

Faculty of Mathematics and Physics
Charles University
8th March 2023



UT2004 bots made easy!

Pogamut 3

Lab 04 – Combat



Warm Up!



- Fill the short test for this workshop

8 minutes limit

<https://tinyurl.com/5h97r39y>

0 vs. 0, i vs. 1 vs. 1

Permalink

<https://docs.google.com/forms/d/e/1FAIpQLSfZfjrBKluD7XA7X7y4vPvNO5dQD-keM5QGFl2jiT2GAogq-w/viewform>

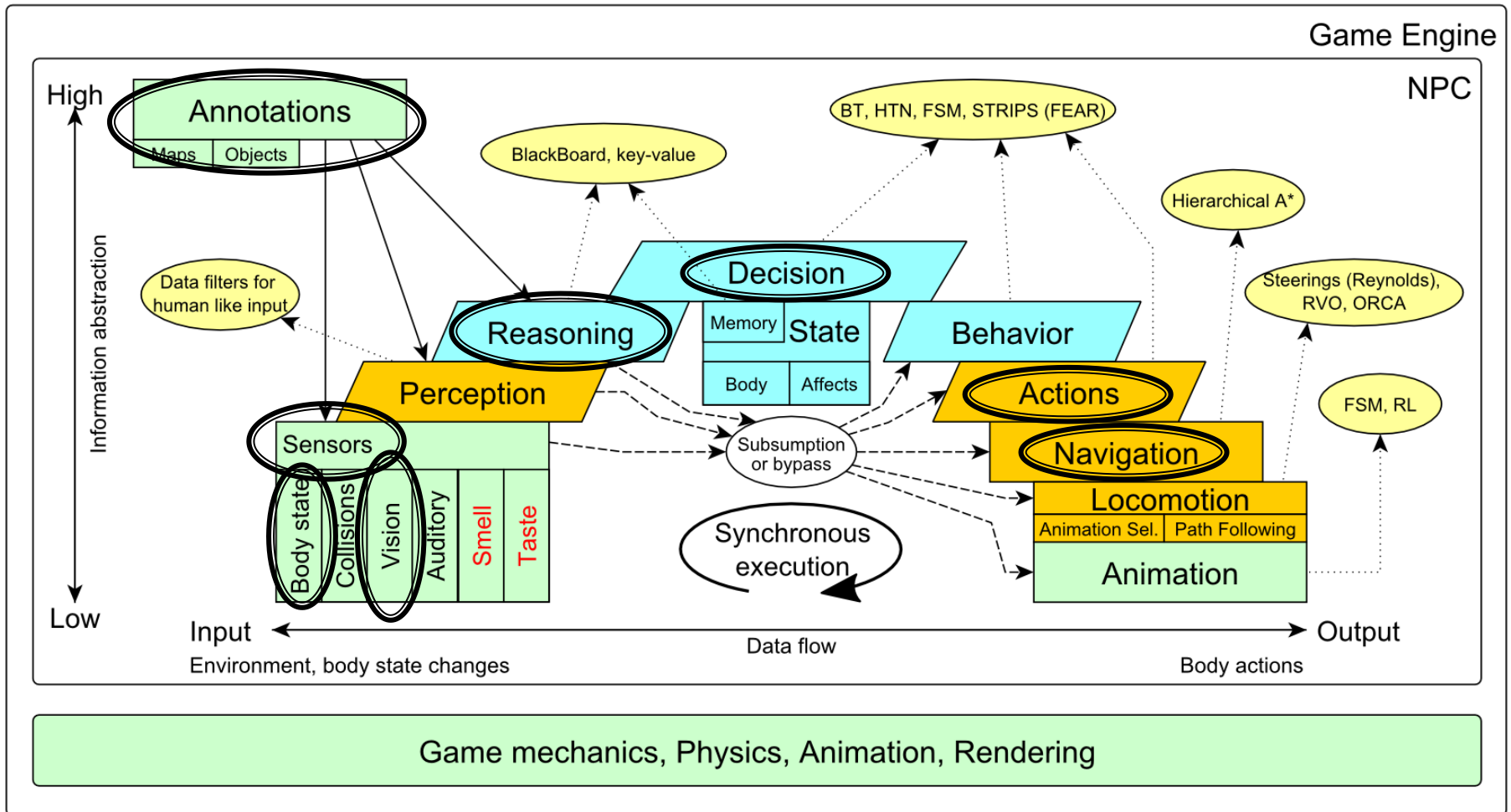
Today's menu



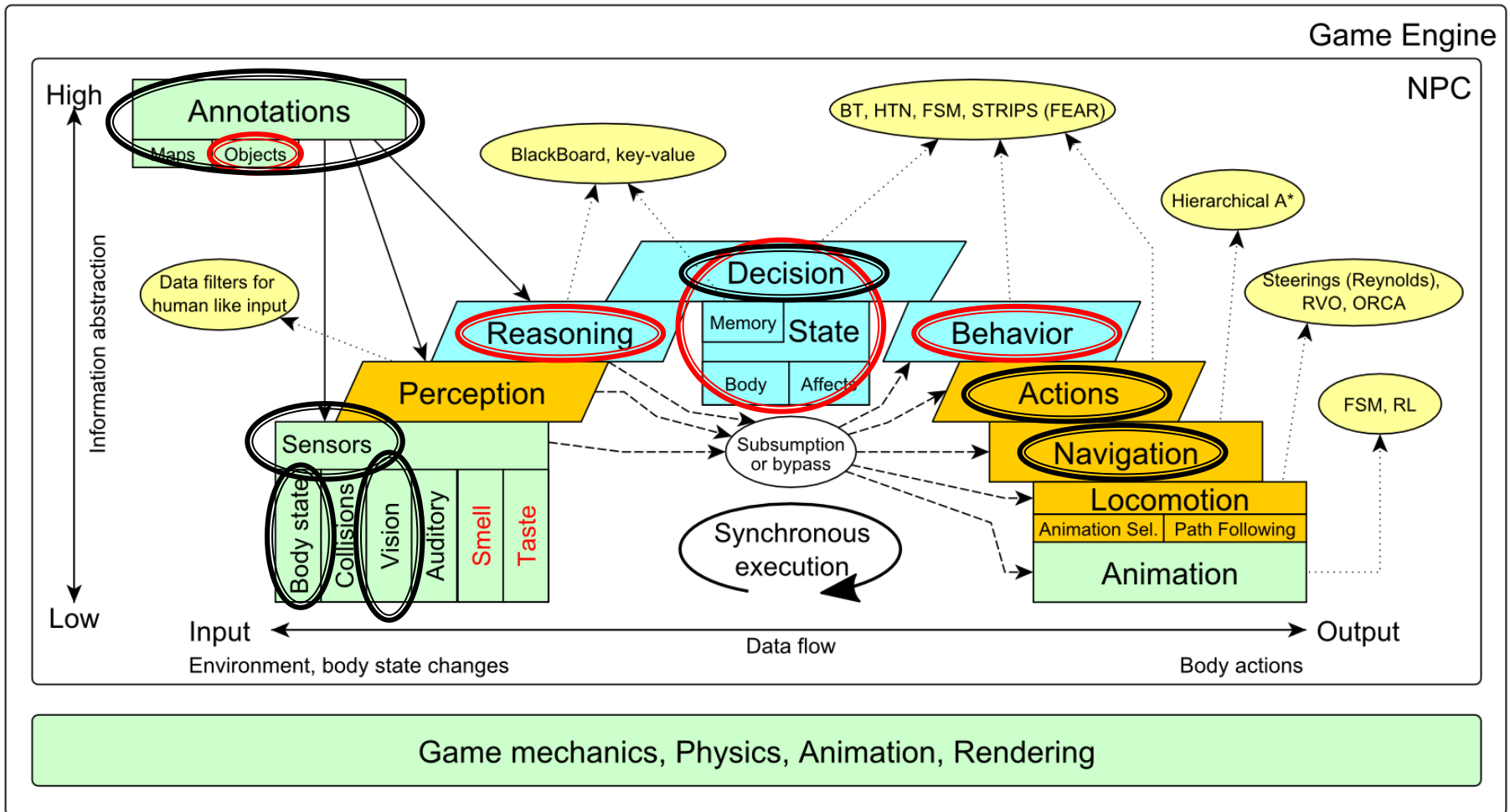
1. Big Picture
2. Homework: 1v1 Deathmatch Bot
3. Weapons & Shooting
4. Behavior Oriented Design w/ Scripting
5. Behavior Oriented Design w/ OOP

Big Picture

Already covered



Big Picture Today



Today's menu



1. Big Picture
2. **Homework: 1v1 Deathmatch Bot**
3. Weapons & Shooting
4. Behavior Oriented Design w/ Scripting
5. Behavior Oriented Design w/ OOP
6. Deathmatch Tournament Announcement

Homework 04

Duelist



- Create **1v1 DeathMatchBot**
 - That arms itself and is able to fight an opponent
 - **Base behaviors** (*required*)
 - Combat
 - Item collection
 - **Advanced behaviors** (*for advanced points*)
 - Pursue (1 points)
 - Projectile dodging (1 points)
 - Lightning shooting (1 points)
 - Medkit (2 points)
 - Up-to 5 advanced points
- Try to come up with methods that can be reused between combat / item collection
- Implementing behaviors via OOP is not required but highly advised for advanced behaviors...

Homework 04

Duelist – **BASE** Behaviors



- **Combat**
 - Use your weapons well: try to get to advantageous position
 - E.g. having a Flak Cannon, you try to close up, and vice versa
- **Item collection**
 - Reason about utility of items to pickup; priority vs. distance

Homework 04

Duelist – **ADVANCED** Behaviors



- **Pursue behavior** (1p.)
 - When you lose the sight; try to pursue your adversary
- **Projectile** (1p.)
 - Try to dodge enemy rockets
- **Lightning gun shooting** (1p.)
 - Time the use of lightning gun, blend it with other weapons
- **Medkit behavior** (2p.)
 - Try not to fight always; get a medkit if nearby instead

Homework 04

Duelist - Template



- Create **1v1 Duelist** Stub
 - <https://tinyurl.com/26yzk2mz>
- Full link
 - https://drive.google.com/file/d/1oPQMnAEew4FGqowRbZN4zFbBTmh_z5X_t/view?usp=sharing

Weapons & Shooting



What & When & How

(More gameplay info at [PlanetUnreal](#))

