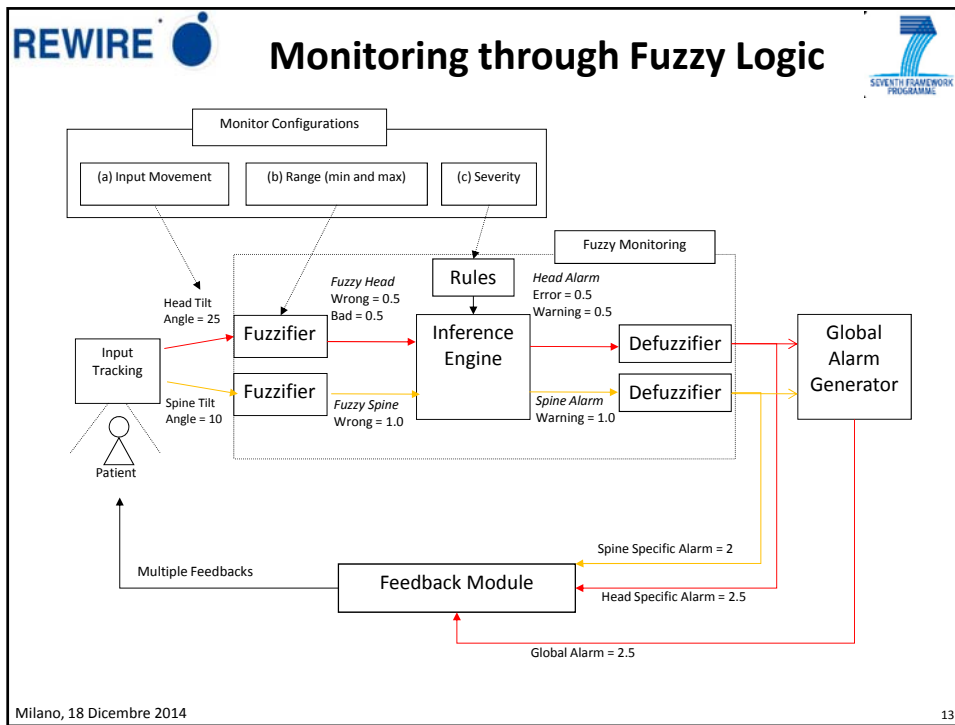
Milano, 18 Dicembre 2014 11



REWIRE  **Monitoring by our Virtual Therapist** 



- Now, we need to create the monitors so that:
 - They encode the therapist's knowledge
 - They can be easily configured by therapists
 - They are robust to uncertainty

Milano, 18 Dicembre 2014 12



REWIRE  **Fuzzy Monitoring in detail** 

- The Fuzzy Rules are generated automatically based on the monitors defined by the therapist

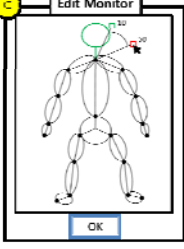
A **Monitors**

Logging	Error	+ Add New Monitor
Features	Features	
Head Frontal Tilt ...	Head Frontal Tilt ...	
Head Sagittal Tilt ...		

B **Features**

Head Frontal Tilt
Head Sagittal Tilt
Spine Frontal Tilt
Spine Sagittal Tilt
Shoulder Front Tilt
...
OK

C **Edit Monitor**





OK

→

If HEAD TILT is OK, then ALARM is SILENT
 If HEAD TILT is RISKY, then ALARM is LOG
 If HEAD TILT is BAD, then ALARM is WARNING
 If HEAD TILT is WRONG, then ALARM is ERROR

Milano, 18 Dicembre 2014 15

REWIRE  **Monitoring: result** 

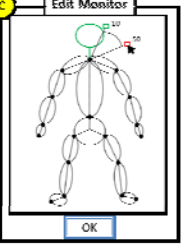
A **Monitors**

Logging	Error	+ Add New Monitor
Features	Features	
Head Frontal Tilt ...	Head Frontal Tilt ...	
Head Sagittal Tilt ...		

B **Features**

Head Frontal Tilt
Head Sagittal Tilt
Spine Frontal Tilt
Spine Sagittal Tilt
Shoulder Front Tilt
...
OK

C **Edit Monitor**




OK

→

Game Engine and VT

←

Tracked Inputs



Milano, 18 Dicembre 2014 16

REWIRE  **Monitoring: video** 

VIDEO TIME!

