ChatGPT

Redesigning a Norn Genome: Engine Analysis and Plan

Engine Analysis

Neural Architecture: The Creatures series features an innovative artificial brain of roughly 9–10 interlinked lobes (regions), containing on the order of 900 neurons and thousands of synaptic connections (dendrites) [1]. Each lobe has a defined position and size within a fixed neural grid (64×48 in C3/DS), which determines how many neurons it contains [2] [3]. Early games (C1/C2) limited each lobe to connecting with at most two other lobes, but the later **Creatures Evolution Engine** (used in Creatures 3 and Docking Station) introduced **tracts** that allow far more complex interconnections among lobes [4] [5]. Neurons in this artificial life simulation are simple processing units that hold a numeric state which decays over time (simulating short-term memory) [6]. Dendrite connections propagate signals one-way between neurons, with adjustable weights and even inhibitory effects [7]. Uniquely, the C3/DS engine implements each neuron and dendrite as a tiny register-based program (the **SVRules** system) rather than a static weight value [8] [9]. This means each neuron's behavior is defined by a sequence of operations (e.g. summing inputs, applying thresholds, releasing chemicals) that can **mutate** without crashing the game – *every possible mutation produces a syntactically valid "program"*, even if nonsensical [10] [11]. This clever design yields a robust neural network where evolutionary mutations can alter neuron function or connectivity (within limits) while avoiding fatal errors. One important limitation is that in C3/DS the brain's structure (lobes) cannot increase via mutation – e.g. Norns won't spontaneously grow new lobes in later generations [12]. Therefore, enhancements to brain size or layout must be engineered via genetics tools rather than expected from natural mutation.

Biochemical System: Complementing the neural substrate is a rich biochemical simulation [13]. A Norn's genome defines dozens of **chemicals** (up to 255 identifiers possible) along with their initial concentrations, metabolic **half-lives**, and reactions [14] [15]. For example, there are chemicals corresponding to hormones, nutrients, waste products, neurotransmitters, etc., each decaying at a gene-specified rate. The *half-life gene* is in fact the longest gene in the genome, encoding the decay rate of every chemical – a mutation here can dramatically alter lifespan or metabolism (e.g. a longer half-life for the *"Life"* chemical can produce longer-lived or even quasi-immortal creatures) [15]. Chemical **reactions** (another gene type) convert one set of chemicals into another at defined rates [16] [17], enabling complex feedback loops and homeostatic mechanisms. (One cannot create something from nothing – at least one reactant must be present – though mutations have been known to create pathological loops like converting stored energy into toxins [18] [19].)

Crucially, the brain and biochemistry are deeply interwoven. **Chemical emitters** in the brain release neurotransmitters or hormones into the bloodstream based on neural activity [20]. For instance, the standard Norn brain has emitters that release *reward* or *punishment* chemicals when certain neurons fire, reinforcing learning. In Creatures 3, a special **neuroemitter** gene causes a Norn to produce adrenaline, fear, and "crowded" chemicals whenever it *sees a Grendel* (predator) – a built-in fight-or-flight response [21]. Conversely, **chemical receptors** can modulate neuron properties in response to body chemistry [22]. For example, high levels of a "coldness" chemical might alter neuron leak rates or trigger shivering behavior via receptor-linked neurons [22]. Many such links exist: the brain's **drive lobe** neurons reflect hunger, pain, tiredness, etc., rising and falling with chemical levels, while the decision-making neurons drive actions that in turn affect those chemicals (e.g. eating food reduces hunger chemicals). This creates a closed loop: *internal drives stimulate behavior; behavior changes chemical levels; and chemical feedback then alters neural state.* The game also employs an external **stimulus system**: events in the world (an object, the hand, etc.) can send stimulus signals to a creature. Each **stimulus gene** specifies how a particular event affects the creature's chemicals and drives [23]. For instance, being slapped might inject **Punishment** chemical and increase "Pain" drive, whereas hearing a friendly voice might reduce "Loneliness" drive by releasing comfort chemicals. In short, the engine simulates a rudimentary endocrine and nervous system: hunger, fatigue, fear, and other drives are chemically implemented and tied into neural circuits.