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!
- Will not be present in any map we will be playing



■ **UT2004ItemType.** U DAMAGE PACK

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



Weapons

Module representing the inventory



- `this.weaponry`
 - As bots has only “weapons and ammo” as items, this represents the inventory of bots
 - It contains both sensory methods for reading the state of the inventory as well as action methods
 - The most frequently used methods:

```
weaponry.getCurrentWeapon()
```

```
weaponry.getWeapons()
```

```
weaponry.hasWeapon(UT2004ItemType)
```

```
weaponry.hasPrimaryLoadedWeapon(UT2004ItemType)
```

```
weaponry.hasSecondaryLoadedWeapon(UT2004ItemType)
```

```
weaponry.changeWeapon(UT2004ItemType)
```

Weapons & Shooting

WeaponPreferences



- Weapons' effectiveness depends on distance to a target
 - ⇒ Thus you should create different priority list for various "ranges"
- The definition of weapon preferences wrt. the distance are wrapped in class **weaponPrefs**
- In **weaponPrefs**, you can specify "**ranges**" and associate them with "weapon priority lists" + you can specify "**fallback** weapon priority list" in case you have no weapon in the inventory/loaded/defined for the given target distance
 - ⇒ If **range** preferences fails, **fallback** is used
- When specifying the weapon, you also specify whether to fire with primary or with secondary mode

Weapons & Shooting

WeaponPreferences



Defining fallback priority list
(from the highest priority to the lowest)

```
weaponPrefs.addGeneralPref(UT2004ItemType.MINIGUN, true); // true == primary  
weaponPrefs.addGeneralPref(UT2004ItemType.LINK_GUN, false); // false == sec.
```

Defining ranges + associating them with the priority list
(from the highest priority to the lowest)

```
weaponPrefs.newPrefsRange(400) // close combat range: 0-400  
    .add(UT2004ItemType.FLAK_CANNON, true)  
    .add(UT2004ItemType.LINK_GUN, true);  
weaponPrefs.newPrefsRange(1150) // medium combat range: 400-1150  
    .add(UT2004ItemType.MINIGUN, true)  
    .add(UT2004ItemType.ROCKET_LAUNCHER, true);  
weaponPrefs.newPrefsRange(2500) // long combat range: 1150-2500  
    .add(UT2004ItemType.LIGHTNING_GUN, true)  
    .add(UT2004ItemType.SHOCK_RIFLE, true);
```

The best place to initialize `weaponPrefs` is within the
`botInitialized(...)` method.

