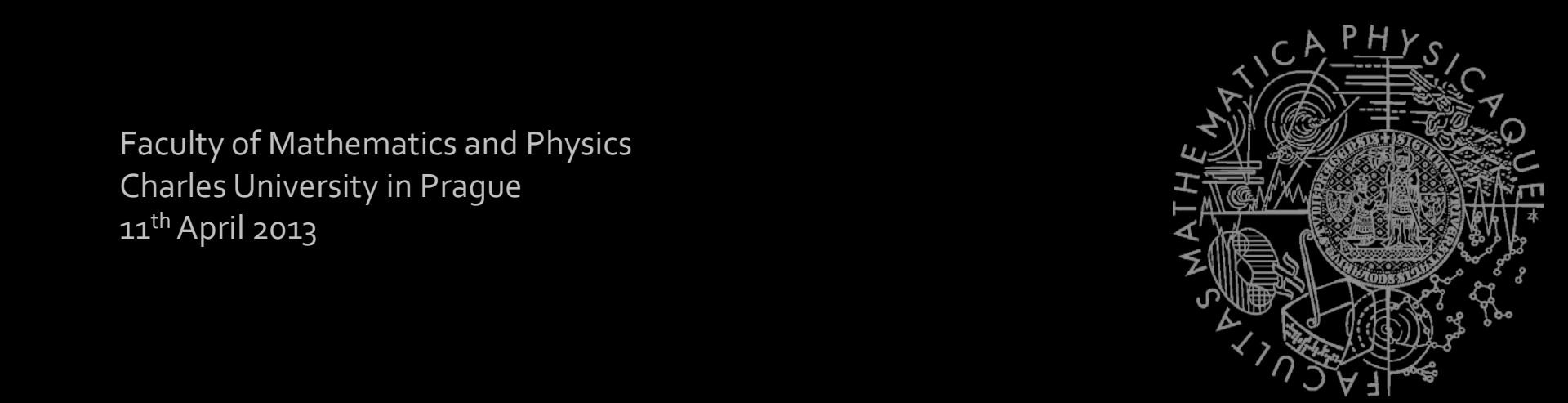
Faculty of Mathematics and Physics
Charles University in Prague
11th April 2013



UT2004 bots made easy!

Pogamut 3

Lecture 7 – Items and Weapons

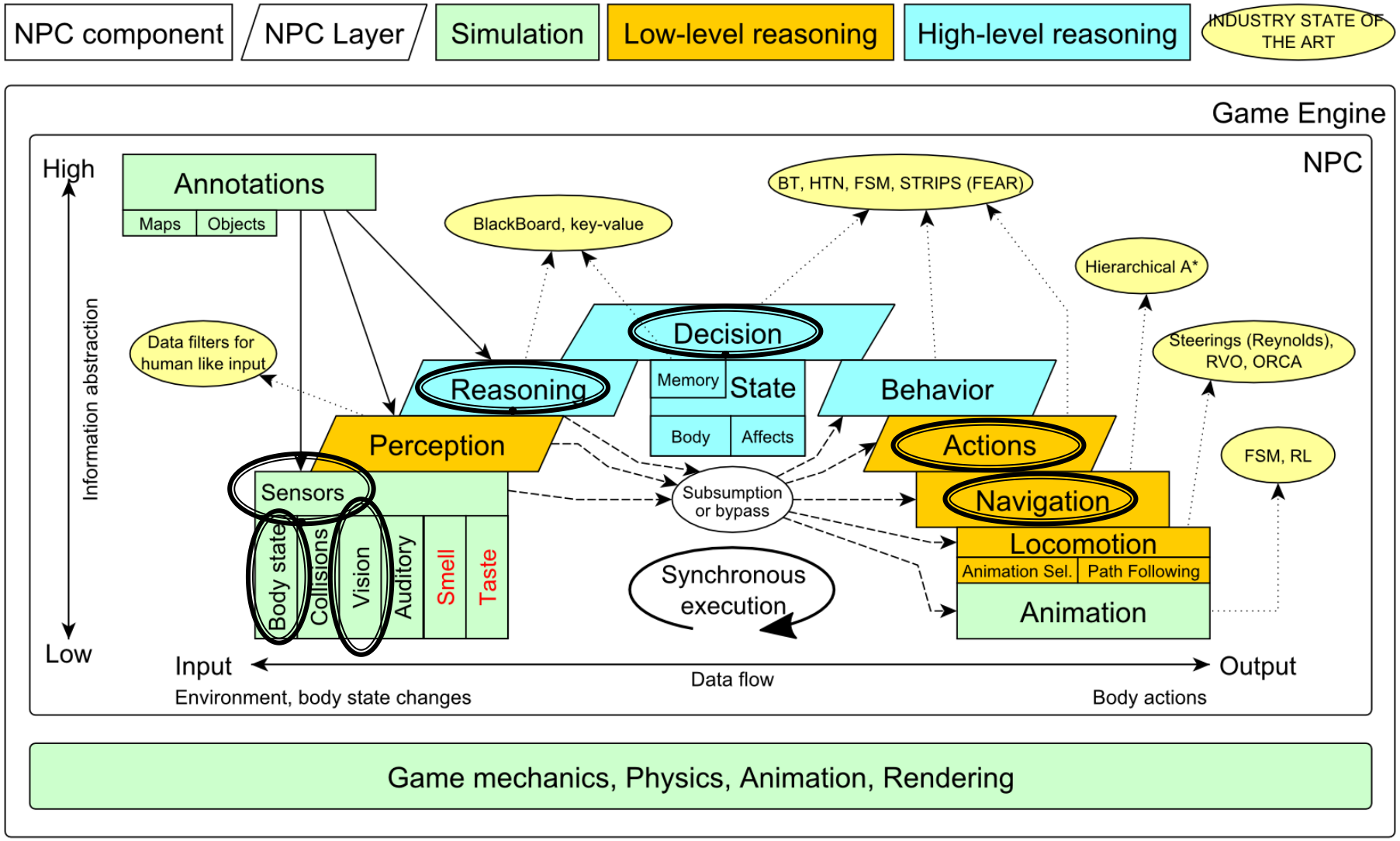
Warm Up!