Behavior and Learning: With a neural network, drives, and stimuli in place, Norn behavior emerges from both pre-programmed responses and learned associations. Some behaviors are **instinctive**, defined by *Instinct genes* that pre-train the brain with certain reflex-like connections or reward expectations. For example, a typical Norn has an instinct that "when hungry and food is present, pushing the food is good" – effectively wiring a bit of initial preference to encourage eating. Instincts give newborn creatures a small head-start (they "know" some basics), but most complex behavior comes from learning. Norns learn via **reinforcement**: whenever they perform an action (e.g. pressing a toy, eating a fruit, running away from a Grendel), if that action satisfies a drive or produces a positive chemical change, a spike of reward chemical (dopamine analog) is released; if it harms them or is against their needs, punishment chemical is released. The neural network's plasticity (through adjustable synapse weights and neuron rules that respond to reward/punishment) means the creature will strengthen connections that lead to good outcomes and weaken those that lead to bad outcomes. The *Concept lobe* serves as a memory bank of past events and their emotional valence, so a Norn can remember, for example, *"I pushed the vendor and got food when hungry – that was good,"* or *"I wandered into the pond and almost drowned – that felt bad."* Over time, these memories influence the *Decision lobe*, which integrates current perceptions (sights, sounds via other lobes) and drives to choose an action. You can literally watch this process in action with the right tools: the Neuroscience Kit's **Decisions tab** visualizes which decision neurons fire (each corresponds to an action like *"push food"* or *"run left"*), and even shows live readouts of the creature's **Reward** and **Punishment** levels as it learns from consequences [24] [25]. This biologically-inspired learning mechanism, while simplified, gives rise to emergent behavior.

That said, the engine has **limitations**. The simulation had to run in real-time on 1990s hardware [26], so neural processing is not extremely deep (neurons do simple linear or logic operations). Sometimes creatures make irrational choices or fail to learn ideally, due to the limits of their brain design or conflicting drives. There are known bugs/quirks in the default genomes – for example, some drives or stimuli were miswired or certain genes never functioned as intended – many of which the community has since patched (more on this below). The world itself also constrains behavior: for instance, navigation uses built-in heuristics (Norns don't literally "see" the whole map, they rely on sensorimotor processing and the game's pathfinding). Nonetheless, within these limits, the Creatures engine is remarkably capable: it simulates digestion, metabolism, immune responses, organ health, and even *mutation* across generations. Each Norn carries a haploid genome of hundreds of genes [27], and when two breed, their offspring's genome is a mix with possible point mutations, enabling Darwinian evolution in the game. The engine's fundamental constraint is that all possible genes and behaviors must be predefined in the code – Norns can't evolve completely new organs or drives beyond what the engine supports, and extreme mutations may result in non-viable creatures. Therefore, to **maximize intelligence or other traits**, we must intelligently redesign the genome within the engine's existing framework (e.g. adding neurons up to the allowed limit, tuning chemical pathways, but not exceeding hard limits like 255 chemicals or 1024 neurons per lobe [3]).

Toolset Integration

Designing and testing a complex genome from the ground up requires leveraging community tools and resources. Fortunately, the Creatures community has developed a robust toolkit for genetics, brain visualization, and behavior monitoring:

- **Official Genetics Kits (C1, C2, C3/DS)** – These are the original gene editing programs released by CyberLife/Creature Labs. The **Creatures 3/Docking Station Genetics Kit** provides a full GUI to edit every gene in a creature's DNA and then hatch an egg from that genome [28] [29] . With it, you can add or remove genes (lobes, neurons, emitters, instincts, etc.), modify chemical half-lives, change life stage durations, and so on. It was once a paid tool but is now freely available [28] . The Genetics Kit is still one of the most powerful editors, though being a standalone application it requires running on a compatible system (Windows, with some tweaks for modern OS). It also allows live injection of eggs and some limited CAOS (the game's scripting language) for testing [30] . For our purposes, the Genetics Kit could be used for fine-grained editing of gene parameters that newer tools might not expose directly.