Weapons & Shooting

Shooting



Shooting with **WeaponPrefs** is easy!

```
Player enemy = players.getNearestVisiblePlayer();  
  
shoot.shoot(weaponPrefs, enemy);  
  
shoot.shoot(weaponPrefs, enemy,  
             UT2004ItemType.LIGHTNING_GUN);  
            // use weapon prefs, but do not use (==ignore)  
            // the lightning gun  
  
shoot.setChangeWeaponCooldown(millis);  
            // prevent weapons over-switching  
  
shoot.stopShooting();  
            // do not forget that, otherwise, your bot might get stuck  
            // shooting indefinitely
```

Weapons & Shooting



Time your shooting – Cooldown class

Sometimes you need to perform the behavior “once in a time” => Cooldown

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

```
if (lightCD.tryUse()) {  
    shoot.shoot(weaponPrefs, enemy);  
} else {  
    shoot.shoot(weaponPrefs, enemy,  
        UT2004ItemType.LIGHTNING_GUN);  
}
```

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  
}
```

BOD w/Scripting



Behavior Oriented Design w/ Scripting

Let's build some example rules
and see where they fail...



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

Duelist

BOD with SHRP



1. `if (seeEnemy) combatBehavior()`
2. `collectItems()`

For the sake of brevity, let's assume **prioritized SHRP**, only the first firing rule gets executed.

■ 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?

Duelist

BOD with SHRP



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

■ Problems?

- `seeEnemy & !wantToFight & !hunting`
 - What should we be doing? Why is this wrong?
 - `collectItems()` does not expect contact with the enemy!

Duelist

BOD with SHRP



```
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?

Duelist

BOD with SHRP



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

■ Problems (hint: SHRP)?

- Maintain focus + cover fire does not know where to navigate next
=> We're breaking If-Then rules here!
4 must be executed in parallel with 5!

Duelist

BOD with Scripting



```
if      ( seeEnemy() & wantToFight() )
    combatBehavior();
else if ( hunting() )
    interceptEnemy();
else if ( seeEnemy() )
    coverBehavior();
else {
    if ( seenEnemy() ) {
        maintainFocus();
        coverFire();
    }
    collectItems();
}
```

Let's switch to
Scripting now.

■ Design?

- Is it wise to decouple “triggering conditions” and “behavior code”?
- Not that much – behavior code will need to use triggering conditions as asserts (defensive style!)

Duelist

BOD with Scripting



```
1. if (combatBehavior()) return;  
2. if (interceptEnemy()) return;  
3. if (coverBehavior()) return;  
4. respondToLastEnemy();  
5. collectItems();
```

■ Design!

- Better way is to integrate triggering conditions into the behavior itself!

```
boolean combatBehavior() {  
    if (!(seeEnemy & wantToFight)) return false;  
    ... combat behavior code ...  
    return true;  
}
```

First
precondition
checking.

Then action(s)
execution.

BOD w/Scripting



Behavior Oriented Design w/ Scripting

On the nature of actions & body resources

Duelist

On the Nature of Actions



- Every bot (virtual agent in general) body has several body resources that we can schedule an action for in parallel
- In case of UT2004
 - There are only two (three-four) of them
 1. Movement
 2. Shooting (+ weapon selection)
 3. (Focus)

```
navigation.setFocus(ILocated / null)
```
 - 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

Duelist

On the Nature of Actions



- Behaviors / Body resources
 - 1. Combat / Intercept (possible decomposition)
 - Movement Approach, Retreat, Projectile dodging, Aiming, Item picking
 - Shooting Weapon selection and fire rates
 - Focus Movement (lifts), Aiming
 - 2. Item picking
 - Movement Navigation / Item prioritization
 - Focus Sometimes because of lifts
 - Practically, the bot is executing all body resources in parallel; defaults slipped in if not specified (e.g. during movement, focus forward if not specified otherwise)
- => Be careful with switches and correctly maintain shooting/navigation/focus between behaviors



On the Nature of Actions

■ How to write more reusable behaviors?

2. E.g. Item picking

- Movement Navigation / Item prioritization
- Focus Movement (lifts)

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

- ⇒ Sort of “subsumption architecture” approach
- ⇒ Provides better control over the behavior
- ⇒ Better reusability

BOD w/OOP



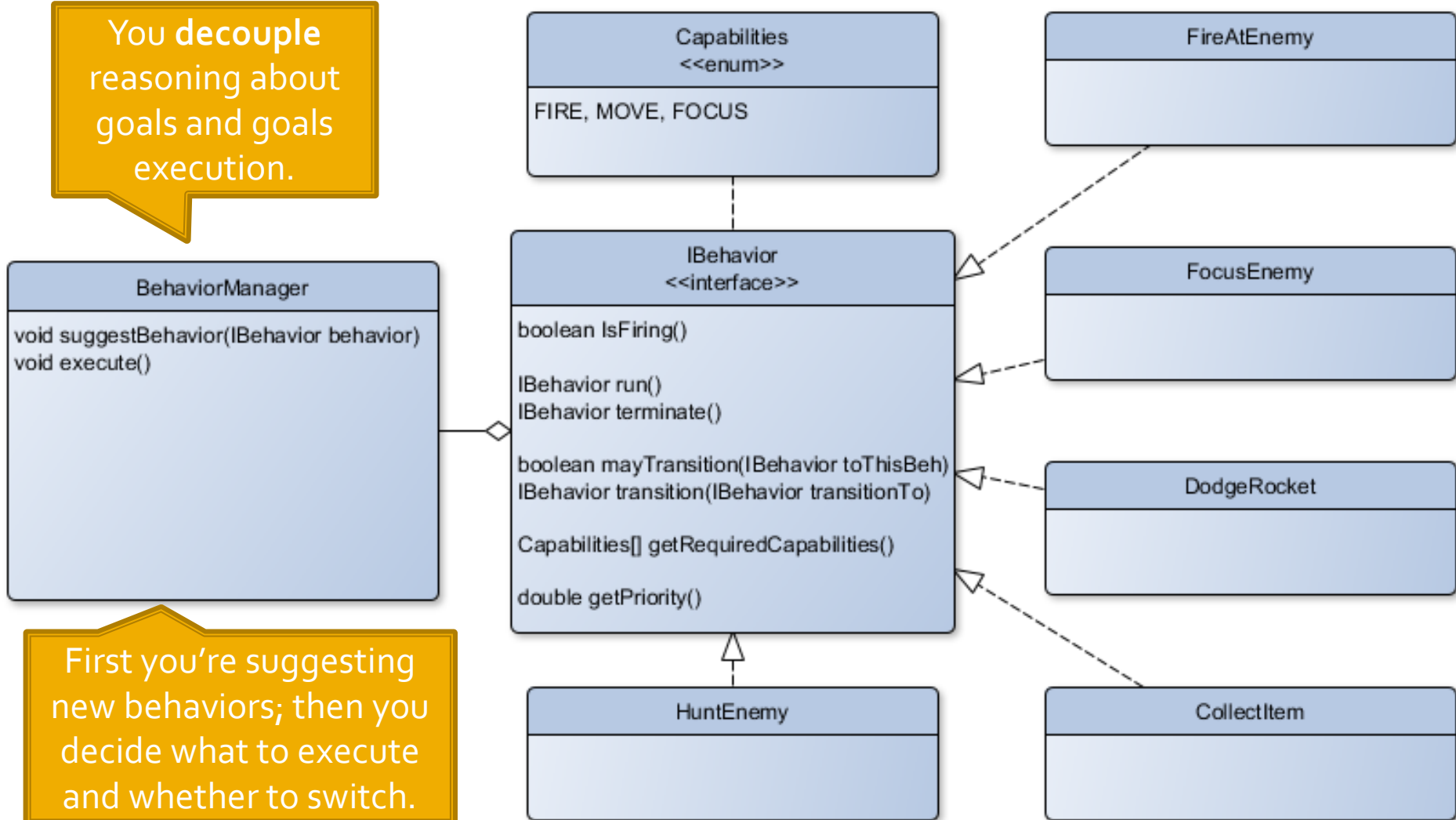
Behavior Oriented Design w/ OOP
Unleash the inner architect...