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

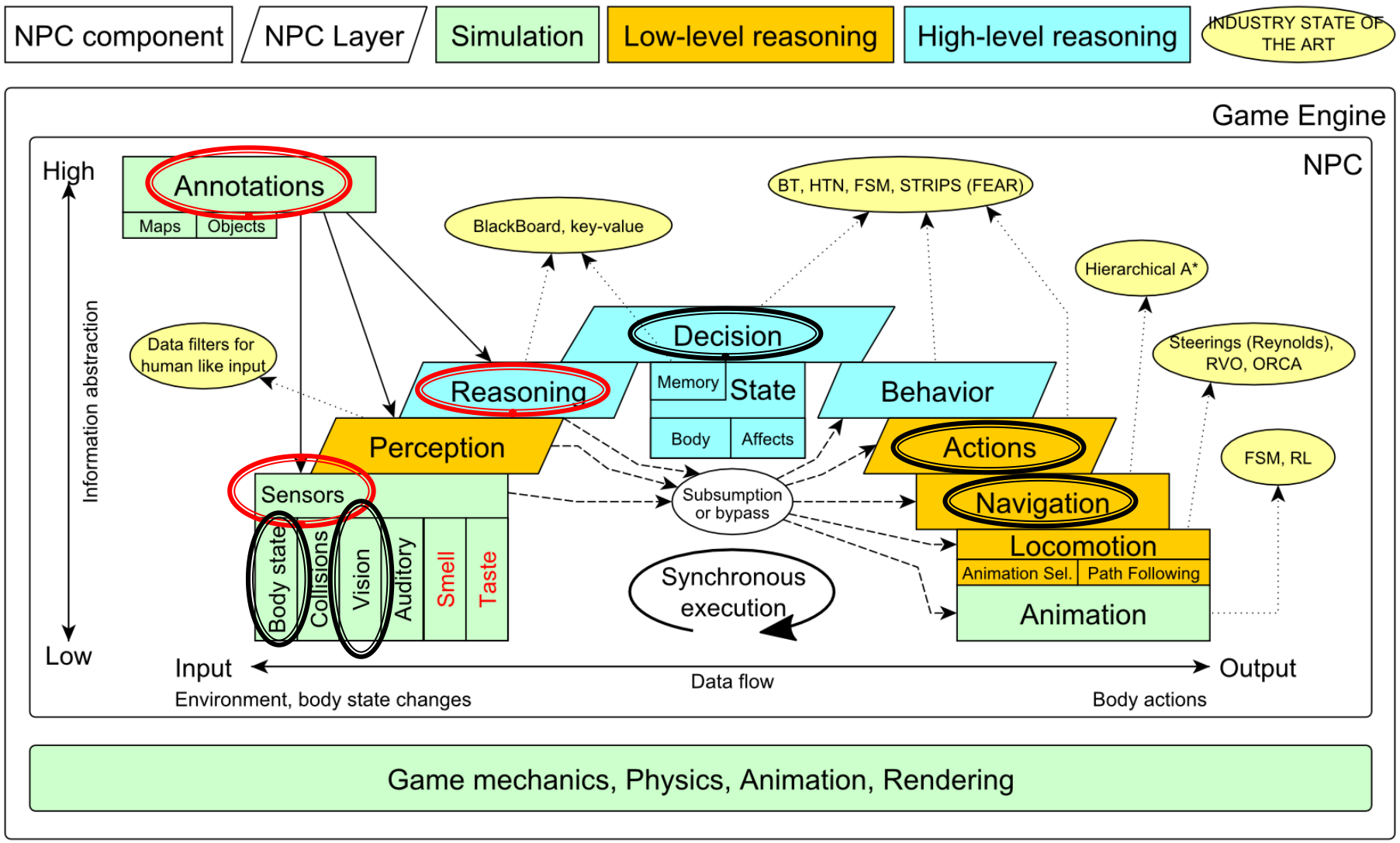
Today's menu

1. **Big Picture**
2. Pogamut World Abstraction
3. Navigation intermezzo
4. Items
5. Weapons introduction

Big Picture
Already covered



Big Picture
Today



Today's menu

1. Big Picture
2. **Pogamut World Abstraction**
3. Navigation intermezzo
