

- У вас теперь два операционных уровня на диаграмме, один сверху, один снизу
  - Было выдвинуто предположение, что фабрики должны быть на операционном уровне, т.к. они создают (то есть оперируют) объекты бизнес-логики. Но также эти фабрики используются MapCreator'ом, который находится на уровне бизнес-логики, поэтому, наверно, лучше сделать так, чтобы фабрики тоже были на уровне бизнес-логики.
- Кто вызывает ChangeState у моба? Кто вообще знает про класс StateMob, чтобы его туда передать?
  - - ChangeState(state: StateMob) вызывается в мобе в зависимости от количества здоровья. Следовательно, про класс StateMob должен знать класс Mob.
- "Мобы могут находиться в двух состояниях: паническом и состоянии по умолчанию. Оно задается при создании моба." --- это не то, что хотелось. При создании у моба состояние по умолчанию, использующее стратегию, которая задается при создании как раз. В паническое состояние моб переключается, если у него мало хитпойнтов.
  - При создании моба задается состояние по умолчанию. Состояние изменяется в зависимости от количества здоровья. Состояние определяет какую стратегию будет использовать моб.
  - Стратегия по умолчанию зависит от типа моба: например, ближний боец в состоянии по умолчанию имеет агрессивную стратегию, а боец дальнего боя в состоянии по умолчанию имеет пассивную стратегию (стреляет издалека). Каждый тип моба при снижении здоровья до определенного уровня меняет свое состояние на паническое состояние, что ведёт к применению панической стратегии. В панической стратегии моб восстанавливает здоровья до определенного уровня, и когда этот уровень будет достигнут, состояние меняется на состояние по умолчанию, и применяются стратегии агрессии и пассивного обстрела для ближнего и дальнобойного моба соответственно.

Работа со стратегиями и состояниями такова.

- Mob.InitializeState(state: DefaultState, mobType : Mob)
  - При создании моба на фабрике вызывается метод Mob.InitializeState(state: DefaultState, mobType : Mob), благодаря которому созданный моб имеет состояние по умолчанию. Далее в этом же методе через метод stateMob.ChooseStrategy(strategy : Strategy) выбирается стратегия, агрессивная или пассивная, зависящая от моба. И затем вызывается метод stateMob.LetsBehave(), в котором выполняется конкретная стратегия, реализующая контракт Strategy.Behave(mob : Mob).
- Mob.ChangeState(state: StateMob, mobType : Mob)
  - При изменении здоровья до определенного уровня (как при получении урона, так и при регенерации здоровья) у моба вызывается метод ChangeState(state: StateMob, mob : Mob), в который передается либо стратегия по умолчанию, либо стратегия паники (при регенерации здоровья до определенного уровня или понижении здоровья до определенного уровня соответственно). Далее выбирается стратегия через stateMob.ChooseStrategy(strategy : Strategy): паникующая или агрессивная либо пассивная. Затем вызывается метод stateMob.LetsBehave(), в котором выполняется конкретная стратегия, реализующая контракт Strategy.Behave(mob : Mob).

- "В зависимости от состояния зависит то" --- это вообще не по-русски :)

- :)
- В зависимости от состояния выбирается стратегия, на которую переключается моб.

- Оружие и боёвка относятся к тому же уровню, что и пользовательский интерфейс?  
Это оооочень странное решение

- Про оружие. Оружие, действительно, хорошо бы в предметную область, т.к. оно может быть нетривиальным для конкретной игры. Вас смутило то, что класс оружия на операционном уровне нарисован (согласен, надо бы перетащить оружие вниз, на уровень предметной области)? Это было нарисовано вместе, т.к. классы оружия, боевой системы реализуют паттерн "Стратегия". Не обязательно их вместе рисовать?
- Боевая система на операционном, т.к. она отвечает за инициализацию атаки. Например, запуск снаряда. После запуска, снаряд перемещается по карте и может столкнуться с объектом. Это столкновение обрабатывает реализация `ICollision` у объектов. Снаряд может подвинуть предмет, например. Либо, достигнув цель со здоровьем, донести на языке оружия, что пора понижать здоровье.
- Про опыт: умирая, моб сообщает персонажу, что пора повышать опыт за убийство. Мобы должны знать о персонаже, чтобы найти его. Пусть знают и то, как сказать радостную новость о заработанном опыте.

Такое решение разве плохо? Коллизии заимствованы из работы с Unity.

- Судя по диаграмме, у персонажа может быть один `Item`, у `Item`-а много персонажей. И если предмет в инвентаре, зачем ему координаты?

- Изначально предмет ведь валяется на карте. Его нужно подобрать.

- А боевая система знает об атакуемой цели?

- Ответ на этот вопрос в начале сообщения — это `ICollision`. Оружие сталкивается с целью и происходит обработка столкновения.