Milano, 18 Dicembre 2014 17

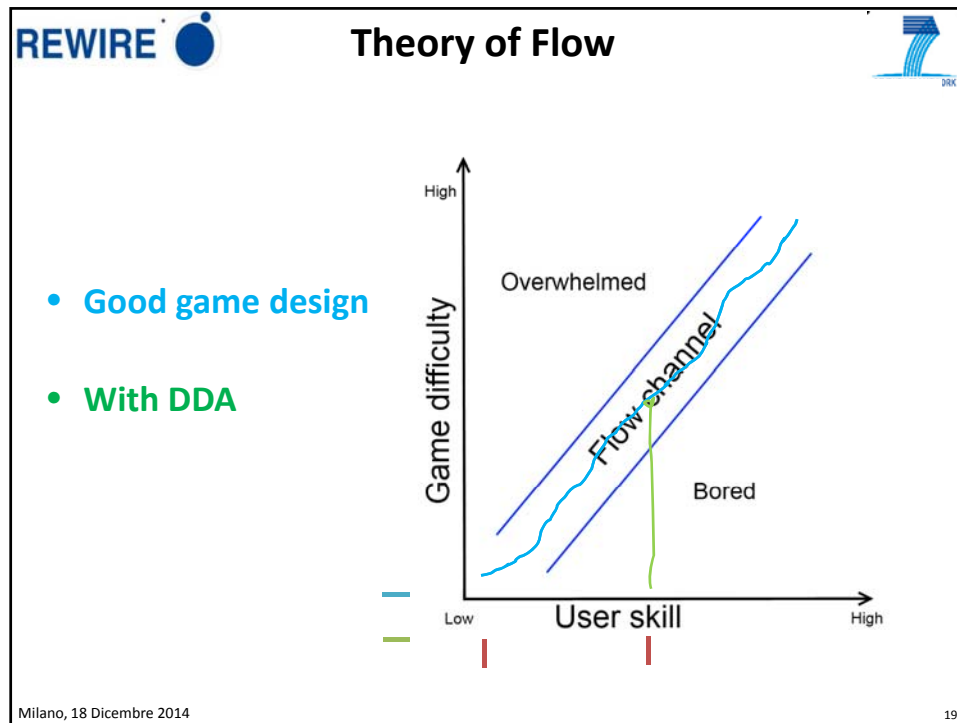
REWIRE  **Dynamic Difficulty Adaptation** 



- **Goal:** adapt the game to the capabilities of the user, maintaining challenge and avoiding frustration
- We want the games to be playable by people with very diverse conditions




- This is done even in entertainment games, but it is much more critical for us!


Milano, 18 Dicembre 2014 18




REWIRE  **DDA concepts** 

- Performance ***P***
 - We estimate how good or how badly the player is doing
- Adapted parameter ***x***
 - This is directly related to the difficulty of the game
 - Examples: enemy health in *Oblivion*, accuracy correction in *Max Payne*
- Adaptation delta ***dx***
 - This is the amount of change of the adapted parameter
- At each adaptation step: **$x = x + dx$**
- And: **$dx = f(P)$**

Milano, 18 Dicembre 2014 20



DDA methods




- Simple heuristics
 - Chose an estimation period (every N trials)
 - At each end of a period estimate the performance as number of hits on the total trials:


$$P = N_{success} / N_{tot}$$
 - Adapt the parameter based on a performance threshold T

$$dx = f(P) \text{ (example: } P/100)$$
 - if $P > T$: $x += dx$
 - if $P < T$: $x -= dx$

Milano, 18 Dicembre 2014
21




Extending DDA with CI




- What can we do to increase DDA efficacy?
 - We want any parameter to be adapted regardless of its actual function
 - ***The adaptation function should take into account previous history***
- Our approach: use the QUEST Bayesian method to estimate the player's skill from previous trials and converge to an optimum

Milano, 18 Dicembre 2014
22




QUEST concepts




- **QUEST: a Bayesian adaptive psychometric method**
[Watson&Pelli 1983]
 - Adapts a threshold towards a success percentage, following a set of trials
- **Goal: find x at which the patient's response is 70%**
- **For DDA, we define:**
 - D : the data, outcome of the previous trials
 - $f_x(x)$: our initial guess of x
 - $f_{x|D}(x|D)$: the posterior pdf, the estimated **success function** given the data
- **Bayes theorem:**

$$f_{x|D}(x|D) = \frac{f_x(x)f_{D|x}(D|x)}{f_D(D)}$$

Milano, 18 Dicembre 2014 23





QUEST method




- This can be rewritten into a recursive function:

