[https://github.com/codenjoyme/codenjoy/pull/122](https://github.com/codenjoyme/codenjoy/pull/122" \o "https://github.com/codenjoyme/codenjoy/pull/122" \t "_blank)

там смотреть будет очень сложно, потому как файлы мигрировали из основного репозитория в subrepo, часто переезжали с пакета в пакет, переименовывались

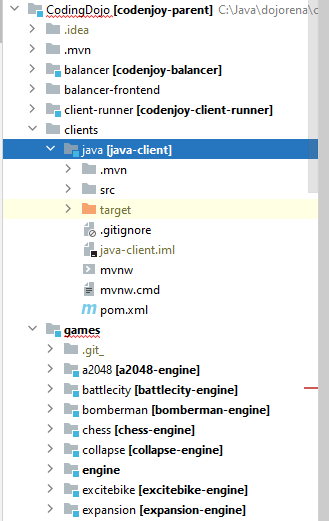
​

момент - gitflow не получилось применить для subrepo, там везде все изменения в master (я упустил этот момент как-то). Но в целом не страшно, т.к. основной репозиторий цепляется не за имя ветки, а за конкретный коммит. Но в следующий раз надо будет попробовать.

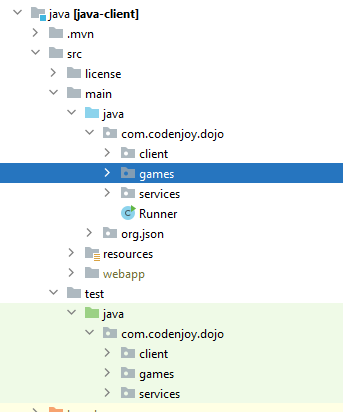
Хитрый момент с перепаковкой сырцов, чтобы все работало как прежде. Там надо будет созвон отдельный - расскажу как все работает.

​

Если вкратце вот наш репозиторий новый, он так же сабрепо подключен к основному



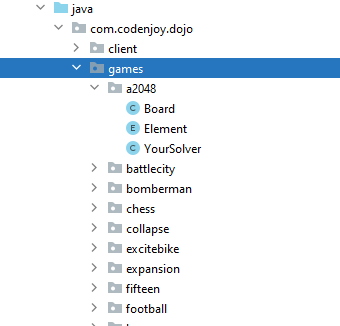
его структура такая



 client - пакет содержит все что касается клиентского кода

 services - содержит все, что пришлось забрать из Engine сюда, потому как client зависит от этого стафа

 games - внутри купа игрушек по пакетам побиты

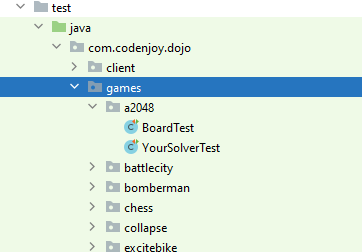


в каждой чаще всего встречаются

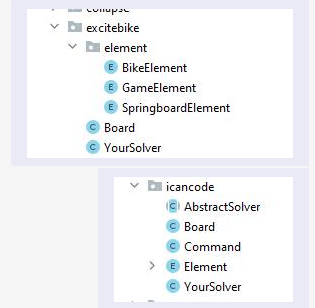
* Board - с методами-дополнениями для борды
* Element (который теперь тут сторится, раньше числился в игрушке как Elements)
* и YourSolver для написания кода играющего (реализовать с 1 солвером для всех игр не получилось)