Duelist

BOD with OOP



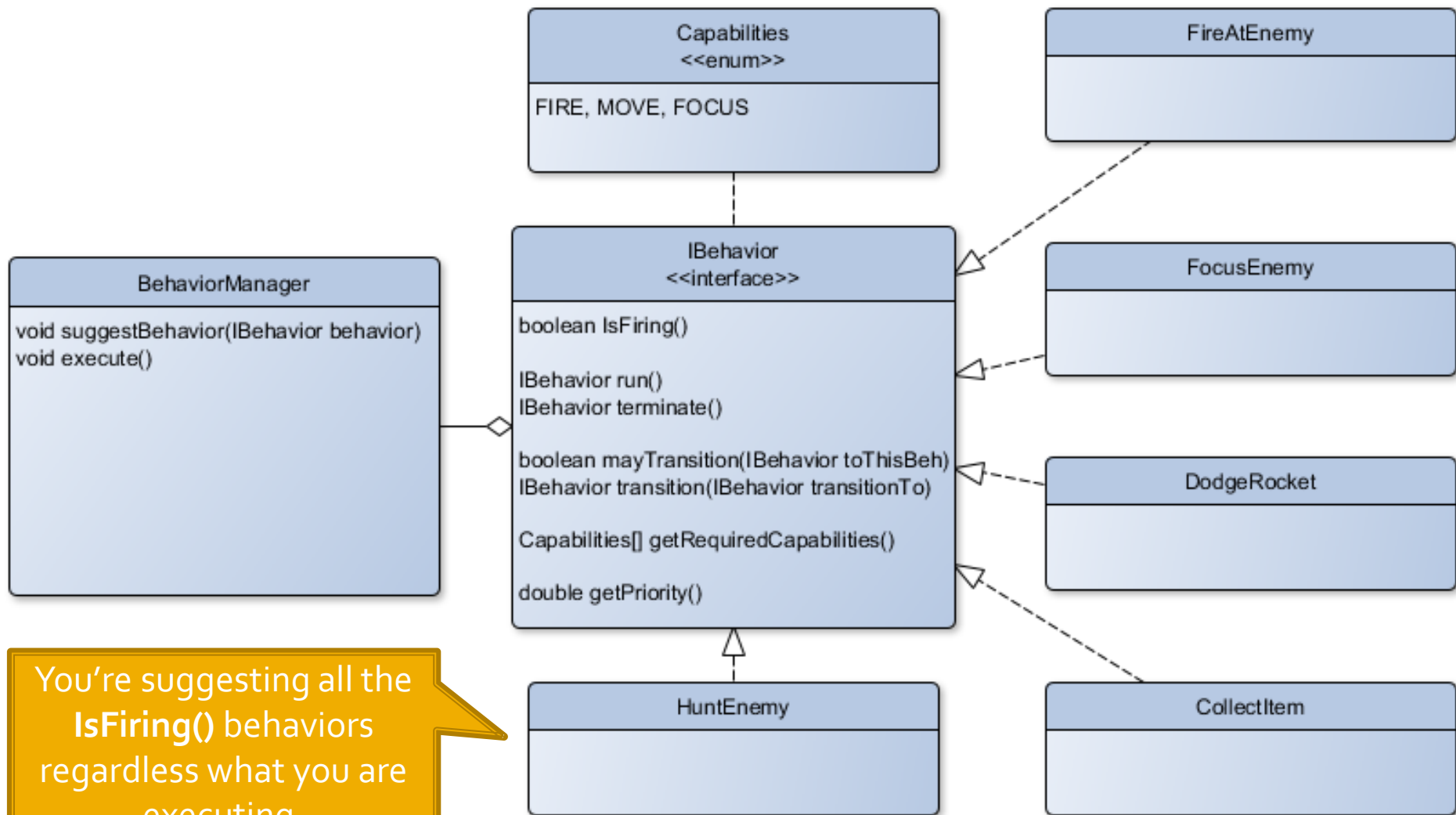
You decouple reasoning about goals and goals execution.



First you're suggesting new behaviors; then you decide what to execute and whether to switch.

