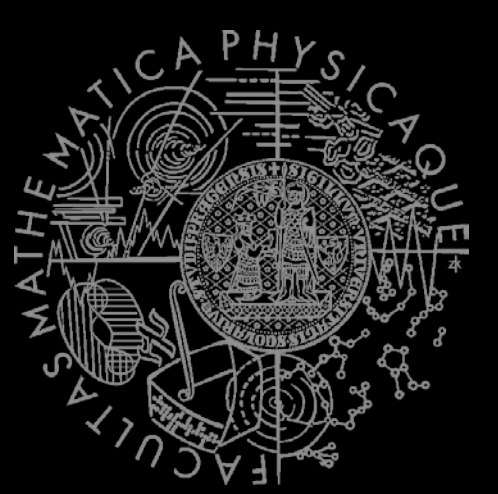
University of West Bohemia, Pilsen
23th March 2015



UT2004 bots made easy!

Pogamut 3

Tricks, AStar, Visbility
CTF & Communication,
Tournaments

Practice Lesson
Outline

Pogamut 3 Platform

1. **Review of the last practice**
2. Tricks
3. Visibility + AStar
4. Capture the Flag (CTF) & TeamComm
5. Tournaments

Pogamut World Abstraction

Basics

Objects (IWorldObject):

- Player
- Item
- NavPoint
- Self
- IncomingProjectile

Events (IWorldEvent):

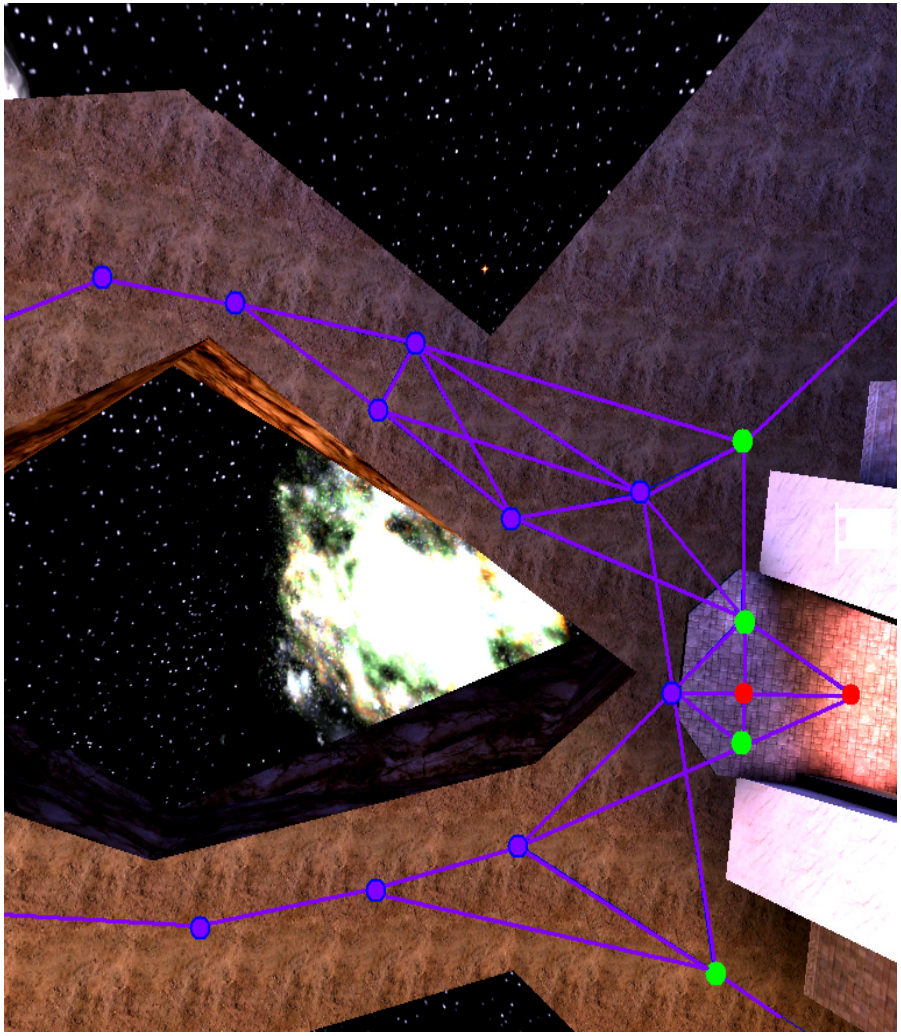
- HearNoise & HearPickup
- BotDamaged & BotKilled
- PlayerDamaged & PlayerKilled,
- Bumped
- GlobalChat