​так же в тестах для каждой игрушки

* заготовка под тест для солвера
* и тест для борды

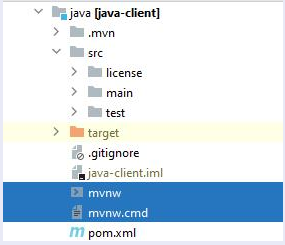


иногда для некоторых сильно умных игрушек, структура чуть меняется



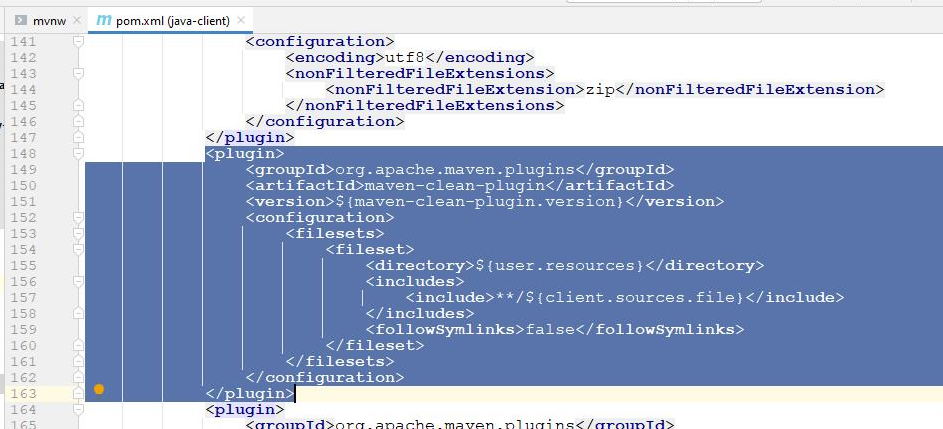
но суть та же. Тут три роли: борда, элементс мапа и солвер игрока

