Разделение юнитов на команды.

Концепция в чем, ты делаешь интерфейс, в котором будет get-тер с ID-шником команды юнита, создаешь класс реализующий этот интерфейс (у меня это [AliveTeamMember.cs](https://github.com/DanP1e/AI_Teams/blob/development/Assets/Scripts/Team/AliveTeamMember.cs)). Это нужно для полиморфизма, все наследники [AliveTeamMember.cs](https://github.com/DanP1e/AI_Teams/blob/development/Assets/Scripts/Team/AliveTeamMember.cs) также будут являться членами команд и у них обязательно будет свойство “ TeamId ”, с этим и будем работать.

Если ты будешь использовать рейкасты или триггреы, для проверки на принадлежность к команде, тут ты прямо можешь искать у попавшегося объекта : AliveTeamMember или его наследника, если такого нет, значит это не то что нам нужно, делаем return.

Вот мы нашли потенциального члена команды, смотрим его ID. С этой информацией делаешь что нужно.

Если тебе нужно искать, например всех своих союзников на сцене (как я сделал поиск подходящей цели для атаки в [GuardianUnit.cs](https://github.com/DanP1e/AI_Teams/blob/development/Assets/Scripts/Creatures/Unit/GuardianUnit.cs), в методе TryFindTarget), то создаешь класс который будет содержать список всех членов команд (у меня это [TeamsCollection.cs](https://github.com/DanP1e/AI_Teams/blob/development/Assets/Scripts/Team/TeamsCollection.cs)), проходишь по списку, если ID == тот, что тебе нужен, делаешь что нужно. (так тупо удобнее и безопаснее искать чем через FindObjectWith..., но нужно позаботиться о добавлении юнитов в список).