Duelist

BOD with OOP

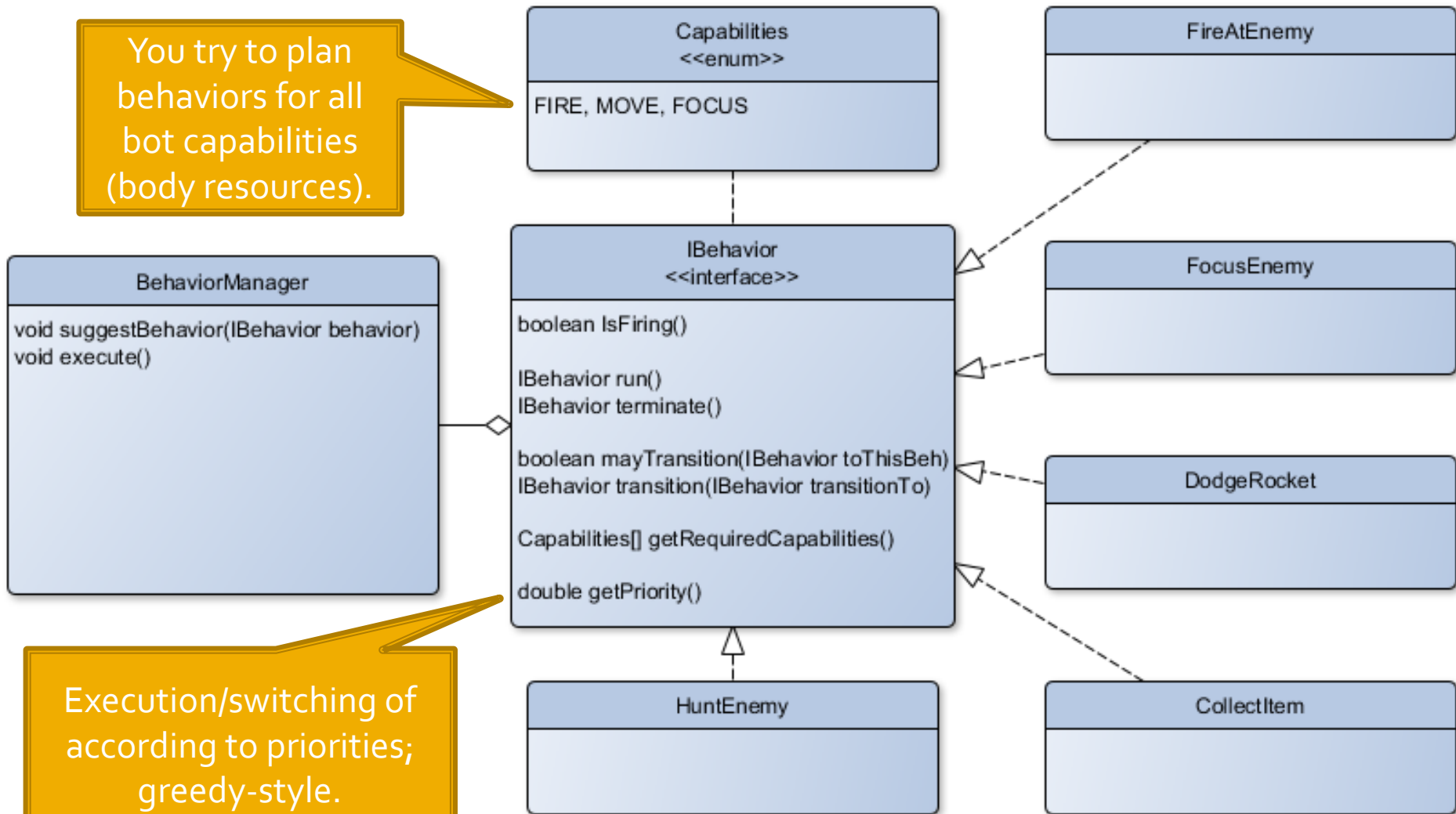


Duelist

BOD wit OOP



You try to plan behaviors for all bot capabilities (body resources).



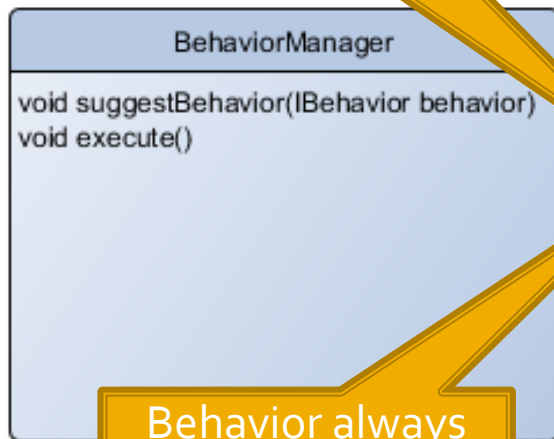
Execution/switching of according to priorities; greedy-style.