- Use modules, listeners and Pogamut helper classes!
  - this.players, this.items, this.info …
  - MyCollections, DistanceUtils

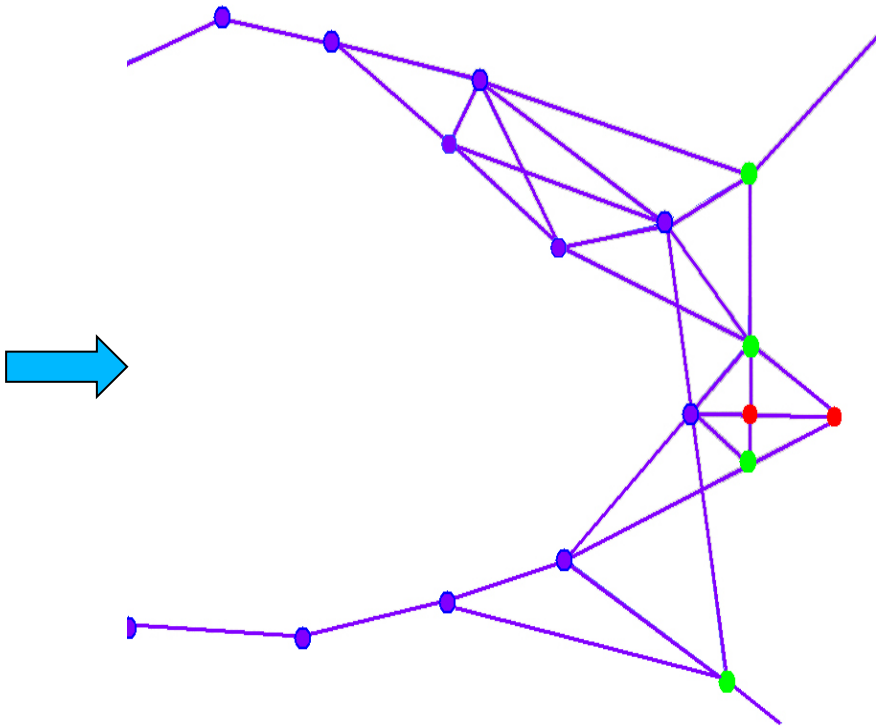
```
if (this.players.canSeePlayers()) {  …  }

@EventListener(eventClass = GlobalChat.class)
public void chat(GlobalChat chatEvent) {
    …
}
```

UT2004 World Abstraction

Navigation graph



#Navpoints in the map = 100 – 5000



Navigation

Stages

1. Decide where to go (Decision making!)
   - `items.getSpawnedItems(ItemType)`
   - `DistanceUtils.getNearest(…)`
   - `MyCollections.getRandom(…)`
   - `fwMap.getNearest(…)`
2. Plan and follow the path
   - UT2004Navigation (this.navigation)

Navigation

UT2004Navigation

- Complete navigation wrapper
  - **`UT2004Navigation`**`(…, UT2004PathExecutor, FloydWarshallMap, …) (this.navigation)`
  - Handles both path planning & path following
  - Can be called repeatedly
- Main methods
  - `navigation.navigate(…)`
  - `navigation.isNavigating()`
- Uses
  - `FloydWarshallMap (this.fwMap)`
  - `StuckDetectors`
  - `UT2004PathExecutor`

Navigation
FloydWarshallMap

- Pogamut path planner using **Floyd Warshall** algorithm ($O(n^3)$!)
  - Used by UT2004Navigation
  - Access by `this.fwMap`
- Methods of interest
  - `fwMap.getNearest…(…)`
    - Works the same as in DistanceUtils, except the distance is measured by the path length
    - Its ok to "spam" it (e.g. checking all items in each step), the nowadays computers can handle it

Practice Lesson
Outline

Pogamut 3 Platform

1. Review of the last practice
2. **Tricks**
3. AStar
4. Visibility
5. Capture the Flag (CTF) & TeamComm
6. Tournaments

Items, Weapons, Shooting
Shooting

- Shooting with `WeaponPrefs` is easy!

```
Player enemy =
    players.getNearestVisiblePlayer();

shoot.shoot(weaponPrefs, enemy);

shoot.shoot(weaponPrefs, enemy,
    ItemType.ROCKET_LAUNCHER);
// do not use rocket launcher

shoot.shoot(weaponPrefs, enemy);

shoot.setChangeWeaponCooldown(millis);
```