​в корне я оставил mvnw навсякий, может надо будет билдить игроку



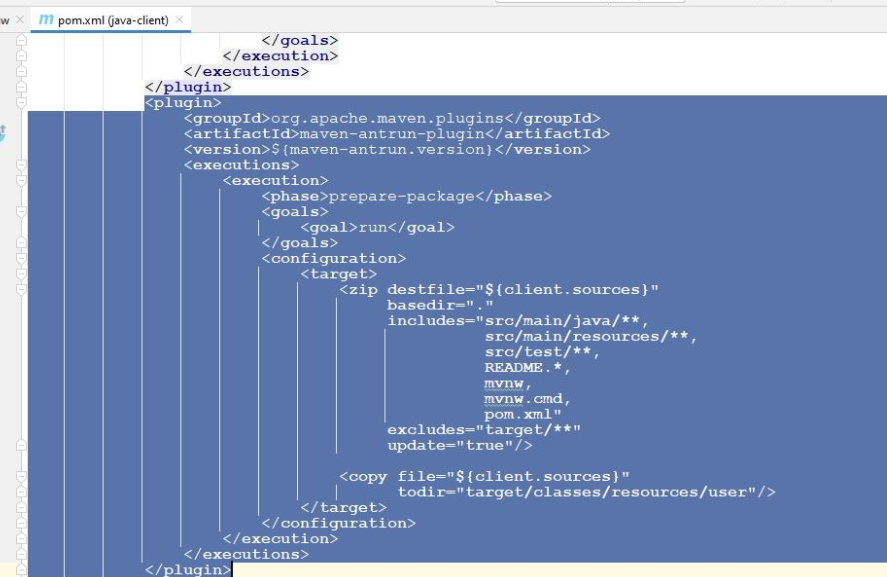
чтобы не ставил мавена на тачку

если зайти в pom.xml то там есть два момента

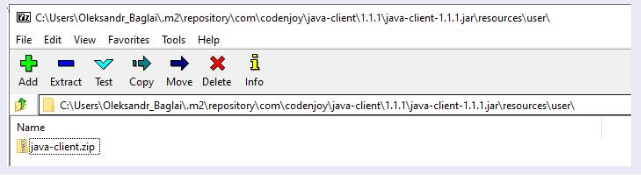


так мы подчистим файл с сырцами

а так мы его создаем во время сборки



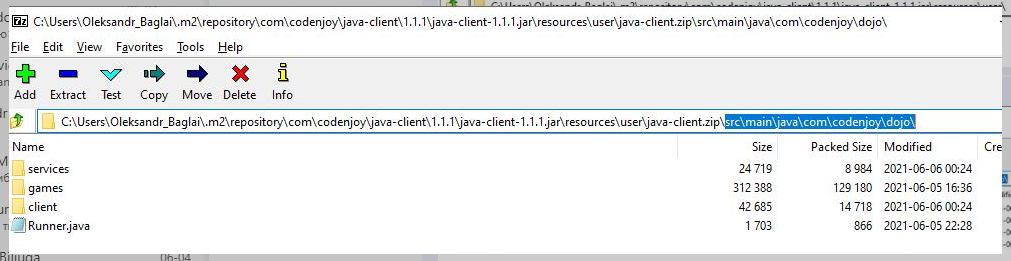
zip файл расположится в инсталируемом в .m2 папку jar этого проекта (при mvn clean install), ляжет zip в папке /resources/user/java-client.zip, а внутри будут все сырцы этого проекта





внутри сырцы java client

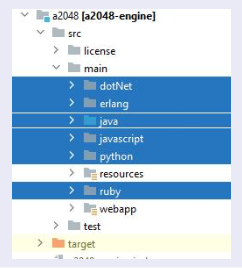




дальше немног моагии

​

каждая игрушка сама должна позаботиться о своих клиентах и созрать zip с ними. Раньше это делалось во время билда jar игры подобным образом ant'ом внутри maven'а из сырцов



все остается (пока не мигрируем все клиенты) как и раньше с одним лишь исключением - теперь java сырцы находятся в /resources/user/java-client.zip стороннего с точки зрения игры jar

​

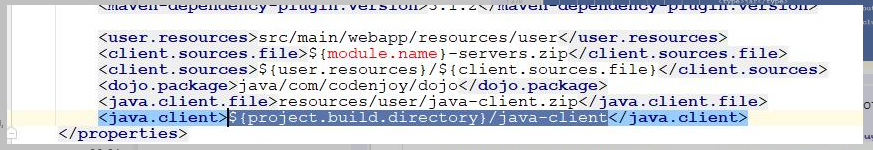
а потому будем колдовать

​

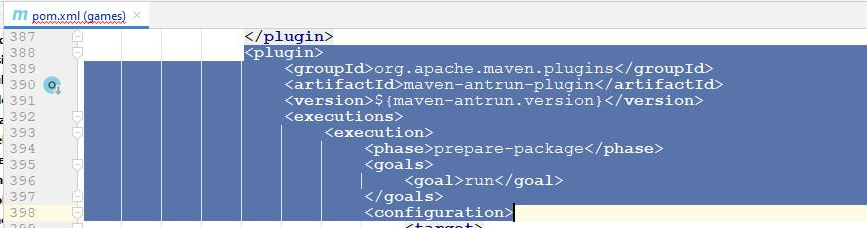
в games/pom.xml есть плагин



он распакует из java-client.jar дипенденси (которая явно в плагине указана) тот /resources/user/java-client.zip в папочку target/java-client



а после на другой фазе отработает ant



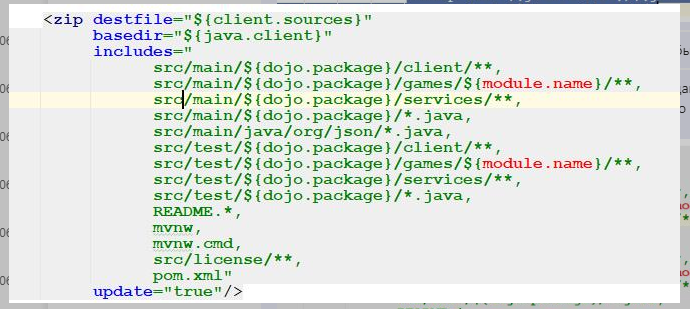
вначале он распакуте zip в той же папке



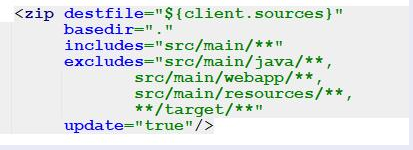
чтобы добраться до сырцов java-client

​

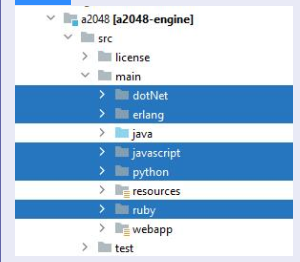
потом скопирует в <game>-servers.zip (тот что уже сервер отдавать будет) только те сырцы, что нужны (пакеты services, client, games/<game> со всем что надо для запуска проекта) - это java часть. Другие игры не включатся, только games/<game>