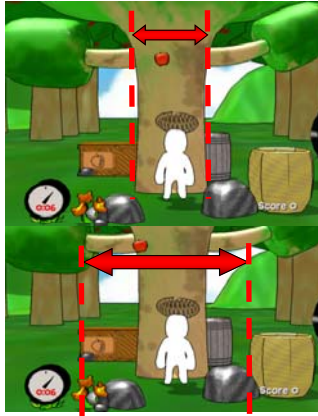
$$Q_i(x) = Q_{i-1}(x) + \begin{cases} S(x - x_i) & \text{if success} \\ F(x - x_i) & \text{if failure} \end{cases}$$
- At each new trial, the parameter value x is taken as the mean of $Q_i(x)$

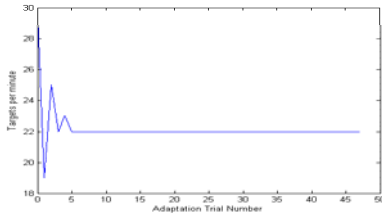
Milano, 18 Dicembre 2014 24

REWIRE  **Results** 

- The parameter gets adapted and converges to an optimal value after a few trials










Milano, 18 Dicembre 2014 25

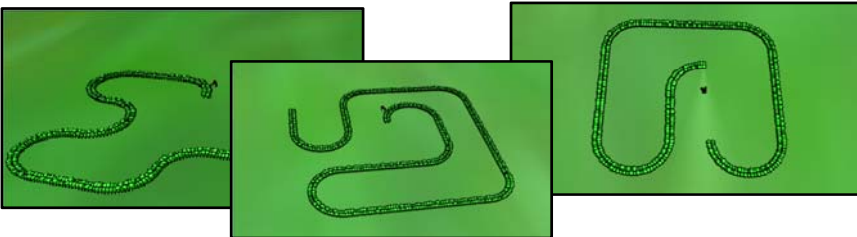
REWIRE  **Adaptation Video** 

VIDEO TIME!



Milano, 18 Dicembre 2014 26

REWIRE  **Controlled Randomization**  

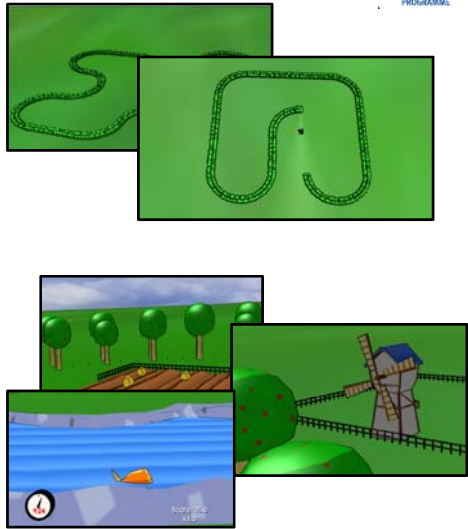
- Remember that the patients will have to perform the same exercise many, many times!
- Even if the games are fun, they would be come boring pretty soon
- **Goal:** introduce small changes to the games to make them more variable and thus more interesting




Milano, 18 Dicembre 2014 27

REWIRE  **What do we randomize?** 


- Gameplay
 - Movement patterns
 - Gameplay parameters
- Graphics
 - Placement of *props*
 - Colors and textures
- Music
 - Randomized choice of background music



Milano, 18 Dicembre 2014 28




Procedural Content Generation




- We can go further with the whole «randomizing» thing: we can procedurally generate actual assets from scratch!
 - *“Procedural content generation (PCG) is the programmatic generation of game content using a random or pseudo-random process that results in an unpredictable range of possible game play spaces.”* - <http://pcg.wikidot.com/>


Milano, 18 Dicembre 2014
29





PCG in Rewire




- A first PCG method is already used for generating the paths
 - Starts from one end
 - Travels randomly
 - Avoids self-collisions