4. Items
5. Weapons & Shooting

Pogamut World Abstraction

Items overview

Objects (IWorldObject):

- Player
- **Item**
- NavPoint
- Self
- IncomingProjectile

Events (IWorldEvent):

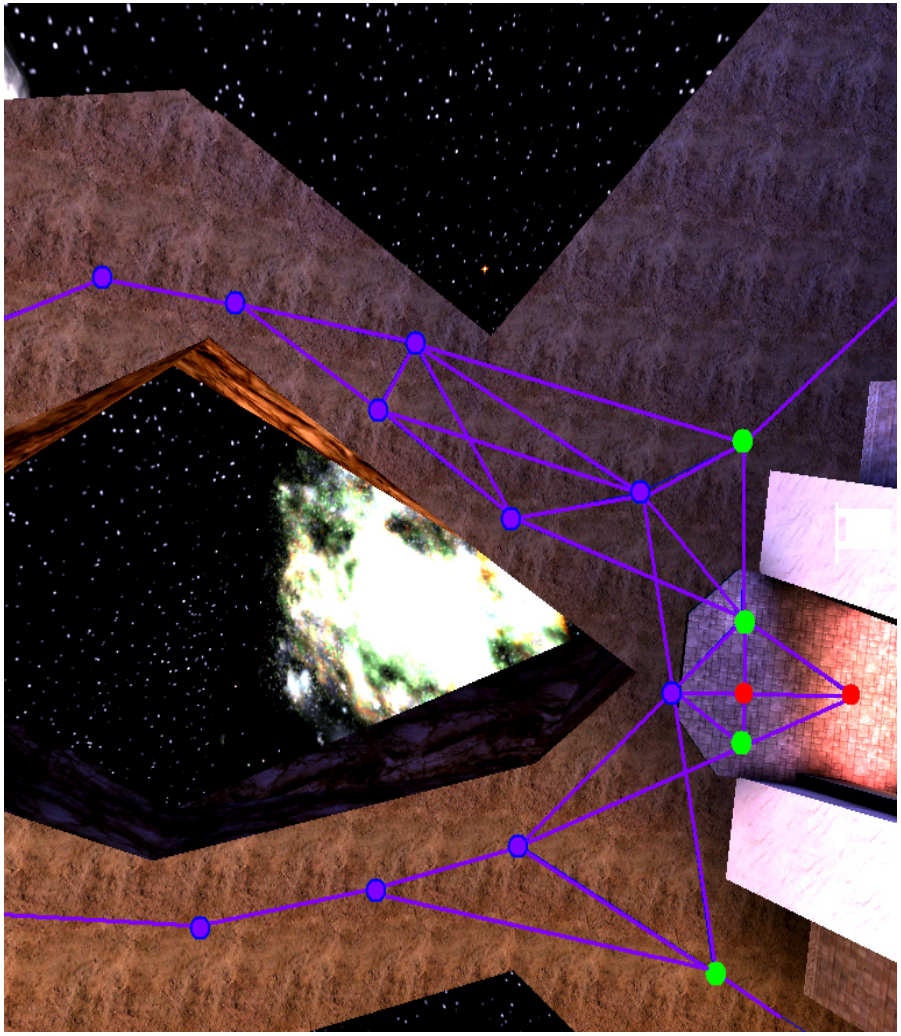
- HearNoise & HearPickup
- BotDamaged & BotKilled
- PlayerDamaged & PlayerKilled,
- **ItemPickedUp**
- GlobalChat

