

Faculty of Mathematics and Physics  
Charles University in Prague  
19<sup>th</sup> April 2016



UT2004 bots made easy!

# Pogamut 3

Lecture 8 – BOD & DeathMatch



# Warm Up!



- Fill the short test for this workshop
  - 8 minutes limit
  - <https://goo.gl/rP71DQ>
  - [https://docs.google.com/forms/d/1V\\_4N2HcBRrrDTGliJbRCsTI3JXOlinO1oKOlo2PVMtM/viewform](https://docs.google.com/forms/d/1V_4N2HcBRrrDTGliJbRCsTI3JXOlinO1oKOlo2PVMtM/viewform)

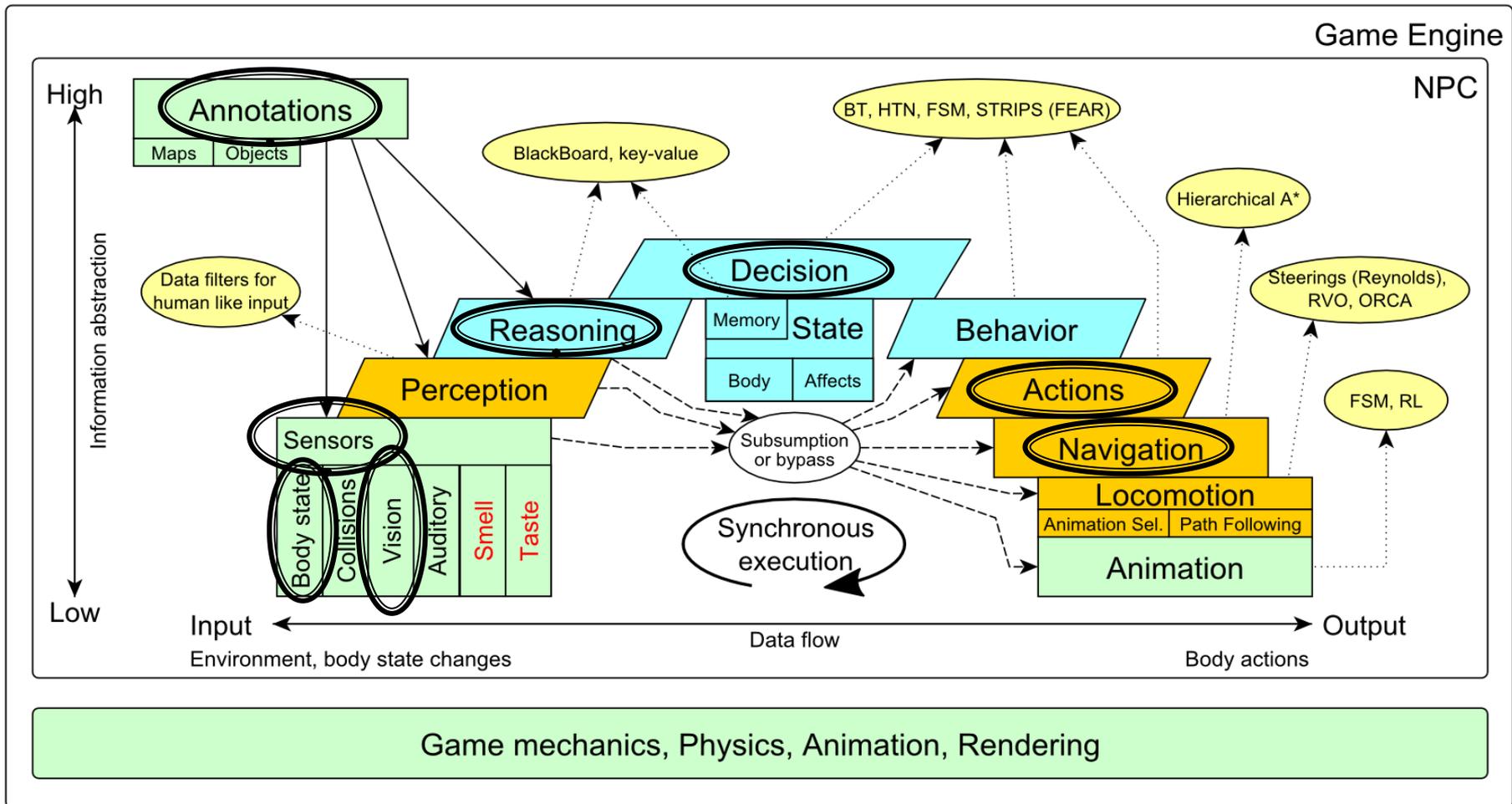
# Today's menu



1. **Big Picture**
2. Weapons & Shooting
3. DM & BOD

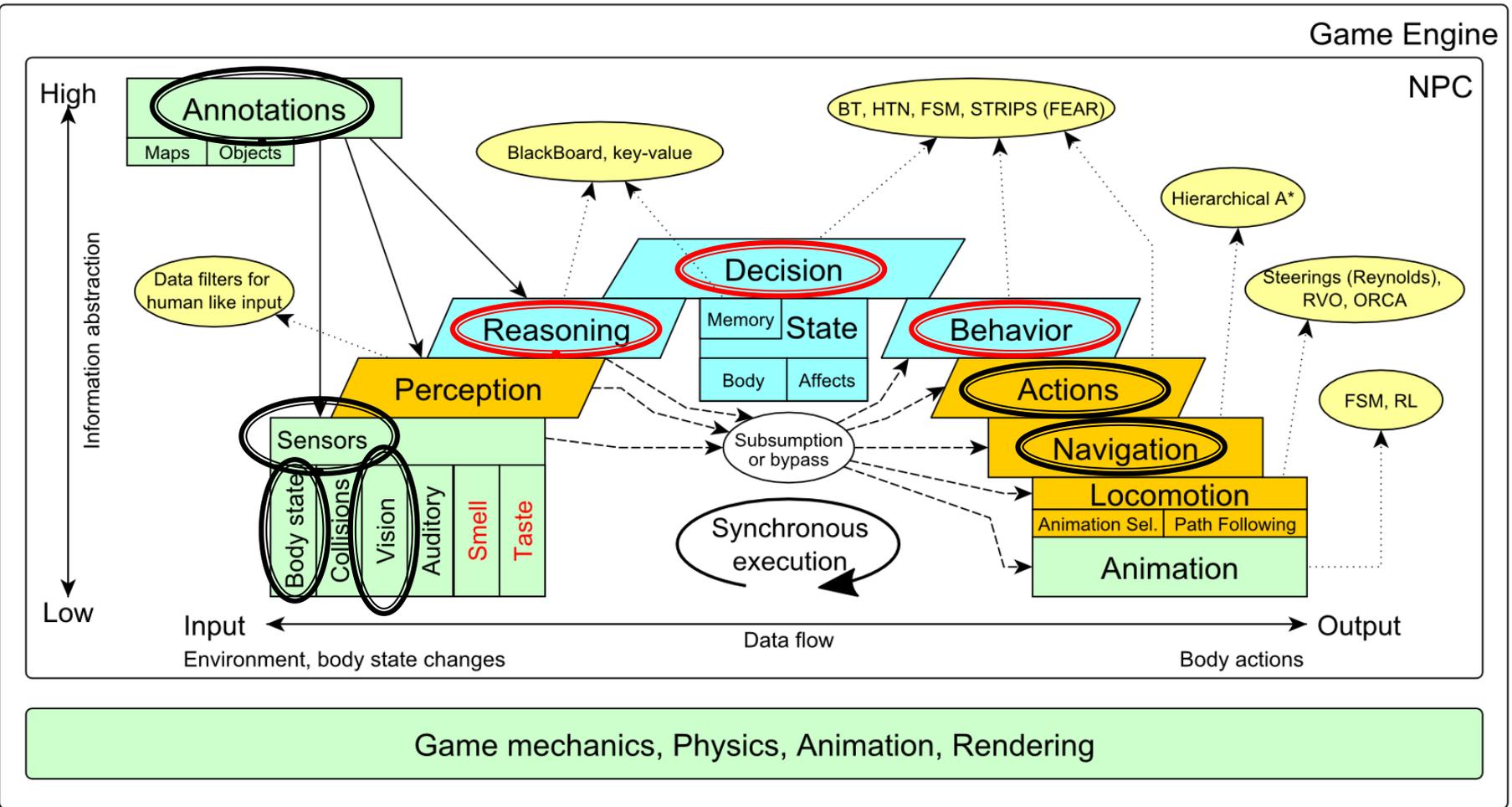
# Big Picture

## Already covered



# Big Picture

## Today



# Today's menu



1. Big Picture
2. **Weapons & Shooting**
  - <http://planetunreal.gamespy.com/View.php?view=UT2004GameInfo.Detail&id=26>
  - <http://unreal.standardof.net/unreal-tournament-2004/weapons-and-tactics/>
3. DM & BOD

