## Особенности программной реализации

Опыт прошлой реализации выявил множество неоправданных недоработок языка golang и в новой версии уже нет попыток использовать многопоточность, приняты меры против одновременной записи и чтения “карт”, но, все же, несмотря на проблему цикличности включений пакетов, пакеты собраны в папки (кроме последнего пакета psychic, где файлы навалены в одну папку).

Основным элементом для создания образов различного вида является “структура”. Функции сохранения объектов образов и загрузки их из файлов настолько унифицированы, что становится возможным создать для этого универсальную функцию (пока не сделано). Это же касается функций работы с деревьями.

Всякий раз учитывается возможные размеры файлов памяти и нагрузка при работе функций так, что есть немалый потенциал для накопления данных и работы с ним. Избыточность искусственно созданных данных позволила оценить такой предел для персонального компьютера средней мощности.

Схема Beast получилась достаточно сложной и поэтому сделано описание функций по пакетам и в виде общей графической схемы. Но понять схему бед ясного представления ее функциональности – нереально. Так что в данном отчете будет такое общее описание.

### Описание схемы:

[Описание схемы](ччччч) подразделяется на описание схемы Btast и кода Пульта.

### Общая схема:

[ХХХХХХХХХХХХХХХХ](чччччч)

### Базовая структура

Варьируя базовые прошивки, эволюция создала огромного множество вариантов живых существ, различающихся особенностями наследственных механизмов адаптации. Различия начинаются от качества и возможностей рецепторных датчиках воздействия и кончая самыми эволюционно молодыми образованиями. Но никакие наследственные механизмы не справляются со сложными особенностями окружающих условий и воздействий. Поэтому множество механизмов адаптивности закладываются наследственно так, чтобы иметь возможность уточнить их реакции в период активной жизни особи. Они, естественно, имеют приоритет перед более древними реакциями и принцип такой приоритетности прослеживается на всех уровнях системы индивидуальной адаптивности.

В проекте основой распознавания иерархии образов является Дерево.

Чем меньше образов на данном уровне ветки дерева, тем более надежно распознавание в природных (и любых дискретных) сетях и тем более быстрый поиск в программных (последовательный алгоритм обработки). Эффект использования усложняющихся образов проявляется в фиксированном числе уровней (слоев колонок fornit.ru/43642).

Т.к. задается определенная **наследуемая** структура, то возникает возможность размножения с использованием наследственных структур партнеров, которые по заданному алгоритму обогащают и усредняют наследственную структуру нового существа. В проекте начато развитие такой функциональности.

### Проект не эмулирует, а представляет собой схема живого существа

Понятие живого существа сложилось из обобщения наблюдаемых признаков живых существ. Мы уверенно различаем в этом смысле камень, наколотую на булавку бабочку от живой бабочки. Стоит таким признакам стать неопределенными, и мы говорим: “как живое” или “как камень”.

Есть наиболее общий признак, который и наделяет все живое наблюдаемыми характеристиками живого (fornit.ru/64924). Это – наличие системы гомеостаза с жизненными параметрами, выход которых из нормы означает приближение к смерти – к неживому. Если такой выход оказывается более невосполнимым, то существо становится мертвым. Живое же существо всем своим поведением в конечном счете восполняет нарушения параметров гомеостаза у себя или у значимых для него другим особям, тем самым поддерживая жизнь вида.

Для параметров гомеостаза, напрямую не связанных с жизнеобеспечением (гон, потребность в общении, потребность в обучении и любопытство) организована цикличность: при нарастании параметра до максимума, он удерживается в течении 20 секунд, а потом сбрасывается. Это позволяет создавать достаточные по времени периоды специфических контекстов реагирования.