Items, Weapons, Shooting

Time your shooting – Cooldown class

- Sometimes you need to perform the behavior "once in a time" => `Cooldown`

```
Cooldown rocketCD = new Cooldown(2000);
                                // millis

if (rocketCD.isCool()) {
   rocketCD.use();
   shoot.shoot(weaponPrefs, enemy);
} else {
   shoot.shoot(weaponPrefs, enemy,
   ItemType.ROCKET_LAUNCHER);
}
```

Items, Weapons, Shooting

Time your behaviors – Heatup class

- Sometimes you need to pursue some behavior for a while => `Heatup`

```
Heatup pursueEnemy = new Heatup(3000);
                                  // millis

if (players.canSeeEnemy()) {
   pursueEnemy.heat();
   // fight the enemy
} else
if (pursueEnemy.isHot()) {
   // pursue the enemy
} else {
   // collect items
}
```

Items, Weapons, Shooting

Dodging Rockets

- Class `IncomingProjectile`
  - `IWorldObject!`

```
@ObjectClassEventListener(objectClass = IncomingProjectile.class,
                          eventClass = WorldObjectAppearedEvent.class)
public void projectileAppeared(WorldObjectAppearedEvent<IncomingProjectile> event) {
    IncomingProjectile projectile = event.getObject();
    log.info("PROJECTILE APPEARED: " + projectile);
    if ("XWeapons.RocketProj".equals(event.getObject().getType())) {
        log.info("ROCKET HAS BEEN SPOTTED!");
    }
}

@ObjectClassEventListener(objectClass = IncomingProjectile.class,
                          eventClass = WorldObjectUpdatedEvent.class)
public void projectileUpdated(WorldObjectUpdatedEvent<IncomingProjectile> event) {
    IncomingProjectile projectile = event.getObject();
    log.info("PROJECTILE UPDATED: " + projectile);
}

@ObjectClassEventListener(objectClass = IncomingProjectile.class,
                          eventClass = WorldObjectDisappearedEvent.class)
public void projectileDisappeared(WorldObjectDisappearedEvent<IncomingProjectile> event) {
    IncomingProjectile projectile = event.getObject();
    log.info("PROJECTILE DISAPPEARED: " + projectile);
}
```

Assignment 1

- Let's create **DodgeBot**
  - Create new EmptyBot project
  - First obtain `ItemType.ROCKET_LAUNCHER`
  - Then try to shoot down your enemy switching between rocket launcher and other weapons using `Cooldown`
  - Try to dodge enemy rockets using `move.dodgeLeft()` and `move.dodgeRight()`
  - Listen to `IncomingProjectile` event

```
@ObjectClassEventListener(
   objectClass = IncomingProjectile.class,
   eventClass = WorldObjectUpdatedEvent.class)
public void projectileUpdated(
   WorldObjectUpdatedEvent<IncomingProjectile> event
) { … }
```

Practice Lesson
Outline

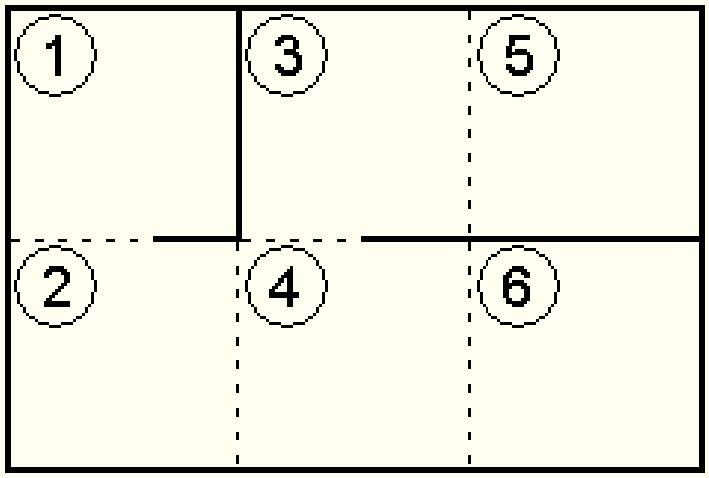
Pogamut 3 Platform

1. Review of the last practice
2. Tricks
3. **Visibility + AStar + NavMesh + Map**
4. Capture the Flag (CTF) & TeamComm
5. Tournaments