# Weapons

## UT2004 weapons guide I – the weak



- **UT2004ItemType.SHIELD\_GUN** (DEFAULT)
  - Melee weapon (can be charged)
  - Secondary mode – shield (handy from distance!)
- **UT2004ItemType.ASSAULT\_RIFLE** (DEFAULT)
  - Weak, basic, inaccurate (can have two)
  - Secondary mode – grenades (charged), don't use
- **UT2004ItemType.BIO\_RIFLE**
  - Fires green blobs, short range, defense weapon
  - Secondary mode – charged (big blob)
  - Avoid ... weak
- **UT2004ItemType.LINK\_GUN**
  - Primary fires rather slow, but decent projectiles
  - Secondary – medium-to-short range beam
    - **Max 1150 distance!**



# Weapons

## UT2004 weapons guide II – the strong



- **UT2004ItemType . FLAK\_CANNON**
  - Shotgun style weapon – deadly at short range
  - Sec. mode is a grenade launcher, don't use
- **UT2004ItemType . MINIGUN**
  - Choose between rapid fire but less accuracy (pri. mode) or slower fire and more accuracy (sec. mode)
- **UT2004ItemType . SHOCK\_RIFLE**
  - Pri. mode is very accurate with medium damage
  - Sec. mode fires slow moving projectiles, that can be detonated by pri. fire making a big explosion (tricky to do though)
- **UT2004ItemType . LIGHTING\_GUN** & **UT2004ItemType . SNIPER\_RIFLE**
  - Sniper rifle – precise, can one-shot others by a headshot
  - Bots can use only pri. fire (sec. is zoom)



# Weapons

## UT2004 weapons guide III – mayhem



- **UT2004ItemType.ROCKET\_LAUNCHER**

- Good old rocket launcher, rockets have splash damage (beware!)
- Secondary mode can charge up to three rockets



- **UT2004ItemType.REDEEMER**

- Unleash nuclear mayhem!
  - big splash damage radius
- Bots can use only primary firing mode!



- **UT2004ItemType.U\_DAMAGE\_PACK**

- Not enough damage? Grab DOUBLE DAMAGE pack and double your damage output!



# Weapons

## Weaponry class



- `this.weaponry`
  - All you wanted to know about UT2004 weapons but were afraid to ask
  - Note that it contains also some obsolete and to-be-deprecated methods...

`weaponry.getCurrentWeapon()`

`weaponry.hasWeapon(UT2004ItemType)`

`weaponry.hasLoadedWeapon()`

`weaponry.hasPrimaryLoadedWeapon()`

`weaponry.hasSecondaryLoadedWeapon()`

`weaponry.getLoadedWeapons()`

`weaponry.changeWeapon()`

...

# Weapons & Shooting

## WeaponPreferences



- Weapons' effectiveness depends on distance to target
- Thus you should create different priority list for various "ranges"
- Wrapped in class **weaponPrefs**

```
weaponPrefs.addGeneralPref(UT2004ItemType.MINIGUN, true);  
weaponPrefs.addGeneralPref(UT2004ItemType.LINK_GUN, false);
```

- **true** -> primary firing mode
- **false** -> secondary firing mode

```
weaponPrefs.newPrefsRange(CLOSE_COMBAT_RANGE = 400)  
    .add(UT2004ItemType.FLAK_CANNON, true)  
    .add(UT2004ItemType.LINK_GUN, true); // 0-to-CLOSE  
weaponPrefs.newPrefsRange(MEDIUM_COMBAT_RANGE = 1150)  
    .add(UT2004ItemType.MINIGUN, true)  
    .add(UT2004ItemType.ROCKET_LAUNCHER, true); // CLOSE-to-MEDIUM
```

- If **range** prefs fails, **general** are used
- You have to experiment! (*== behavior parametrization!*)

# Weapons & Shooting

## Shooting



- Shooting with `WeaponPrefs` is easy!