- **GEN Patcher Suite (v4.6.4 Lite)** – This is a modern community-developed web-based genome editing and patching tool, which will be central to our approach. **GEN Patcher** can load a Creatures `.gen` file (genome) and apply a variety of built-in enhancements and modifications via a user-friendly interface. Notably, it includes a **"Brain scaler"** that lets us resize the brain to a target neuron count. For example, we could scale a Norn brain up to *2400 neurons* or even the engine maximum of *3072 neurons* (which corresponds to filling the entire 64×48 neural grid) with a couple clicks [31] . The tool automatically adjusts lobe dimensions when doing this augmentation. GEN Patcher also has a **"Donor-less CFF"** module – essentially an integrated patch that applies the **CFF (Creatures Full of Fixes) genome improvements** to any loaded genome without needing a separate donor genome file [32] . This means we can easily incorporate community fixes (like the friend-or-foe lobe, corrected instinct genes, etc.) by toggling checkboxes. The suite features *behavior tweak packs* as well: for instance, a **"Libido++"** pack to boost reproductive drive, an **"Ambition++"** pack to increase goal-seeking and curious behavior, or *"Khan Mode"* to amplify aggression (useful for aggressive species) [33] [34] . Each comes with a slider to adjust intensity. We can selectively apply these to emphasize reproduction or intelligence as needed. Impressively, GEN Patcher Lite even includes built-in **visualization tools**: a **Brain Viewer and Lobe Editor** that will render the brain's lobes on the 64×48 grid and allow us to edit each lobe's position, size (W×H), and update interval, as well as display tract connections between lobes [35] [36] . This is extremely useful for verifying our brain modifications – we can literally see the new lobe layout and ensure no overlaps and that tract connections exist. The toolset also has **egg export features**: once our genome is finalized, GEN Patcher can generate a **.cos inject script**, a **DSAG** (Docking Station Advanced Gear) **egg agent**, or a breed file package to inject our custom eggs into the game [37] [38] . In short, GEN Patcher provides an all-in-one pipeline: *load genome → apply fixes/changes → visualize brain → output new genome or egg agent*. We will make extensive use of this for efficiency and to ensure compatibility with its features.

- **Brain and Neuroscience Tools:** To observe and debug creature behavior, we can use both official kits and fan-made tools. The **Neuroscience Kit** (bundled in Creatures 2 and later in C3's Science Kit addon) allows real-time monitoring of a creature's brain activity [39]. It displays each lobe's neurons as a grid of dots and lets us stimulate neurons to see responses (e.g. firing a sensory neuron to test a reaction) [40]. We can watch neurons light up as the creature thinks, and even see which decision neuron fires to trigger an action [41] – essentially a live thought monitor. It also shows **blue/red connection lines** between neurons, corresponding to different dendrite types, and provides readouts for **Reward/Punishment levels** so we can watch learning in progress [24]. Outside the game, the community-created **Brain in a Vat** tool is another way to "enjoy watching the brain in action" without all the in-game interface [42]. This tool can load a brain (or run an isolated brain simulation) and show neuron firing, useful for advanced AI tinkering or observing how a modified brain behaves in isolation. We will use these tools to verify that our enlarged/modified brain still functions (no lobe update errors or dead neurons) and that our Norns are forming associations as expected (e.g. checking that the new friend-or-foe lobe neurons activate appropriately in social situations).

- **Biochemistry Monitors:** The **Biochemistry Set** for C3/DS is an official utility to graph and inspect chemical levels in a creature over time [43]. We can use it to ensure our genome's metabolic loops function as designed – for example, monitoring that a *"dopamine"* (reward) chemical spikes when a desired action is done, or that the creature isn't suffering from chemical imbalances. The Biochemistry Set provides a live chart of up to 6 chemicals at once [44], and even allows injecting chemicals via a syringe interface for experiments (we can, say, test our Norn's response to an injected dose of a new hormone). Additionally, there are in-game kits like the **Health Kit** (shows vital signs and organ status) and **Science Kit** (with organ details and genetic scan) which will help ensure our creatures remain healthy under the new genome [45] [46]. For breeding and lifecycle, the **Breeder's Kit** in C2/C3 displays fertility, pregnancy progress, and even allows injecting aphrodisiac agents [47] – useful for observing how our modifications to reproductive drive and biochemistry affect actual mating outcomes.

- **Genome Analysis Utilities:** To assist with the genome editing process, we have tools like **Gendiff/Gencompare**, which can compare two genome files and list all differences [48]. This will be handy when cross-referencing our custom genome against the base CFF genome or other variants, to ensure we didn't miss any crucial edits. There are also platform-independent editors such as **liveGMS** and **easyGMS**, which provide alternative interfaces to edit genetics (these might be simpler for quick tweaks or if one is not on Windows) [49]. The community has compiled **Stimuli reference charts** [50] that list all stimulus numbers and their default effects on standard creatures – an invaluable reference when we adjust stimulus/response genes (we need to know, for example, that stimulus #79 is the "eat food" event and normally decreases hunger and provides reward, so if we create a new instinct or change it, we consult these notes).

