

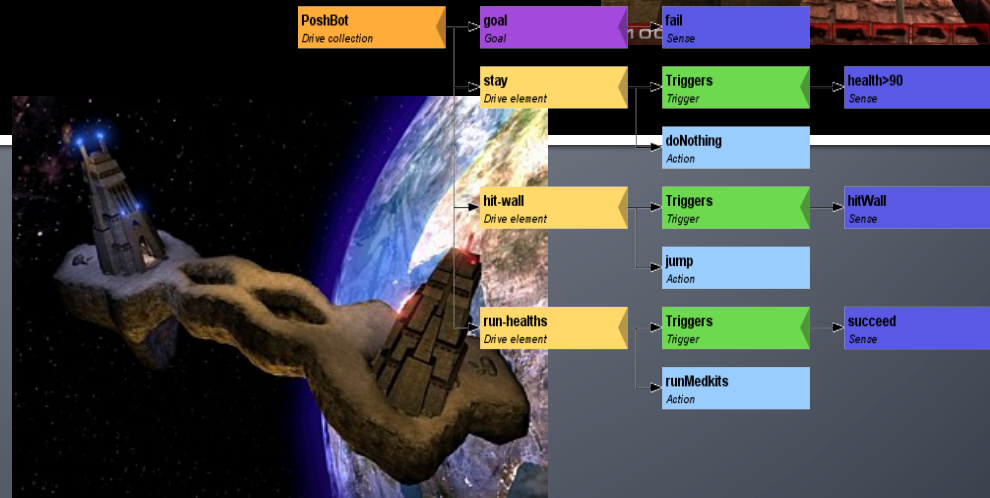
Faculty of Mathematics and Physics
Charles University in Prague
2nd May 2013



UT2004 bots made easy!

Pogamut 3

Lecture 10 – CTF



Warm Up!



- Fill the short test for this lesson
 - 6 minutes limit

Today's menu



1. **Navigation Tips**
2. Capture the Flag (CTF)

Navigation Tips

Useful classes



- **MyCollections**

```
Item targetItem = MyCollections.getRandomFiltered(  
    items.getSpawnedItems(ItemType.Category.WEAPON), myIFilter  
);
```

- **fwMap**

```
fwMap.getNearestFilteredItem(...)
```

- **DistanceUtils**

- **Heatup**

- **Cooldown**

Navigation Tips

Anti-stuck I



- **Always handle STUCK event!**
 - Use TabooSets to temporarily filter items/navpoints you were stuck going to
 - This prevents your bot to cycle in 1) pick the same item, 2) run towards it, 3) stuck

```
TabooSet<Item> tabooItems = new TabooSet<Item>(bot);
this.navigation.addStrongNavigationListener(

    new FlagListener<NavigationState>() {
        @Override
        public void flagChanged(NavigationState changedValue){
            switch (changedValue) {
                case STUCK:
                case PATH_COMPUTATION_FAILED:
                    tabooItems.add(targetItem, 30);
                break;
            }
        }
    }
);
```

Navigation Tips

Anti-stuck II



- Then don't forget to use your taboo set to filter the items you are picking from!

```
if (!this.navigation.isNavigating()) {  
    targetItem = MyCollections.getRandom(  
        tabooItems.filter(items.getSpawnedItems().values())  
    );  
    this.navigation.navigate(targetItem);  
}
```

Navigation Tips

yaPOSH



- In yaPOSH, a good place to register navigation state listeners, TabooSets and weapon preferences is in the constructor of bot context class, e.g.:

```
public AttackBotContext(UT2004Bot bot) {
    super("AttackBotContext", bot);
    // IMPORTANT: Various modules of context must be initialized.
    initialize();

    // INITIALIZE CUSTOM MODULES
    getWeaponPrefs()...;
    ...
}
```

Navigation Tips

aStar – plugging the path to navigation



1. Get starting point

```
NavPoint from =  
    navigation.getNearestNavPoint(  
        info.getLocation()  
    );
```
2. Get target point

```
NavPoint to =  
    MyCollections.getRandom(  
        navPoints.getNavPoints().values()  
    );
```
3. Find the path

```
List<NavPoint> path =  
    aStar.findPath(from, to).getPath();
```
4. Wrap the path

```
PrecomputedPathFuture pf =  
    new PrecomputedPathFuture(from, to, path);
```
5. Execute it

```
this.navigation.navigate(pf);
```


Today's menu



1. Navigation Tips
2. **Capture the Flag (CTF)**

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.

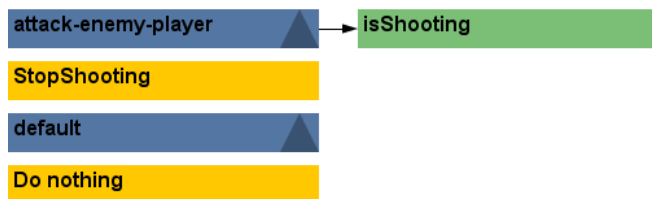
```
@EventListener(eventClass = TeamChat.class)  
public void teamChat(TeamChat event) {  
    ...  
}
```

yaPOSH

ActionResult.FINISHED WARNING !



- Action returning in `run()` method **FINISHED** tells yaPOSH to re-evaluate plan immediately to search for a new action
- This can be used to your advantage (parallel actions), but has a caveat!
- Consider plan, where `StopShooting` returns **FINISHED** in `run()` immediately:



- Makes sense, because as we send stop shoot command in `run()`, the action is done...
- The problem is that yaPOSH re-evaluates the plan immediately to search for a new action and guess what it finds? `StopShooting` again. Why?
 - Because `isShooting` sense will be returning the same value it was before! The environmental state is not changed. yaPOSH re-evaluates immediately! We are stuck in infinite loop and no more environmental updates will ever come (even at first glance no exceptions raised).
- For these types of actions always return **RUNNING_ONCE** !

Assignment

(or Homework)



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

Assignment

Cheatsheet



- **Locations of interest**
 - `this.ctf.getOurBase();`
 - `this.ctf.getEnemyBase();`
- **Useful info about the game (could be senses)**
 - `this.ctf.canOurTeamScore();`
 - `this.ctf.canEnemyTeamScore();`
 - `this.ctf.isEnemyFlagHome();`
 - `this.ctf.isOurFlagHeld();`
 - `this.ctf.isBotCarryingEnemyFlag();`
 - Our bot
- Flag is represented by FlagInfo object
 - `this.ctf.getOurFlag();`
 - Can check `isVisible()`...

Questions?

I sense a soul in search of answers...



- We do not own the patent of perfection (yet...)
- In case of doubts about the assignment, tournament or hard problems, bugs don't hesitate to contact us!
 - Jakub Gemrot (Monday practice lessons)
 - jakub.gemrot@gmail.com
 - Michal Bída (Thursday practice lessons)
 - michal.bida@gmail.com