Duelist

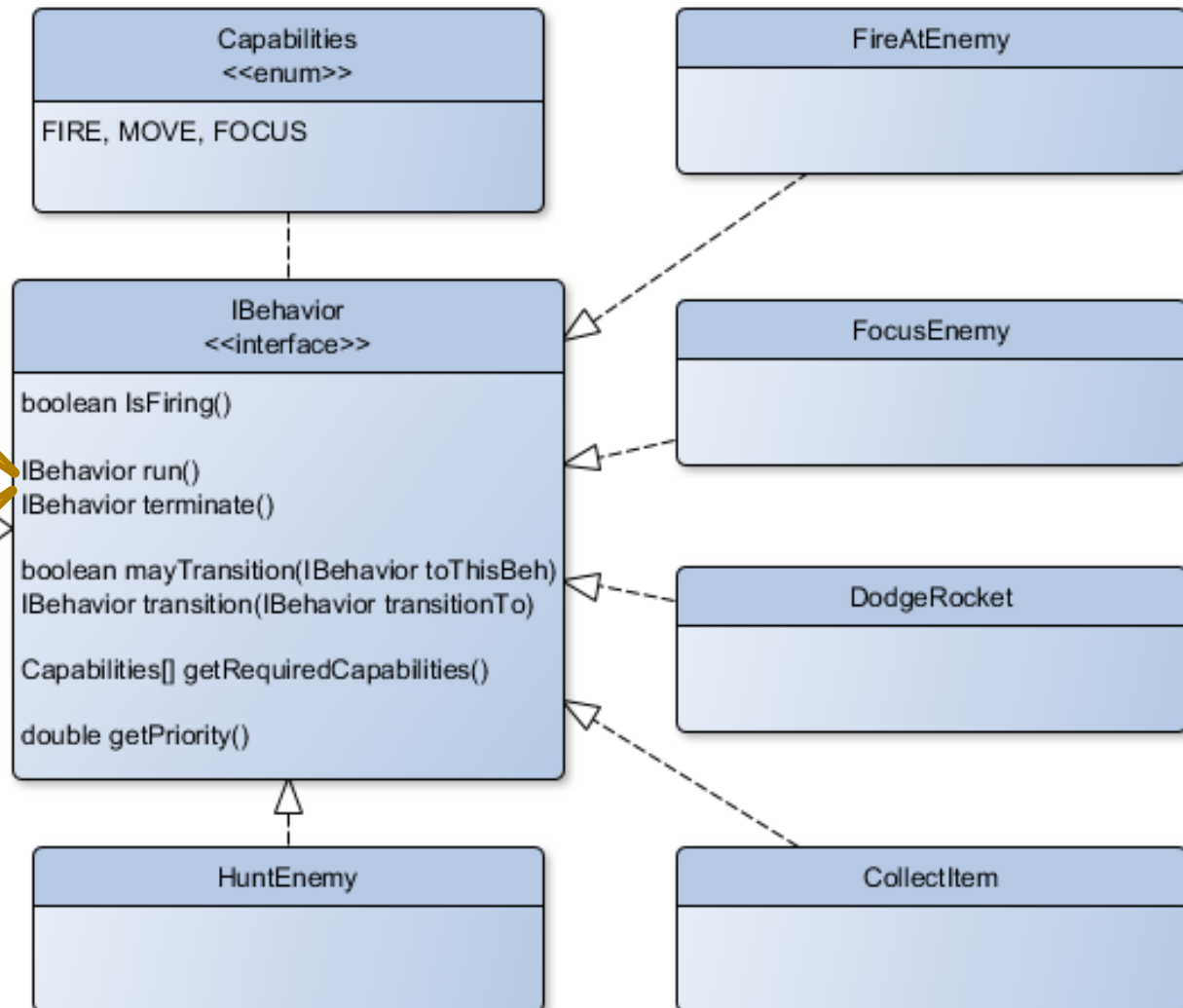
BOD with OOP



When behavior “is running” you typically returns “this”, once it ends, you return null.



Behavior always returns “behavior to execute next”, allows for SEQUENCES, ALTERNATIVES.

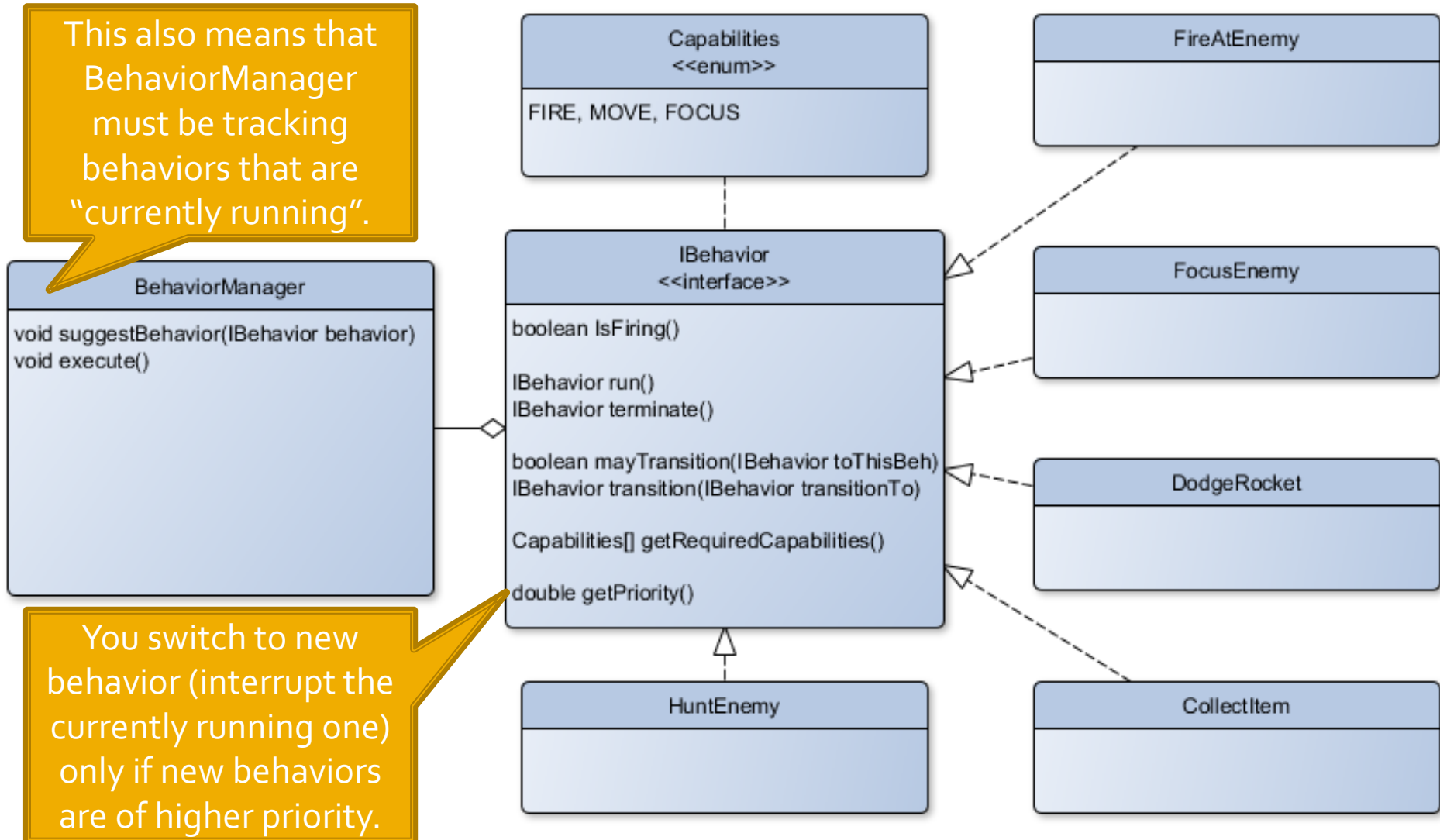


Duelist

BOD with OOP



This also means that BehaviorManager must be tracking behaviors that are "currently running".



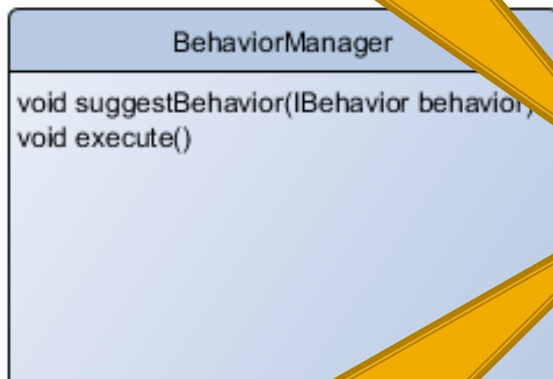
You switch to new behavior (interrupt the currently running one) only if new behaviors are of higher priority.