- **Scripting and PRAY Tools:** Because deploying a new genome often means distributing it as a breed or an agent, tools like the **PRAY Builder** help package genome files into egg agents [51]. GEN Patcher already includes an agent packaging option (with DSAG and standard **.agent** outputs) to automate this [38]. If needed, we can manually use **PRAY scripts** – the community even has a *PRAYEgger* script generator that asks a few questions and outputs a ready-made PRAY file for an egg agent [52]. Additionally, if we venture into any CAOS coding (for advanced behavior triggers beyond pure genetics), the **CAOS Debugger** can attach to the game to help debug agent scripts [53]. However, our focus will be on genome (internal) changes rather than external agent coding, except possibly to use existing third-party agents that facilitate breeding (e.g. there are *automatic* *inseminator* gadgets, etc., but those are optional once our genome itself encourages ample breeding).

In summary, we will combine these tools as follows: use **GEN Patcher** for rapid genome prototyping (applying known fix-packs and resizing the brain), use the **Genetics Kit or GMS editors** for any fine editing not covered by GEN Patcher, then employ visualization kits (Neuroscience/Brain-in-a-Vat, Biochem Kit) to observe our "upgraded" Norns in action and ensure the changes have the intended effect. We'll also use reference materials from creatures.wiki and past projects (e.g. sample genomes like **Civet Norn's full genome listing** [54]) to guide our edits.

Genome Design Plan

With the engine's capabilities and constraints in mind, we propose to **redesign a Norn genome from scratch** (using an existing improved genome as a foundation) to maximize both **intelligence** and **reproductive success**. The strategy involves enhancing the brain's capacity, refining biochemistry for motivation and learning, and tuning instincts/behaviors – all while staying within the simulation's limits for stability and compatibility.

1. Brain Structure Optimization

Expand Neural Capacity: We will increase the size and neuron count of key brain lobes to give Norns a higher "mental resolution." The concept lobe (which stores event memories) is a prime candidate – in the default genomes it contains ~500–600 neurons [55] , but many brain grid cells are unused. Using GEN Patcher's **Brain scaler**, we can target, say, **2400 neurons total** (the tool's default augment) or even push to the ~3000 maximum [56] . This expansion will be done in a balanced way: rather than one giant lobe, we allocate neurons to various functions: - The **Concept/Idea lobe** gets a major expansion so creatures can remember more unique events or concepts. A larger memory capacity means a smarter creature that can distinguish fine differences and retain lessons longer. - The **Perception and Attention lobes** (handling sensory inputs and focus) will also be enlarged modestly, ensuring the Norn can process more simultaneous stimuli without being overwhelmed. - We will **introduce new lobes** that have been proven useful in community genome projects. Notably, a **"ForF" (Friend or Foe) lobe** will be added to support social intelligence. This lobe, present in CFE/CFF genomes, lets creatures form lasting memories of individuals they like or dislike [57] . Technically, it gives the Norn neurons to encode other creature IDs and an affinity value. By adding the ForF lobe (and its associated tracts), our Norns can, for example, remember that a particular Norn is their friend (or that a Grendel should be feared) beyond the immediate moment. This prevents the default situation of "goldfish memory" where a Norn forgets who hit them as soon as the stimulus is over. - Additionally, we add the **"Fullness" lobe** introduced in the CFF genome [58] [59] . This lobe helps regulate eating behavior by representing how full the creature's stomach is, separate from the hunger drives. By including it, our Norns can learn to stop eating when full and better manage their nutritional needs (preventing overeating or starvation due to lack of satiety signal). - If possible, we consider adding a **"Sensorimotor" integration lobe** or other small lobes that some advanced genomes use for specific behaviors (e.g. the 2017 genome or TWB added certain brain tracts for improved drowning response and other behaviors [60] [61]). However, these often rely on game scripts; at minimum we will ensure the brain has the lobes and tract for the **Emergency Drowning** response (so that if the Norn is suffocating, it triggers an instinct to seek home/air) [62] . This involves a tract from a sensory input (oxygen level) to a decision to move upward or "go home" when drowning – a small but life-saving circuit.