Visibility Abstraction

Visibility Matrix

- `Visibility` class
  - Contains precomputed visibility matrix between path points and some points on links
  - Matrices for competition maps already present

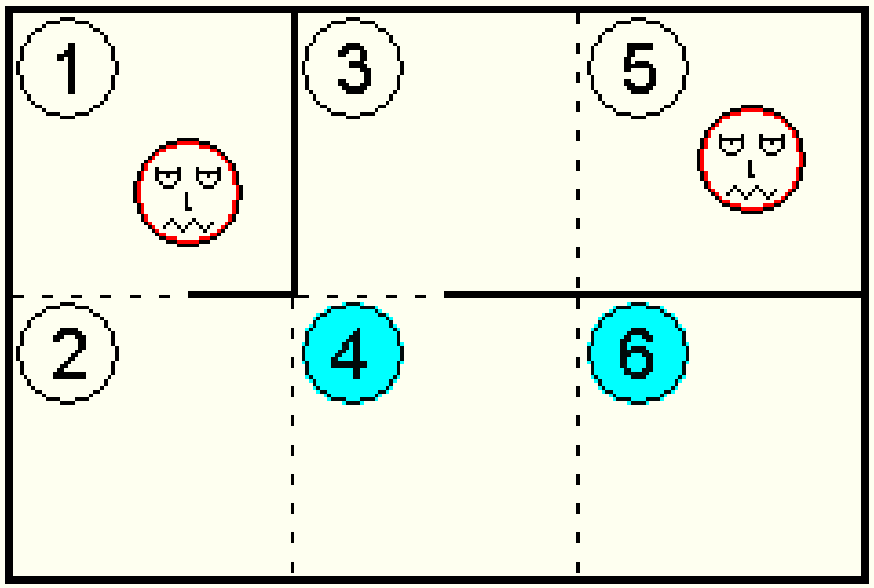


	1	2	3	4	5	6
V_1	1	1	0	0	0	0
V_2	1	1	0	1	0	1
V_3	0	0	1	0	1	0
V_4	0	1	0	1	0	1
V_5	0	0	1	0	1	0
V_6	0	1	0	1	0	1

Visibility Matrix

How to get to cover?

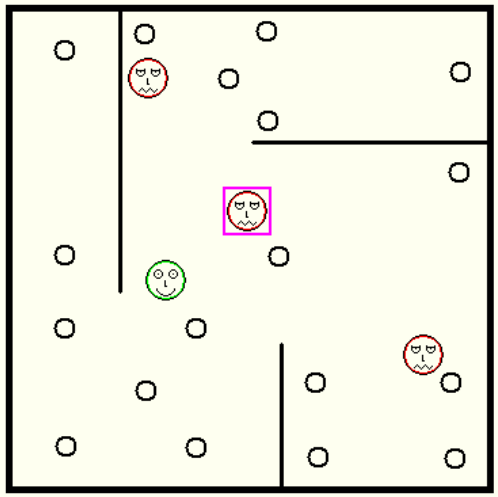
- How to find the cover?
  - Enemies … $E_1..E_k$
  - Safe waypoints … $S = \neg \bigvee_{i=1}^{k} V_{E_i}$



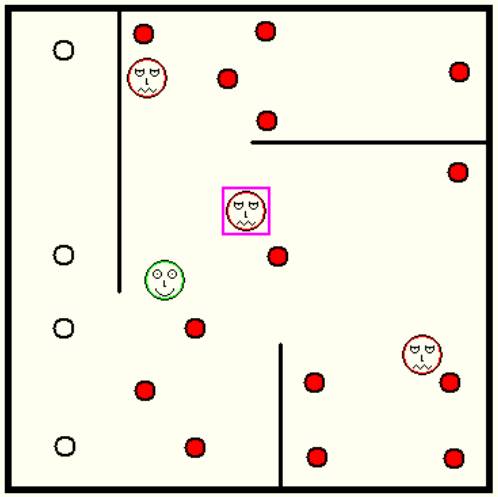
	1	2	3	4	5	6
V_1	1	1	0	0	0	0
V_2	1	1	0	1	0	1
V_3	0	0	1	0	1	0
V_4	0	1	0	1	0	1
V_5	0	0	1	0	1	0
V_6	0	1	0	1	0	1