Milano, 18 Dicembre 2014
30

REWIRE  **PCG in Rewire** 

- For the reward system of REWIRE, we aim to let the patient build a farm out of generated assets





Milano, 18 Dicembre 2014 31


REWIRE  **PCG in Rewire** 

- We generate plants using L-systems:
 - A Lindenmayer system is a parallel rewriting system and a type of formal grammar
 - It possess a high degree of self-similarity
 - Like musical scores or like plants!
- Axiom: ***a***
- Production: ***a -> ab***
- Iterations:
 - ***a***
 - ***ab***
 - ***abb***
 - ***abbb***



Milano, 18 Dicembre 2014 32












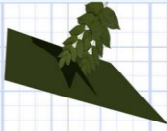















REWIRE  **PCG in Rewire** 

- Results:



Milano, 18 Dicembre 2014 33

REWIRE  **Interactive Evolutionary Computation** 



		
2 g:0 v:9 s:4  	18 g:0 v:10 s:4  	57 g:7 v:8 s:4  
		
34 g:1 v:9 s:4  	76 g:12 v:12 s:2  	78 g:12 v:12 s:2  
		
66 g:9 v:12 s:2  	77 g:12 v:11 s:2  	74 g:11 v:16 s:2  

Milano, 18 Dicembre 2014 34

REWIRE  **PCG in Rewire: Video** 

VIDEO TIME!

Milano, 18 Dicembre 2014 35

REWIRE  

Other Applications of CI in games

Milano, 18 Dicembre 2014 36

REWIRE  **Procedural Content Generation** 

- In commercial games:





Milano, 18 Dicembre 2014 37

REWIRE  **PCG: Dungeon Generator** 

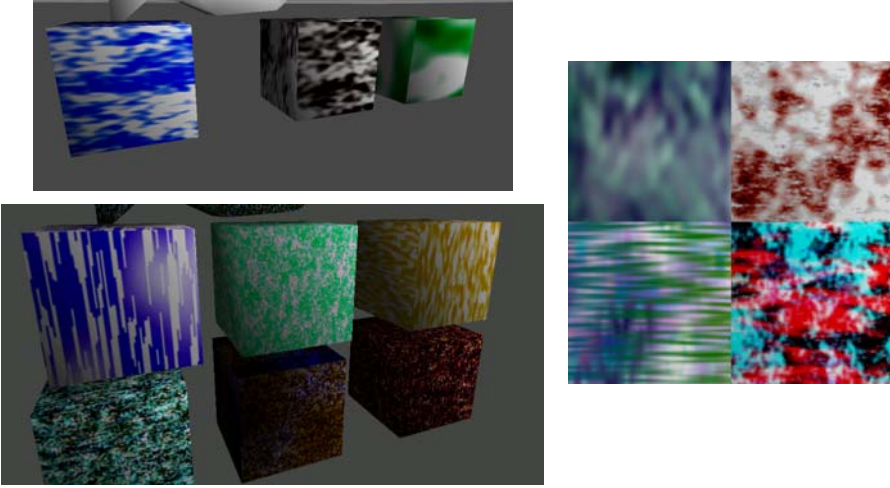
- Recursive Backtracker algorithm generates a maze
- Parameterized for ease of use