Optimize Lobe Parameters: For each lobe, we will review and adjust parameters like **update rate** (how frequently the lobe's neurons tick) and **leakage** (memory decay rate). A faster update rate for the decision lobe, for example, can make a creature more responsive (thinking ticks happen more often). We must be cautious not to overspeed it (to avoid erratic behavior or high metabolic cost). We'll use values from known good genomes: the CFF and TWB projects often fine-tuned these. The **GEN Patcher's Lobe Editor** will allow us to tweak X/Y positions and sizes – we must ensure no lobes overlap on the 64×48 grid and that all neurons remain within bounds [36]. We might place new lobes (ForF, Fullness) in previously empty brain regions. The tool's tract viewer helps verify that connections (dendrite tracts) link the correct lobes after any repositioning [63].

Tract and Rule Adjustments: With more neurons and lobes, we should also enrich the connectivity. We will add additional **tracts** (neural pathways) or expand existing ones so that important information flows reach the decision center. For example, we ensure that: - The **Drive lobe → Decision lobe** connections are set to additive mode (this is the *"Drive->Comb tract learning fix"* from CFE [64]). In the original genome, a bug caused new drive inputs to overwrite old ones rather than sum, impairing learning. We will use the corrected tract so that multiple drives can influence a decision neuron properly (e.g. both hunger and tiredness can weigh on whether to sleep or eat, rather than one canceling the other) [64]. - Connect the new **Fullness lobe** into the decision-making circuit so that if a creature is extremely full, it might choose to rest instead of continue eating. Also link Fullness to the **Hunger drive** neurons via receptors: fullness should reduce hunger drive chemically. - Wire the **ForF lobe** into the Norn's approach/avoid decisions regarding other creatures. Practically, this means tracts from the ForF lobe to the Decision lobe (or to an intermediate "Combination" lobe that influences social behavior). A high "like" value in ForF could encourage the Norn to approach or mate with a creature, whereas a high "dislike" could feed into the same fear/flight response as seeing a Grendel. - **Sensory improvements:** If possible, increase the resolution of the *Vision* lobe (which maps what the creature sees) and *Auditory* lobe. More neurons here mean the creature can distinguish more object types or directions. We will also ensure that **noun and verb lobes** (which handle language, i.e. recognizing object names and action words) are not neglected – a smarter Norn should learn language quickly. We may not need to expand these much, as language is limited by game vocabulary, but we will include all enhancements from CFF that, for example, fix the **Yes/No** understanding (CFF genomes allowed creatures to properly respond to the hand's yes/no commands) [65].

The net effect of these brain changes is a Norn with a **larger, faster brain** that can form more complex thoughts and memories. By adding established extra lobes, we aren't venturing into unknown territory – we're using improvements that have been tested by the community (CFE/CFF) to yield smarter decisions (like remembering friends and moderating hunger). All these changes will be done in a way that remains **compatible with GEN Patcher's capabilities**: notably, GEN Patcher's "Donor-less CFF" option will automatically add the ForF lobe, Fullness lobe, and all associated genes we need [32] [57]. We will leverage that, then manually adjust neuron counts via the brain scaler (which the tool supports up to the 3072 limit) [66]. By sticking to GEN Patcher's known feature set, we ensure our custom genome can be saved, exported, and injected without errors.

2. Biochemical Enhancements

To support higher intelligence and prolific breeding, the biochemistry must provide the right balances of neurochemicals, energy, and feedback loops:

Enhanced Neurotransmitters: Learning in Norns is driven by **reward** and **punishment** chemicals (often analogized to dopamine/serotonin). We will verify that the **reward pathway** is robust – e.g. when a creature satisfies a drive (lowers hunger, etc.), the genome should emit a burst of reward. The default genomes do this, but we can tune the amount or half-life of the reward chemical so that the reinforcement is neither too fleeting nor too excessive. A slightly longer-lasting reward chemical (within safe limits) could help the Norn better reinforce good behaviors (essentially a stronger "dopamine hit" for doing the right thing). However, we must also ensure *punishment* chemicals (like when they do something harmful) are effective but not debilitating – if punishment is too high or lasts too long, it can stress the creature and inhibit exploration. So a balance will be struck: quick, sharp punishment to discourage bad actions, and moderately lasting reward to encourage repetition of good actions.