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

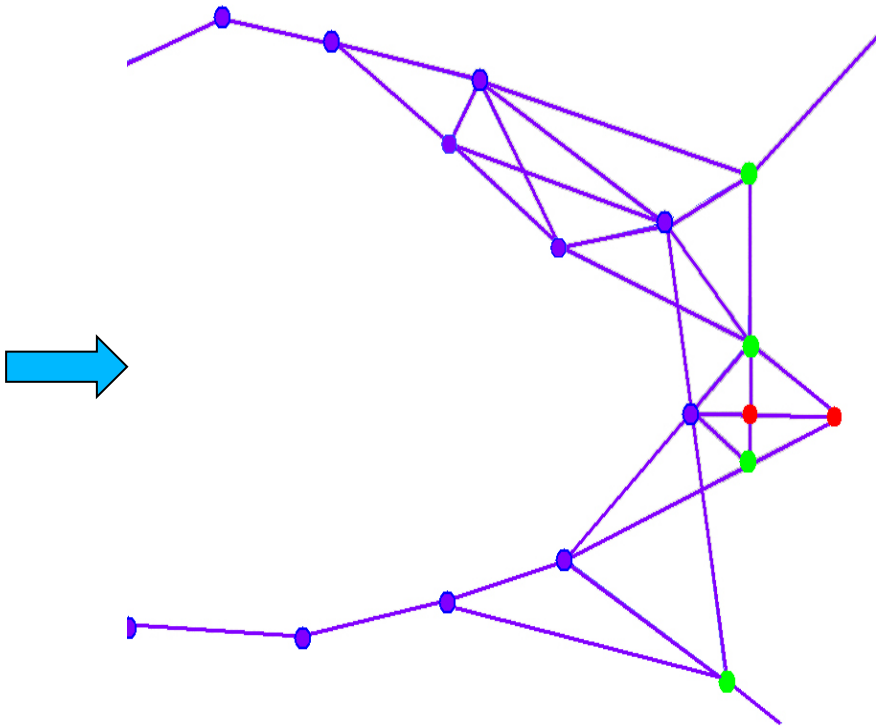
```
if (this.items. getSpawnedItems().values().size() > 0) {  …  }

@EventListener(eventClass = ItemPickedUp.class)
public void itemPickedUp(ItemPickedUp event) {
    …
}
```

UT2004 World Abstraction

Navigation graph



#Navpoints in the map = 100 – 5000



UT2004 World Abstraction

Nav link/NavPoint types

- **NavPoints**
  - **InventorySpot**
  - JumpPad
  - Lift
  - Teleport
  - Door
  - PlayerStart
  - SnipingSpot
  - …
- **Nav links**
  - Walk
  - Jump
  - Lift
  - Door
  - DoubleJump
  - …

Today's menu

1. Big Picture
2. Pogamut World Abstraction
3. **Navigation intermezzo**
4. Items
5. Weapons introduction

Navigation

Step by step

General steps:

1. Decide where to go
2. Plan the path (list of navpoints)
3. Follow the path

Navigation

Step by step

Improvements:

1. Decide where to go
2. Plan the path (list of navpoints)
3. Follow the path
   - **Watch for meaningfulness!**
4. **Check that you have truly grabbed the item!**

Navigation

Stages

1. Decide where to go (Decision making!)
   - `items.getSpawnedItems(ItemType)`
   - perform reasoning
     - It's OK to compute paths to all spawned items every `logic()`
2. Plan and follow the path
   - this.navigation.navigate(item)

Navigation

Stages

3. Follow the path
   - Do you still believe that item your running for is spawned?
     - It might have been picked up by your opponent!

```
if (!items.isPickupSpawned(item))
{ replan(); }
```

Navigation

Stages

4. Check that you truly grabbed the item!
   - UT2004 navigation is not 100% precise
     - It might stopped running just right before the item!

```
@EventListener(eventClass = ItemPickedUp.class)
public void itemPickedUp(ItemPickedUp event) {
    if (itemRunning.getId().equals(event.getId()) {
        // I have picked the item!
    }
}
```

Today's menu

1. Big Picture
2. Pogamut World Abstraction
3. Navigation intermezzo
4. **Items**
5. Weapons introduction

Items
Basics

- `Item` (accesible via **`this.items`**!)
  - More "spawning location" than item
    - `items.isPickupSpawned(item)`
  - Unique `UnrealId` => Can be used in `Set`, `Map`
  - `ILocated` ~ `getLocation()` ~ X, Y, Z
  - `IViewable` ~ `isVisible()`
  - Always has corresponding `NavPoint` instance
    - `NavPoint itemNP = item.getNavPoint()`
  - Described by `ItemType`
    - `item.getType()`

