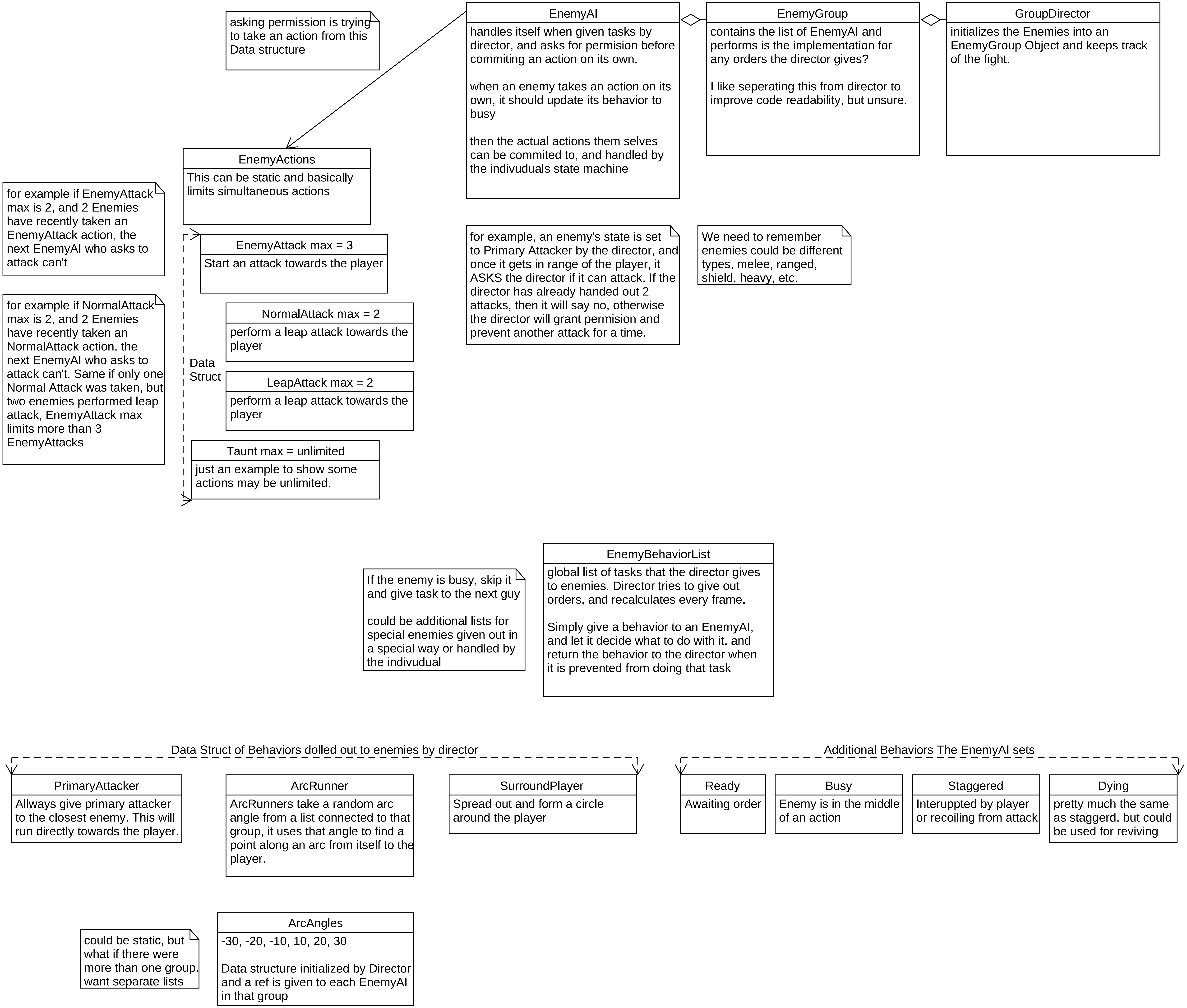

asking permission is trying to take an action from this Data structure

EnemyAI

handles itself when given tasks by director, and asks for permision before commiting an action on its own.

when an enemy takes an action on its own, it should update its behavior to busy

then the actual actions them selves can be commited to, and handled by the indivuduals state machine

EnemyGroup

contains the list of EnemyAI and performs is the implementation for any orders the director gives?

I like seperating this from director to improve code readability, but unsure.

GroupDirector

initializes the Enemies into an EnemyGroup Object and keeps track of the fight.

EnemyActions

This can be static and basically limits simultaneous actions

for example if EnemyAttack max is 2, and 2 Enemies have recently taken an EnemyAttack action, the next EnemyAI who asks to attack can't

for example if NormalAttack max is 2, and 2 Enemies have recently taken an NormalAttack action, the next EnemyAI who asks to attack can't. Same if only one Normal Attack was taken, but two enemies performed leap attack, EnemyAttack max limits more than 3 EnemyAttacks

Data Struct

EnemyAttack max = 3

Start an attack towards the player

NormalAttack max = 2

perform a leap attack towards the player

LeapAttack max = 2

perform a leap attack towards the player

Taunt max = unlimited

just an example to show some actions may be unlimited.

for example, an enemy's state is set to Primary Attacker by the director, and once it gets in range of the player, it ASKS the director if it can attack. If the director has already handed out 2 attacks, then it will say no, otherwise the director will grant permision and prevent another attack for a time.

We need to remember enemies could be different types, melee, ranged, shield, heavy, etc.

If the enemy is busy, skip it and give task to the next guy

could be additional lists for special enemies given out in a special way or handled by the indivudual

EnemyBehaviorList

global list of tasks that the director gives to enemies. Director tries to give out orders, and recalculates every frame.

Simply give a behavior to an EnemyAI, and let it decide what to do with it. and return the behavior to the director when it is prevented from doing that task

Data Struct of Behaviors dolled out to enemies by director

PrimaryAttacker

Allways give primary attacker to the closest enemy. This will run directly towards the player.

ArcRunner

ArcRunners take a random arc angle from a list connected to that group, it uses that angle to find a point along an arc from itself to the player.

SurroundPlayer

Spread out and form a circle around the player

Additional Behaviors The EnemyAI sets

Ready

Awaiting order

Busy

Enemy is in the middle of an action

Staggered

Interuppted by player or recoiling from attack

Dying

pretty much the same as staggerd, but could be used for reviving

could be static, but what if there were more than one group. want separate lists

ArcAngles

-30, -20, -10, 10, 20, 30

Data structure initialized by Director and a ref is given to each EnemyAI in that group