Visibility Matrix

Smart attack

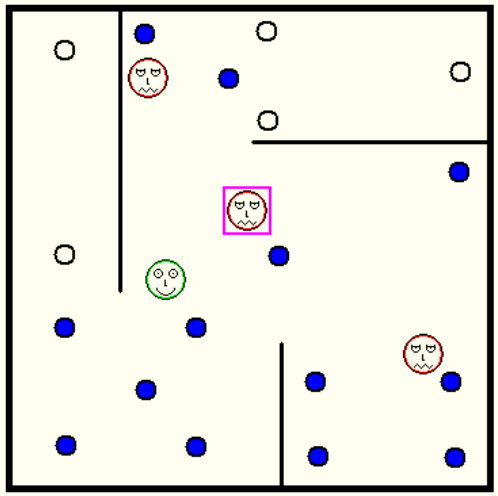


1. Choose target T
2. Others are enemies E_i



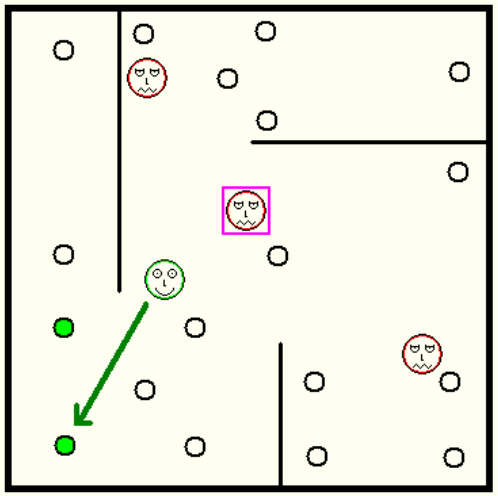
3. Navpoints other enemies E_i can see

$$\bigvee_{i=1}^{k} V_{E_i}$$



2. Navpoints target T is visible from

$$V_T$$



4. Smart place to shoot from

$$V_t \wedge \neg \bigvee_{i=1}^{k} V_{E_i}$$

Visibility Matrix

Interesting methods

- `Visibility` class

  `getNearestVisibilityLocationTo(ILocated)`

  `getCoverPointsFrom(ILocated)`

  `getCoverPointsFromN(ILocated…)`

  `getMatrix()`

- `VisibilityMatrix` class

  `getMatrix()`

  `getNearestIndex(ILocated located)`

Visibility Matrix

Visibility matrix file

- To be able to use the visibility matrix, you need to have a file with the visibility information
- Each map has its own file. E.g.

```
VisibilityMatrix-DM-TrainingDay-all.bin
```

- Place this file in the root of the project folder of your bot
- Get all matrices from svn

svn://artemis.ms.mff.cuni.cz/pogamut/trunk/project/Main/PogamutUT2004Examples/19-VisibilityBatchCreator/visibility-matrices

Navigation

UT2004AStar

- When Floyd Warshall is not enough…
- **UT2004AStar** (access by this.aStar)
  - this.aStar.findPath(from, to, IPFMapView);
- Implement your own custom **IPFMapView:**

```
new IPFMapView<NavPoint>() {

        public Collection<NavPoint> getExtraNeighbors(NavPoint node,
    Collection<NavPoint> mapNeighbors) {}

        public int getNodeExtraCost(NavPoint node, int mapCost) {}

        public int getArcExtraCost(NavPoint nodeFrom, NavPoint nodeTo, int mapCost) {}

        public boolean isNodeOpened(NavPoint node) {}

        public boolean isArcOpened(NavPoint nodeFrom, NavPoint nodeTo) {}
}
```

Level Geometry
Raycasting made Simple

Module: `this.levelGeometry`

Method: `raycast(Location from, Location to)`

Requires static information about the map. Pogamut expects the files to be present at "`${PROJECT_DIR}/map`"

Level geometry files are available inside ZIP at:

- svn://artemis.ms.mff.cuni.cz/pogamut/trunk/project/Main/Addons/ UT2004NavMeshTools\02-UT2004LevelGeom-Preparing_data_for_Recast\output\

NavMesh Raycasting
2D raycast

Module: `this.navMeshModule().getNavMesh()`

Method:
```
public double getDistanceFromEdge(Location
    location, Vector2d vector, double rayLength)
```

Requires static information about the navmesh. Pogamut expects the files to be present at "`${PROJECT_DIR}/navmesh`"