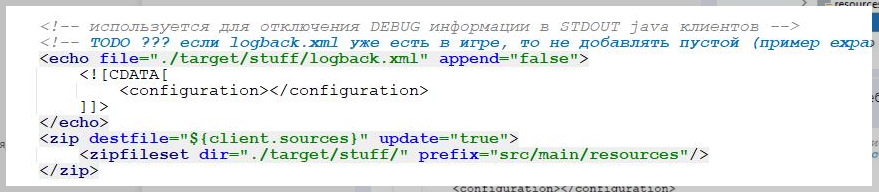
следом отработает копировальщик сырцов других языков из сырцов игрового проекта



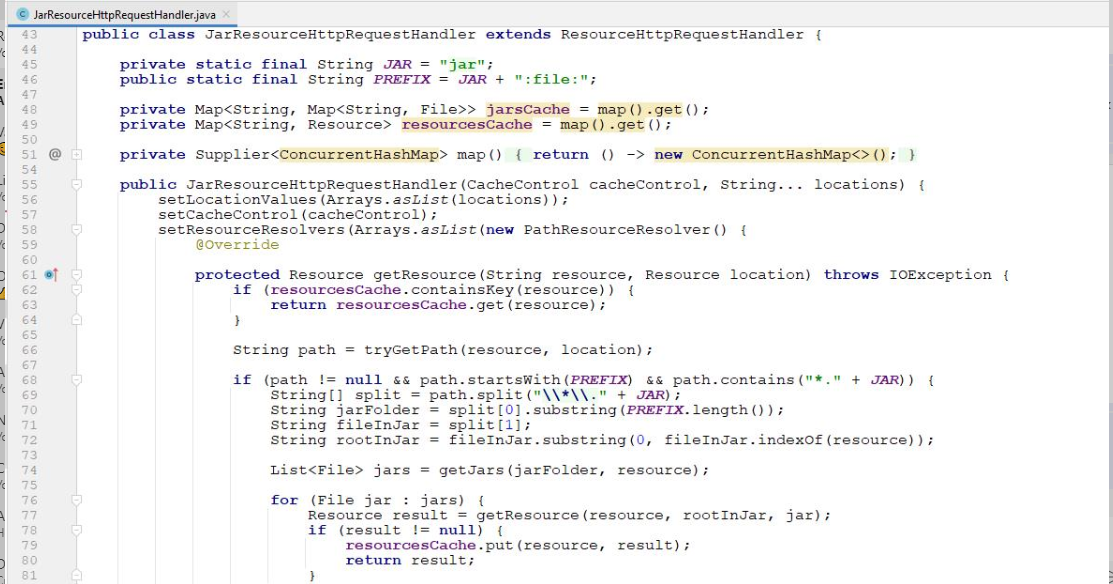
вот этих



дальше скопируется в zip logback.xml пустой для отключени вывода треша дебагового в клиенте (это старая штука)



и наконец перепакованный <game>-servers.zip ляжет в нужное ему место. Тут его будет искать codenjoy server когда будет запрос этого зипа как ресурса. Сервер ищет во всех jar дипенденсях с помощью этого малого



пока что так будет. Это временное решение, пока не перенесем все клиенты каждый в свой репо. Тогда можно будет переписать скрин отдачи клиентов и захардкодить там ссылки на github по каждому языку. А весь этот код выпилить нафиг вздохнув с облегчением )