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

```
shoot.shoot(weaponPrefs, enemy);
```

```
shoot.shoot(weaponPrefs, enemy,  
            UT2004ItemType.ROCKET_LAUNCHER);
```

```
// do not use rocket launcher
```

```
shoot.setChangeWeaponCooldown(millis);
```

# 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,  
    UT2004ItemType.ROCKET_LAUNCHER);  
}
```

# 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  
}
```

# Today's menu



1. Big Picture
2. Weapons & Shooting
3. **DM & BOD**

# Deathmatch Bot

## BOD Revisited

### Behavior Oriented Design

*by Joanna J. Bryson (UK)*

<http://www.cs.bath.ac.uk/~jjb/web/bod.html>

1. Specify top-level decision
  - a) Name the behaviors that the bot should do
  - b) Identify the list of sensors that is required to perform the behavior
  - c) Identify the priorities of behaviors
  - d) Identify behavior switching conditions
2. Recursion on respective behaviors until primitive actions reached

# Deathmatch Bot

## BOD Revisited

1. `if ( seeEnemy ) combatBehavior()`
  2. `collectItems()`
- Simple approach
    1. We either fight the enemy
    2. Or we're collecting some items when not in contact
  - Problems?
    - What if enemy is covering / reappearing?
    - What if I don't want to fight the enemy?



# Deathmatch Bot

## BOD Revisited

1. `if ( seeEnemy & wantToFight )`  
    `combatBehavior( )`
2. `if ( hunting )`    `interceptEnemy( )`
3. `if ( seeEnemy )`    `coverBehavior( )`
4. `collectItems( )`

### ■ Problems?

- What if enemy is reappearing and we're switching between 3 + 4?

# Deathmatch Bot

## BOD Revisited

1. `if ( seeEnemy & wantToFight )`  
`combatBehavior( )`
2. `if ( hunting )` `interceptEnemy( )`
3. `if ( seeEnemy )` `coverBehavior( )`
4. `if ( seenEnemy )` `maintainFocus( ) +`  
`coverFire( )`
5. `collectItems( )`

### ■ Problems?

- We're breaking If-Then rules here! 4 must be executed in parallel with 5!
- Where is "hunting" flag raised & dropped?

# Deathmatch Bot

## On the Nature of Actions



- Every bot has several action “categories” that can be executed in parallel
- In case of UT2004
  - There are only two (three) of them
    1. Movement
    2. Shooting
    3. (Focus)
  - Each category is having multiple actions that interfere with each others / cancel themselves out
  - And there are multiple behaviors that make use of them and they are going to overlap

# Deathmatch Bot

## On the Nature of Actions



- Behaviors
  1. Combat / Intercept (possible decomposition)
    - Movement      Covering, Approaching, Projectile dodging, Aiming, Item picking
    - Shooting      Weapon selection and fire rates
    - Focus          Enemy
  2. Item picking
    - Movement      Navigation, Item prioritization
    - Focus          On path
  
- Practically, the bot is doing both of them at once! (It might just not have the other part specified.)
  
- => Be careful with switches and correctly maintain shooting/navigation/focus

# Deathmatch Bot

## On the Nature of Actions



### ■ Behaviors

#### 2. Item picking

- Movement Navigation, Item prioritization
- Focus On path

```
collectItems(boolean canUseFocus,  
             IFilter<Item> itemFilter)
```

- ⇒ Sort of “subsumption architecture” approach
- ⇒ Provides better control over the behavior
- ⇒ Better reusability (e.g. from `coverBehavior`)

# Assignment 8

(or Homework)



- Create **DeathMatchBot**
  - That arms himself and is able to fight an opponent
    - **Combat, Intercept, Cover, Focus, Item Collecting behaviors**
  - Try to come up with methods that can be reused between combat / item collection

# DM Bot Tournament

## Announcement!



- All your **DeathMatchBots** will automatically take part in DM Bot Tournament
  - 1<sub>vs</sub>1, 10 frags, 10 minutes max
  - DM-10n1-Roughinery-FPS map
- Deadline for submission is **1.5.2016 8:00**

# Send us finished assignment



Via e-mail:

- *Subject*
  - "Pogamut homework 2016 – Assignment X"
    - Replace 'X' with the assignment number and the subject has to be without quotes of course
    - ...or face **-2 score penalization**
- *To*
  - [jakub.gemrot@gmail.com](mailto:jakub.gemrot@gmail.com)
    - Jakub Gemrot (Tuesday practice lessons)
- *Attachment*
  - Completely zip-up your project(s) folder except 'target' directory and IDE specific files (or face **-2 score penalization**)
- *Body*
  - **Please send us information about how much time it took you to finish the assignment + any comments regarding your implementation struggle**
    - *Information won't be abused/made public*
    - *In fact it helps to make the practice lessons better*
  - Don't forget to mention your full name!

# Questions?

I sense a soul in search of answers...



- In case of doubts about the assignment, tournament or hard problems, bugs don't hesitate to contact us!
  - Jakub Gemrot (Tuesday practice lessons)
    - [jakub.gemrot@gmail.com](mailto:jakub.gemrot@gmail.com)