Navmeshes are available inside ZIP at:

- svn://artemis.ms.mff.cuni.cz/pogamut/trunk/project/Main/Addons/ UT2004NavMeshTools\04-NavMeshes\output\

Assignment 2

- Let's create **RunCoverBot**
  - Download RunAwayBot template from here
    - http://tinyurl.com/omm9kwl
    - Permanent link
      - http://diana.ms.mff.cuni.cz/pogamut_files/pilsen/2015/Pilsen-L03-Bots.zip
      - Start DM-1on1-Albatross
  - Implement bot, that will try to run around weapons out of sight of the player
  - Use A*, IPFMapView and VisibilityMatrix to find suitable way
    - Penalize nodes that is visible by the player!

Practice Lesson
Outline

Pogamut 3 Platform

1. Review of the last practice
2. Tricks
3. Visibility + AStar
4. **Capture the Flag (CTF) & TeamComm**
5. Tournaments

Capture the Flag (CTF)
Rules

- Players/bots are divided into two teams (red and blue).
- Each team has a flag in his base.
- The goal of the team is to capture the flag of the opposite team and bring it to their home base.
- When managed, the team scores 1 point.
  - Team can only bring opposite flag home and score a point, if the team flag is in team home base!
- If the flag is dropped it will be returned to home base after some time.

Pogamut CTF support
Bases & game status

- **CTF module**
  - this.ctf
- **Where are the bases?**
  - this.ctf.getOurBase();
  - this.ctf.getEnemyBase();
- **Whats the game status?**
  - this.ctf.canOurTeamScore();
  - this.ctf.canEnemyTeamScore();
- **Am I winning?**
  - game.getTeamScores();
  - info.getTeamScore();

Pogamut CTF support II
Flags

- **I want my flag!**
  - Flag is represented by **FlagInfo** object.
  - this.ctf.getOurFlag();
  - this.ctf.getEnemyFlag();
- **Is someone messing with my flag?**
  - this.ctf.isOurFlagHome();
  - this.ctf.isOurFlagHeld();
- **How about enemy flag?**
  - this.ctf.isEnemyFlagHome();
  - this.ctf.isEnemyFlagHeld();

Pogamut CTF support III
Team communication

- Use **SendMessage** command.
  - this.act.act(new SendMessage().setTeamIndex(info.getTeam()).setText("Help"));
- Listen to team message with **TeamChat** event.

```
@EventListener(eventClass = TeamChat.class)
public void teamChat(TeamChat event) {
        …
}
```

- OLD AND SLOW!!!

Pogamut CTF support IV
UT2004TeamComm 1/3

- Allows you to send Serializable Java objects between members of the team
- Allows you to specify “subchannels”
- You can easily listens to your messages as if they were `IWorldEvent` objects

Pogamut CTF support IV
UT2004TeamComm 2/3


```
@EventListener(eventClass=TCMessage.class)
public void allMsg(TCMessage tcMessage) {
    log.info("@EventListener(TCMessage)”);
}

@EventListener(eventClass=TCHello.class)
public void hello(TCHello hello) {
    log.info("@EventListener(TCHello)”);
}
```

Pogamut CTF support IV
UT2004TeamComm 3/3


```
public class TeamCommBot extends
    UT2004BotTCController<UT2004Bot>

<dependencies>
    <dependency>
      <groupId>cz.cuni.amis.pogamut.ut2004</groupId>
      <artifactId>ut2004-team-comm</artifactId>
      <version>3.7.0</version>
    </dependency>
</dependencies>
```

Example available at:

- svn://artemis.ms.mff.cuni.cz/pogamut/trunk/project/Main/PogamutUT2004Examples/26-TeamCommBot

Assignment 3

- Extend TC controller
- Try to implement bots that are broadcasting own location and tries to run to location being heard

Assignment 4

- Alter **CTFBot**
  - Arm yourself before going into action!
  - Try to get enemy flag!
  - Try to get your flag, if it is stolen!

Practice Lesson
Outline

Pogamut 3 Platform

1. Review of the last practice
2. Tricks
3. Visibility + AStar
4. Capture the Flag (CTF) & TeamComm
5. **Tournaments**

Assignment 5

- Check out example project with CTF Match
- svn://artemis.ms.mff.cuni.cz/pogamut/trunk/project/Main/PogamutUT2004Examples/31-CTFMatch

- And check it out!