Duelist

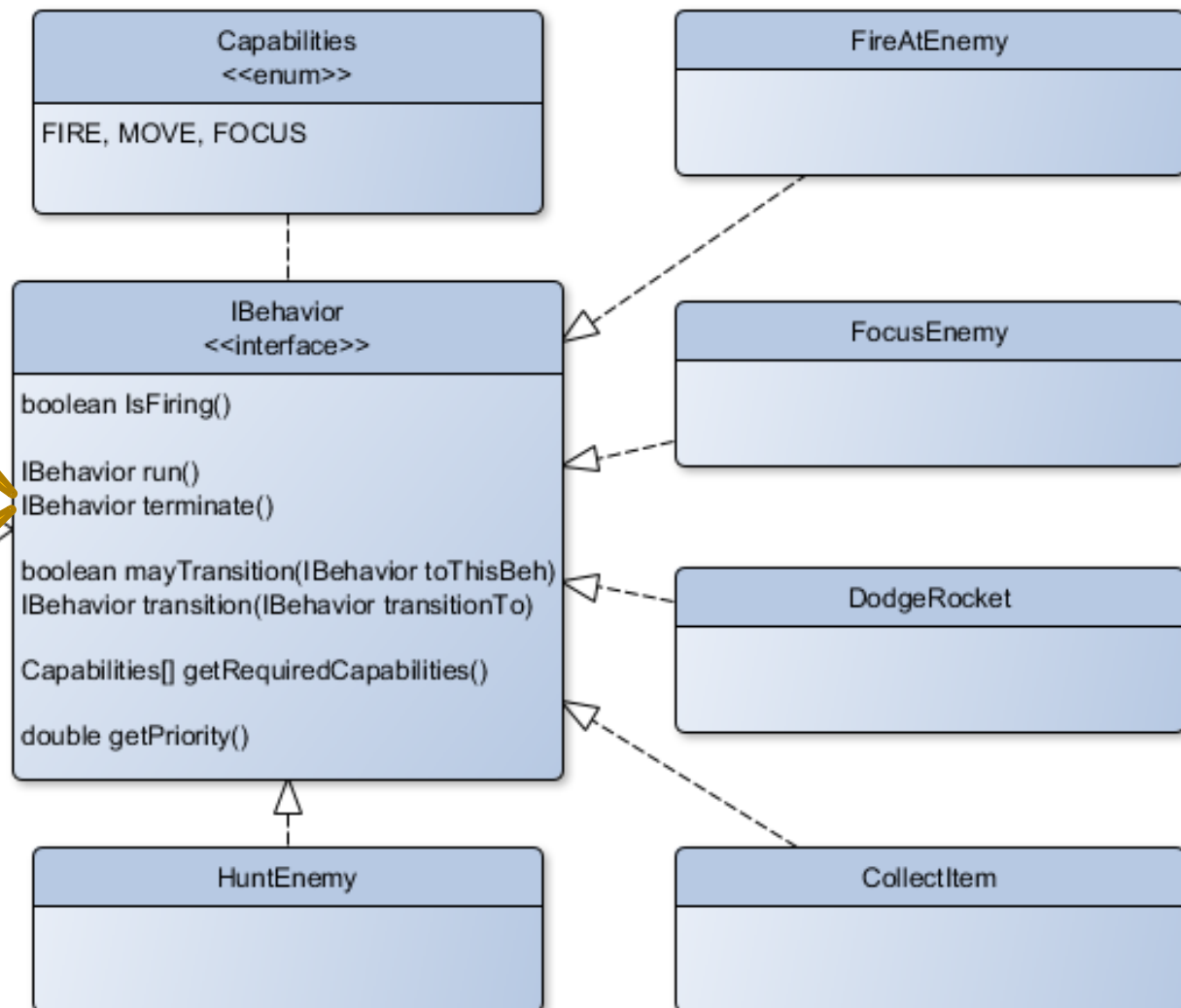
BOD with OOP



If you need to switch into a different set of behaviors, you have to terminate the running behaviors first.



If terminate() does not return NULL, you cannot switch it out yet! Allows to prevent interruption of, e.g. "grab the item that is very near".