Milano, 18 Dicembre 2014 38

REWIRE  **PCG: Textures** 

- Layers of simplex noise






Milano, 18 Dicembre 2014 39

REWIRE  **PCG: Terrain** 





Milano, 18 Dicembre 2014 40

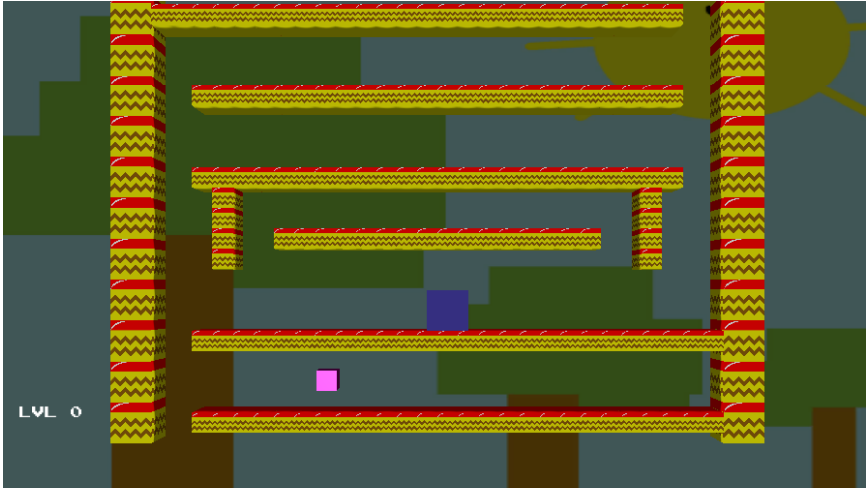
REWIRE  **PCG: Sprites & Meshes** 



Milano, 18 Dicembre 2014 41



REWIRE  **PCG: Level design** 

- Randomized according to difficulty



<http://www.indiedb.com/games/bulla-bolla>

Milano, 18 Dicembre 2014 42

REWIRE  **PCG: Swords** 

VIDEO TIME!

Milano, 18 Dicembre 2014 43

REWIRE  **Fuzzy Logic** 

- In commercial games:



Milano, 18 Dicembre 2014 44

REWIRE  **Fuzzy Tactics** 



Milano, 18 Dicembre 2014 45

REWIRE  **Fuzzy Tactics: Logic Building** 



NGUI
Unif: Marco

Orders

Order	Remove	Up	Down
1. Get Closer	X	↑	↓
2. Attack	X	↑	↓



Library Add

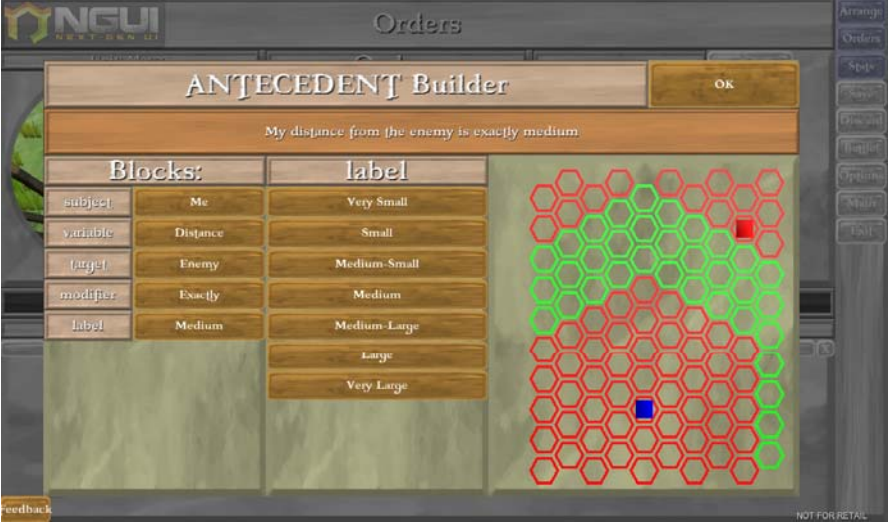
Attack

Conditions	Consequences
My distance from the enemy is exactly medium X	Attack enemy X

feedback NOT FOR POSTAL

Milano, 18 Dicembre 2014 46

REWIRE  **Fuzzy Tactics: Logic Building** 



Blocks:	label
subject: Me	Very Small
variable: Distance	Small
target: Enemy	Medium-Small
modifier: Exactly	Medium
label: Medium	Medium-Large
	Large
	Very Large

Milano, 18 Dicembre 2014 47

REWIRE  **Fuzzy Tactics: Video** 

VIDEO TIME!

Milano, 18 Dicembre 2014 48

REWIRE Evolutionary Computation 





Hastings 2012 - Evolving Content in the Galactic Arms Race Video Game


Milano, 18 Dicembre 2014 49

REWIRE Evolutionary Computation: Video 

VIDEO TIME!

Milano, 18 Dicembre 2014 50

REWIRE  **Reinforcement Learning** 




<http://homes.di.unimi.it/~pirovano/games/RLogue/RLogue.html>


Milano, 18 Dicembre 2014 51

REWIRE  **Reinforcement Learning: Video** 

VIDEO TIME!


Milano, 18 Dicembre 2014 52

REWIRE 




Bonus: Kinect Fusion

Milano, 18 Dicembre 2014 53

REWIRE 

Kinect Fusion



Kinect Fusion is an algorithm developed by Microsoft Research in 2011. The algorithm allows a user to reconstruct a 3D scene in real-time and robustly by moving the Microsoft Kinect sensor around the real scene.

