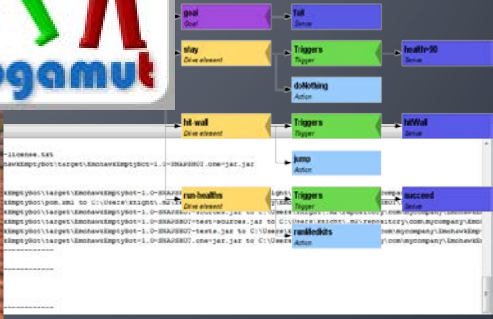
Faculty of mathematics and physics
Charles University at Prague
24th May 2016

UT2004 & UE2 bots made easy!

Pogamut 3

Lecture 13 – Conclusion



Today's menu

1. **Recapitulation**
   - **What has been explained**
2. Checklist for pretest / exam
   - What you should have learned
3. What's next?

Recapitulation I
General Topics

- What has been explained…
  - Basics of Java (ArrayList, HashMap, HashSet, Collections, chaining of setters…)
  - Basics of Pogamut
  - Basics of UT2004 (DeathMatch, CTF, Weapons)
  - Basic problems bots have to solve in 3D realtime environments
  - Basics of AI development
  - yaPOSH

Recapitulation II

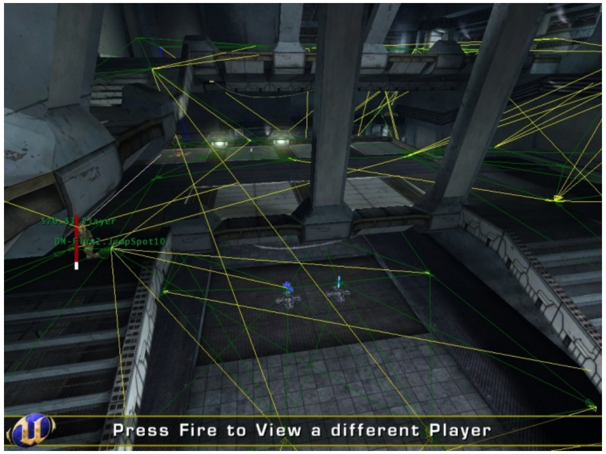
Pogamut modules

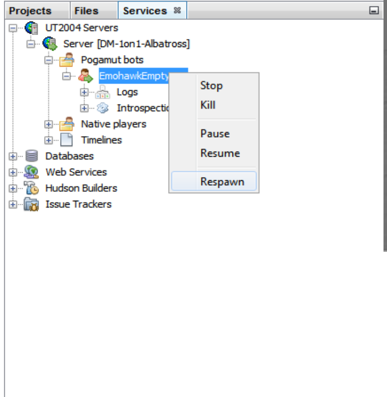
- You should remember Pogamut modules:
  - `move, navigation`
  - `shoot, weaponPrefs`
  - `players, items`
  - `fwMap, aStar, visibility`
  - `MyCollections, DistanceUtils`
  - `world, ctf`
- And how to setup general event listeners
  - `@EventListener(eventClass=TeamChat.class)`

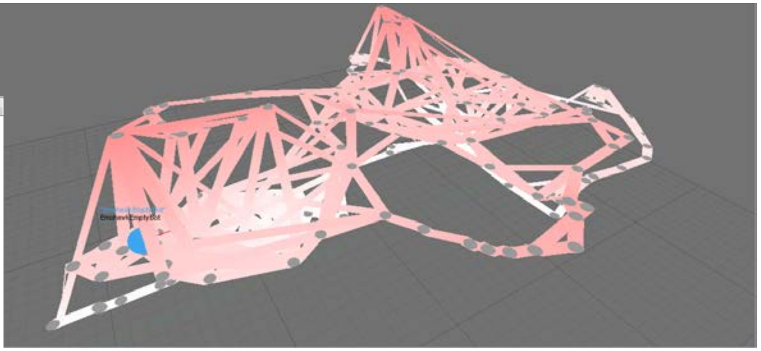
Recapitulation III

Coding and debugging

- You should know how to code the bot and how to debug the code in Pogamut effectively
  - Pogamut NetBeans plugin for map visualization
  - Logging tricks (output window, bot name, text messages, etc.)
  - Debug info in UT2004 provided by GameBots (CTRL + H)
  - How to read Pogamut exceptions
  - What to do when NB aren't showing Javadoc
    - Right click Dependencies -> Download JavaDoc / sources

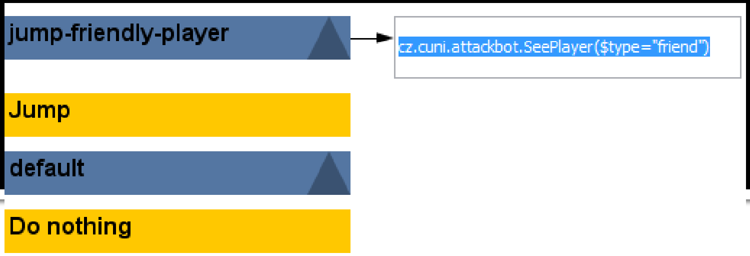




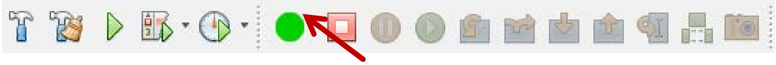


Recapitulation IV

yaPOSH



- You should know how to use yaPOSH…
  - What is **yaPOSH**
  - What is **competence** and **action pattern** and how to use it
  - How to nest **competences**
  - How to **create** new senses and actions
  - How to **parameterize** senses and actions
  - How to add actions, senses, competences & a.p. to POSH plan
  - How to **debug** yaPOSH effectively