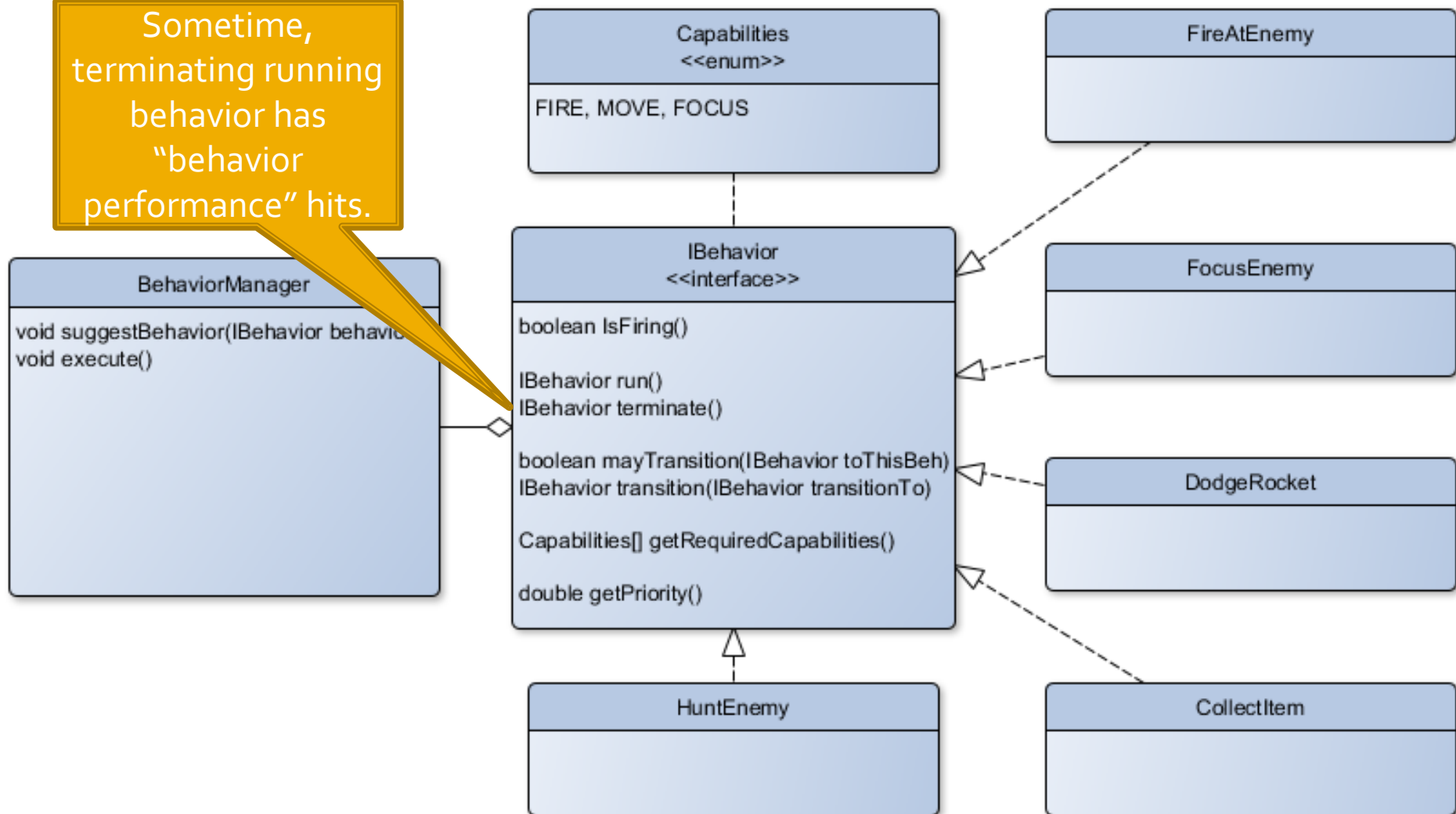


Duelist

BOD with OOP



Sometime,
terminating running
behavior has
"behavior
performance" hits.

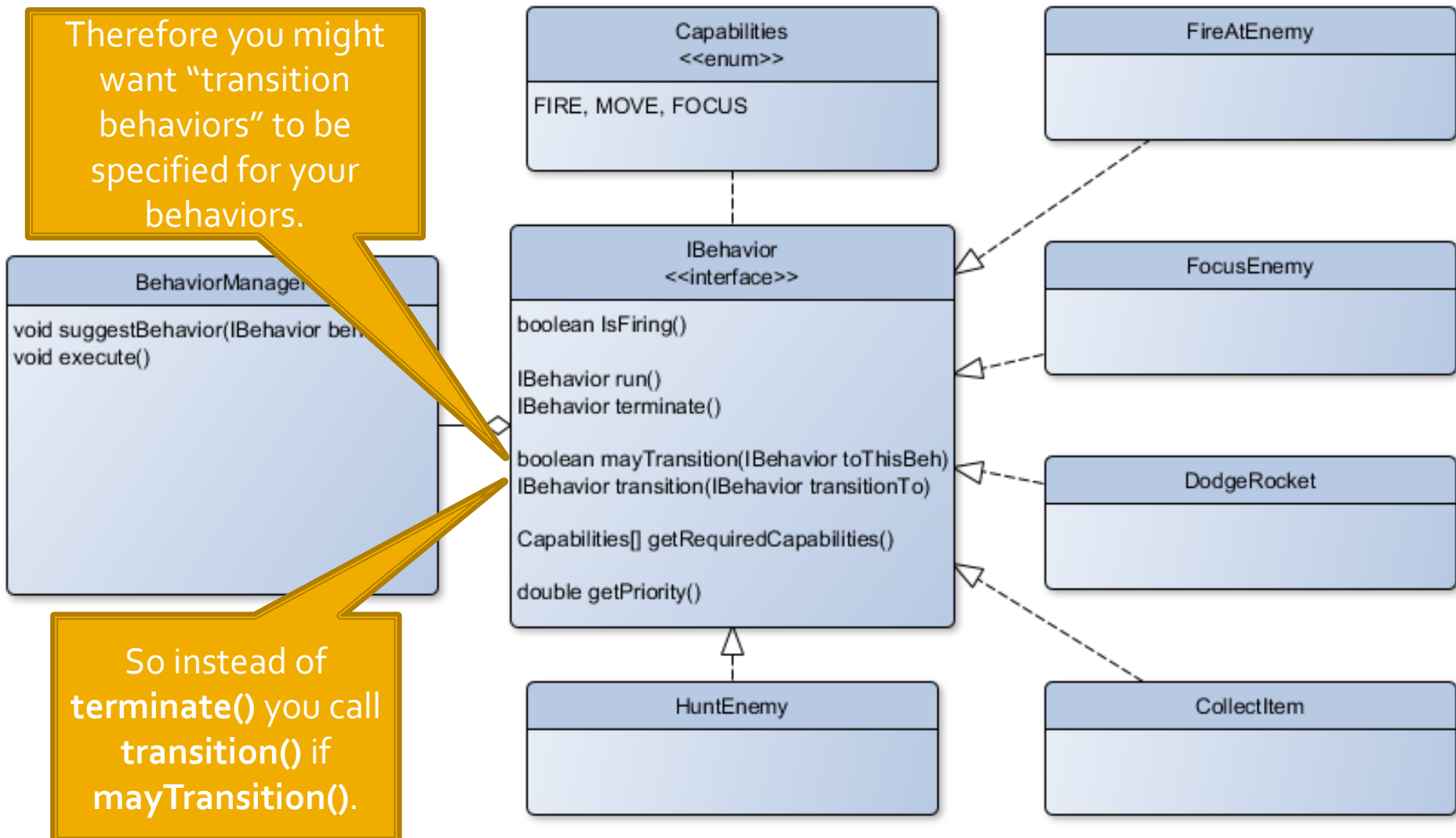


Duelist

BOD with OOP



Therefore you might want "transition behaviors" to be specified for your behaviors.



So instead of **terminate()** you call **transition()** if **mayTransition()**.

Duelist Bot



“Coding” Time!

Homework 04

Submissions



Submissions will happen through Gdrive again.

Once you finish your homework, ZIP UP your project folder COMPLETELY (except the `target` folder) and upload the ZIP file to shared shared GDrive folder into the `04-DuelBot` directory.

DM Tournament!



FIGHT!

DM Bot Tournament

Announcement!



- All **Death Match Bots** will automatically take part in DM Bot Tournament
 - 1 vs. 1, 40 frags, 20 minutes max
 - Format: Table (all vs. all)
 - DM-1on1-Roughinery-FPS map
- For the deadline, check the website!

Questions?

I sense a soul in search of answers...



ASK AT DISCORD!

<https://discord.gg/c49DHBj>