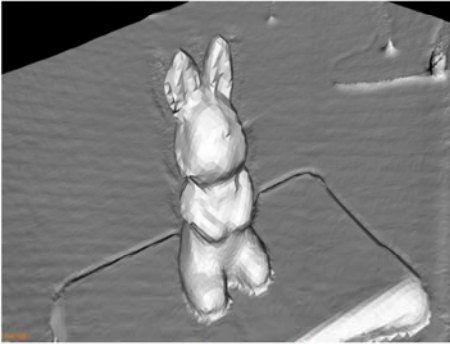




Figure 1 - A rabbit-like statue reconstructed with Kinect Fusion

<http://homes.di.unimi.it/~pirovano/pdf/3d-scanning-pcl.pdf>

Milano, 18 Dicembre 2014 54

REWIRE  **Kinect Fusion: Algorithm** 

- Step 1: Get the depth images from Kinect

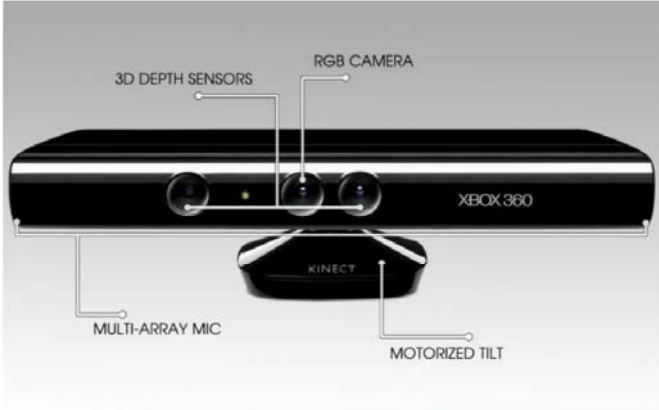





Figure 3 - The Microsoft Kinect sensor



Milano, 18 Dicembre 2014 55

REWIRE  **Kinect Fusion: Algorithm** 

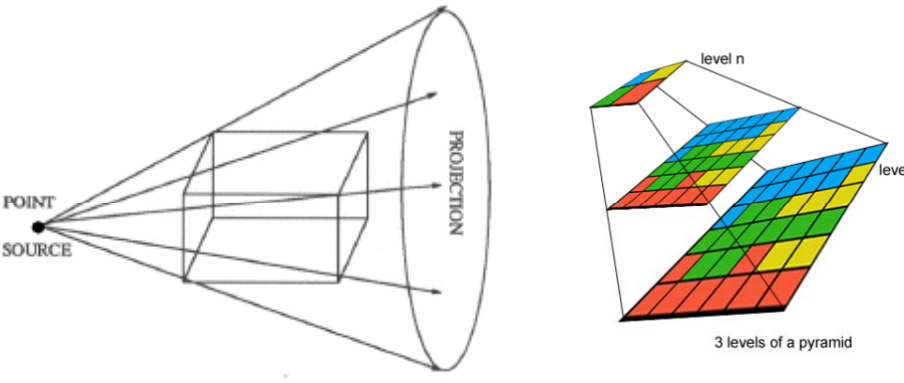
- Step 2: Apply a bilateral filter to the images





Milano, 18 Dicembre 2014 56

REWIRE  **Kinect Fusion: Algorithm** 

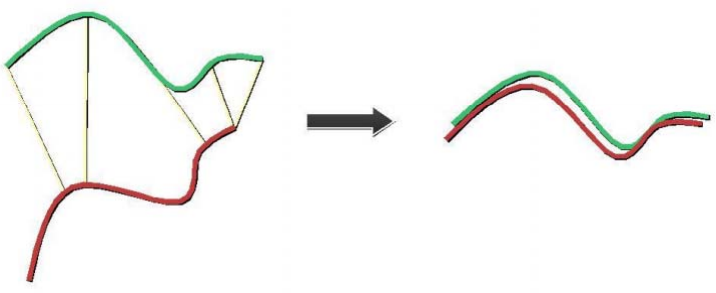
- Step 3: Create a multi-resolution pyramid of 3D clouds through back-projection
 - With vertex and normal information