  - What are the action stages in yaPOSH and when are they called (init(), run() & done())
  - How to prevent accidental yaPOSH cycling in infinite loop (ActionResult.FINISHED mechanics)

Recapitulation V

Online sources

- You should know where to look when you don’t know something: http://pogamut.cuni.cz
  - Pogamut JavaDoc (Documentation)
    - http://pogamut.cuni.cz/pogamut_files/latest/doc/javadoc/
  - Latest (devel) JavaDoc (at Maven Sites)
    - http://diana.ms.mff.cuni.cz/maven-sites/pogamut/
  - Pogamut tutorials (Documentation)
    - http://pogamut.cuni.cz/pogamut_files/latest/doc/tutorials/
  - Pogamut example archetypes
    - http://diana.ms.mff.cuni.cz:8081/artifactory/libs-release-local/archetype-catalog.xml
  - Pogamut forums (Forum)
    - http://diana.ms.mff.cuni.cz/main/tiki-forums.php

Today's menu

1. Recapitulation
   - What has been explained
2. **Checklist for pretest / exam**
   - **What you should have learned**
3. What's next?

Checklist I

I know how to…

- Startup UT2004 dedicated server for Deathmatch & CTF
- Startup UT2004 GUI
- Switch between spectator/player mode inside UT2004
- Run bot project within NetBeans
- Observe bot within the environment
- Stop running bot project from NetBeans

Checklist II

I know how to…

- Get nearest (euclidian / shortest-path-length) and/or visible **`NavPoint`**, **`Item`**, **`Player`**
- Get bot current location and rotation
- What distance of 100 UT units is (~ 1 meter)
- Get concrete **`NavPoint`**, **`Item`**, **`Player`** by ID
- Get information about your bot current weapons
- Set and use weapon preferences for your bot – **`weaponPrefs`**

Checklist III

I know how to…

- Navigate bot to some **`NavPoint`**
- Navigate bot to some **`Item`**
  - How to setup **`ItemPickedUp`** listener
  - How to move a bit in case that navigation was not 100% perfect
- Navigate bot to some **`Player`**
  - ⇒ Use either **`navigation`** or **`nmNav`** and **`fwMap`**
  - ⇒ Setup listener for path events, **STUCK** in particular
- Send text message to global chat
- Start shooting some **`Player`** with weapon preferences
- Get information about CTF game state in **`ctf`** module

Checklist IV

I know how to…

- Start multiple bots at once from **`main()`** method
- Get collection of all **`NavPoints`** in the map
- Get collection of all **`Players`** in the map
- Get collection of all spawned **`Items`** in the map
- Check whether the **`Item`** or **`Player`** is visible
- Use **`MyCollections`**
- Use **`DistanceUtils`**
- Use **`TabooSet`** (initialize, tabooize, filter items)
- Use **`Cooldown`** and **`Heatup`** classes
- Setup listeners, know what is the difference between **`@EventListener`** and **`@ObjectClassEventListener`** and when to use them

Checklist V

I know how to…

- Locate and open yaPOSH plan in within project explorer inside NetBeans (other sources …)
- Edit yaPOSH plan (add, delete, move actions/senses/competences) using GUI
  - Beware of lingering APs/Cs in case of "deleting/stop using" some previously defined APs/Cs
- Check how yaPOSH plan looks in text
- Create new actions and senses
- "Utilize" action's **`init()`**, **`run()`** and **`done()`** methods
- Add variable into your **`Context`** and access it from senses and actions
- Set up parameters of senses and actions and set the parameters in yaPOSH GUI
- Debug the yaPOSH plan inside NetBeans

Checklist VI

I know how to…

- Use **`tcClient`** and send messages to my team and concrete bots by their ID
- Make sure **`tcClient`** is working
  - Single TeamComm server must be running and be connected to the GameBots2004 dedicated server
- Create custom TC message classes
  - Via extending **`TCMessageData`**
- Listen to messages sent via TeamComm
  - **`@EventListener(eventClass=MyCustomMsg.class)`**

Checklist VII

I know how to…

- Use logging to your advantage
  - Too much logging may stuck the bot! (NetBeans bug…)
- Use other means of logging the bot state (text messages, changing of bot name)
- Read Pogamut log and exceptions
- Start project in Debug mode
- Place breakpoints in your code and in the yaPOSH plan

Today’s menu

1. Recapitulation
   - What has been explained
2. Checklist for pretest / exam
   - What you should have learned
3. **What’s next?**
   - **The Exam**

What's next?

Exam

- Exam date already in SIS
  - 3.6.2016, 9:00-17:00
- You will be coding a behavior for a team of bots
  - 3-4 hours of work unless you will have to experiment with Pogamut platform because you have not done your homeworks yet…
- You may come any-time between 9:00-13:00; if you come later, you might not be able to finish an exam's task
  - 17:00 is hard deadline for delivering the behavior!

Concluding remarks

- Write us your ideas, remarks, suggestions about Pogamut modules, API, the whole course syllabus, etc.

- Did you like Pogamut? You can contact us about bachelor and/or master thesis based on Pogamut (or 3D virtual worlds in general).
  - Visit http://bit.ly/amisthesis

The End

That’s all folks, see you on exam!