Items

Important ItemTypes


```
ItemType.FLAK_CANNON
        .MINIGUN
        .LIGHTING_GUN
        .ROCKET_LAUNCHER
        .LINK_GUN

ItemType.SUPER_HEALTH
        .SUPER_ARMOR
        .SHIELD_PACK
        .SUPER_SHIELD_PACK
        .U_DAMAGE_PACK
```

Items

ItemType & Categories

- `ItemType`
  - Enum holding concrete type of the item
  - Part of some `ItemType.Category`
    - Categories are divided based on what items are intended to do
    - `ItemType.Category.HEALTH`
    - `ItemType.Category.ARMOR`
    - `ItemType.Category.SHIELD`
    - `ItemType.Category.WEAPON`
    - `ItemType.Category.AMMO`

Items

Items

- Agent module: `items`

  ```
  items.getAllItems()
  items.getVisibleItems(ItemType)
  items.getSpawnedItems(ItemType)
  items.isPickable(Item)
  ```

- `DistanceUtils`

  ```
  .getNearest(Collection<Ilocated>)
  .getNthNearest(n,Collection<Ilocated>)
  ```

- `fwMap`

  ```
  .getNearestItem(Collection<Item>)
  ```

Items
ItemDescriptor(s)

- Every item is “well” described

```
Item item =
    items.getAll(ItemType.Category.WEAPONS).values()
    .iterator().next();
WeaponDescriptor weaponDesc =
    (WeaponDescriptor)
    descriptors.getDescriptor(item.getType());
if (weaponDesc.getPriDamage() > 50) {
…
}
```

- Ammo/Armor/HealthDescriptor available as well

Today's menu

1. Big Picture
2. Pogamut World Abstraction
3. Navigation intermezzo
4. Items
5. **Weapons introduction**
   - http://planetunreal.gamespy.com/View.php?view=UT2004GameInfo.Detail&id=26

Weapons

UT2004 weapons guide I – the weak

- **ItemType.SHIELD_GUN** (DEFAULT)
  - Melee weapon (can be charged)
  - Secondary mode – shield

- **ItemType.ASSAULT_RIFLE** (DEFAULT)
  - Weak, basic, inaccurate (can have two)
  - Secondary mode – grenades (charged)

- **ItemType.BIO_RIFLE**
  - Fires green blobs, short range, defense weapon
  - Secondary mode – charged (big blob)

- **ItemType.LINK_GUN**
  - Primary fires rather slow, but decent projectiles
  - Secondary – medium-to-short range beam

Weapons

UT2004 weapons guide II – the strong

- **ItemType.FLAK_CANNON**
  - Shotgun style weapon – deadly at short range
  - Sec. mode is a grenade launcher

- **ItemType.MINIGUN**
  - Choose between rapid fire but less accuracy (pri. mode) or slower fire and more accuracy (sec. mode)

- **ItemType.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)

- **ItemType.LIGHTING_GUN, 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

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

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

- **ItemType.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(ItemType)
weaponry.hasLoadedWeapon()
weaponry.hasPrimaryLoadedWeapon()
weaponry.hasSecondaryLoadedWeapon()
weaponry.getLoadedWeapons()
weaponry.changeWeapon()
…
```

Assignment 7
(or Homework)

- Create **CollectorBot**
  - Collects weapons, ammo and armor on the map
  - Run 3 bots on DM-1on1-Albatross
  - What if the item you want to pick up is not there? (e.g. you run two collector bots and the other one got it first) ~ **items.isPickupSpawned(item)**
    - Re-plan!
  - How to check that your bot can pick some item?
    - `items.isPickable(Item)`
  - How to check the bot successfully picked up an item?
  - How to avoid unreachable items?
    - Use `TabooSet`

Assignment
Cheatsheet

- Getting and filtering the items:
  - `this.items.getSpawnedItems(ItemType.Category.WEAPON)`
  - `MyCollections.getFiltered(Collection, new IFilter<Item>() {…})`
- Handling unreachable items:
  - `Navigation.addStrongNavigationListener(…STUCK_EVENT…)`
  - `myTabooSet.add() & myTabooSet.filter(…)`
- Some thin items (e.g. HealthVial) are tricky to pick up! How to be sure that your bot has picked the item up?
  - `ItemPickedUp.class` event

```
@EventListener(eventClass=ItemPickedUp.class)
public void pickedUp(ItemPickepUp event) {}
```

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