Milano, 18 Dicembre 2014 57

REWIRE  **Kinect Fusion: Algorithm** 

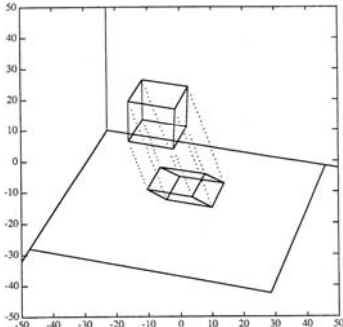
- Step 4: Point clouds from subsequent frames are aligned using Iterative Closest Point (ICP)
 - ICP returns a 6dof orientation matrix
 - Uses the multi-resolution pyramid for performance





Milano, 18 Dicembre 2014 58

REWIRE  **Kinect Fusion: Algorithm** 

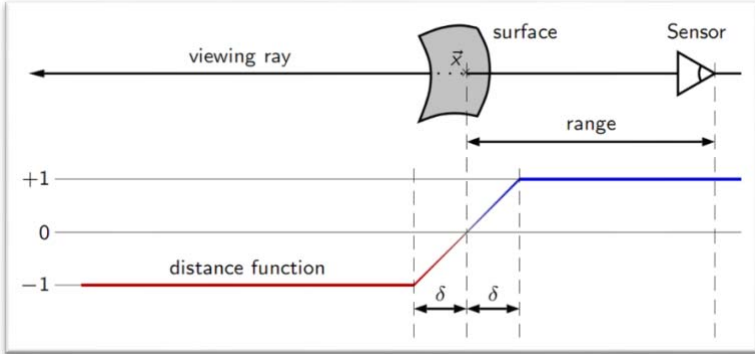
- ICP is modified with a few optimizations:
 - Assume small changes in the camera movement
 - the two clouds are projected onto the same image
 - Points are matched if they fall on the same pixel





Milano, 18 Dicembre 2014 59

REWIRE  **Kinect Fusion: Algorithm** 

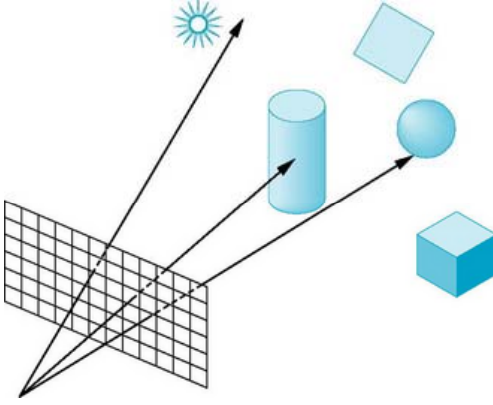
- Step 5: The new cloud is merged with the current compound model to create the surface
 - The surface is extracted using a Truncated Signed Distance Function (TSDF)





Milano, 18 Dicembre 2014 60

REWIRE  **Kinect Fusion: Algorithm** 

- Step 6: Raycasting is performed to render the surface



Milano, 18 Dicembre 2014 61

REWIRE  **Kinect Fusion: Algorithm** 

- Final result!

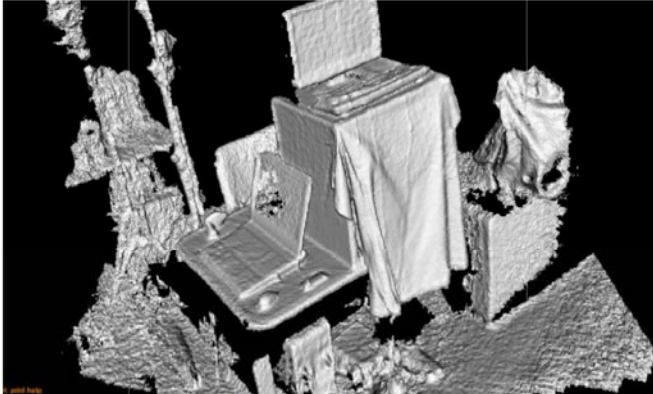




Figure 2 - Part of the AIS-lab at the University of Milan, reconstructed with Kinect Fusion

Milano, 18 Dicembre 2014 62

REWIRE  **Kinect Fusion: Algorithm** 

VIDEO TIME!

Milano, 18 Dicembre 2014 63

REWIRE  **Thank you!** 

michele.pirovano@unimi.it



Milano, 18 Dicembre 2014 64