Drive Chemistry and Motivation: Each primary **Drive** (hunger, thirst, sex drive, boredom, etc.) is represented by one or more chemicals. We plan to adjust these drives to encourage both survival and reproduction: - For **hunger/thirst**, we ensure the metabolic pathways for nutrients are efficient. Using the community's **Gizmo genome** as inspiration (which rebalanced digestion rates), we might increase the rate at which eating food reduces the Hunger drive, so Norns get satisfied quicker and learn "eating = not hungry" clearly. The Gizmo edits aimed to maintain proper protein/fat/starch balance [67], preventing the creature from starving while carrying excess of one nutrient; we will incorporate similar *rebalanced digestive system* genes to avoid any one nutrient running out first. - For **sex drive (libido)**, we will *lower the threshold for mating behavior* and *increase the chemical triggers for mating*. The male reproductive system in default genomes had some issues – the **CFF Male Reproductive Overhaul** fix will be applied, which, among other things, fixes the testosterone emitter and adjusts the **"libido lowerer"** chemical's half-life [68]. In CFF, they extended the libidoLowerer's half-life to shrink the female fertile window (prevent back-to-back pregnancies) [69]. Since our goal is maximizing reproduction, we might moderate that particular change: perhaps use a slightly shorter half-life so females return to fertility sooner after giving birth. However, we must be careful; constant breeding can harm longevity. We will still include a reasonable **pregnancy cost**: in C3, pregnancy consumes stored protein and requires a minimum muscle tissue to conceive [70]. We'll keep that, as it adds realism and prevents malnourished Norns from reproducing themselves to death. But we will introduce a **pro-breeding boost**: GEN Patcher's *Libido++ pack* specifically increases the drive for mating [71]. This likely works by increasing the rate at which the sex drive rises or how strongly creatures seek each other when lonely. We will apply Libido++ at a moderate setting to ensure our Norns have high **social/sexual motivation** – they should actively seek mates when fertile. Additionally, the **"pheromone"** system (creatures emit pheromones when fertile that attract others [72]) will be checked – if needed, we'll amplify that signal so that when a female is in heat, males in the vicinity get a stronger stimulus to approach. - **Energy and lifespan:** To allow both high brain activity and frequent breeding, the creature's energy metabolism must be efficient. We will incorporate the **lactate cycle fix** from CFF/TWB [73]. In the original game, muscle exertion produced *lactic acid* which could build up harmfully. The fix introduces a reversible cycle converting lactic acid to pyruvate when oxygen is plentiful, and back when oxygen is low, essentially using lactate as a backup energy source and preventing muscle damage [73]. This means our active, fast-moving Norns won't suffer as much from running around. We will also ensure that **glucose usage** for movement is properly balanced (the TWB genome made creatures consume glucose for activity, making exercise affect blood sugar) [74]. Our Norns should eat enough to fuel their busy brains and romantic escapades! If needed, we can slightly lengthen the half-life of the **"Life"** chemical so that baseline lifespan is extended a bit, giving them more years to reproduce and learn. For instance, instead of the normal ~5 hour life, we might target 7–8 hours (real-time) on average – this can be done by tweaking the aging chemical's decay rate [15]. This longer lifespan combined with more efficient breeding means more offspring per Norn. However, we'll keep aging in place (no perpetual immortality), to ensure natural turnover and evolution.

Stress and Safety Chemicals: We will integrate the improvements that enhance survival responses: - Enable the adrenaline/fear response genes that were *meant* to trigger when predators are near. In the base game, some of these genes were broken. The community fixes ensure that when a Grendel is detected, the Norn actually releases **Adrenaline** and experiences **Fear**, motivating it to flee or defend itself [75]. We will include that fix so that our intelligent Norns *don't ignore danger*. Adrenaline also has a beneficial effect of temporarily boosting certain bodily functions (in Creatures, it tends to reduce pain and sharpen attention briefly), which can help in emergencies. - Implement the **"Hunger Overwhelmsion" organ** from CFF [76]. This is a genetic addition that makes extreme hunger override other drives. If a creature is starving, this organ (really a set of reactions) will ensure *hunger becomes the top priority*, preventing a scenario where a Norn, say, plays with toys (satisfying boredom) while starving to death. This is important for safety – it kicks in a survival instinct. - We will also include the **"Swearing lets off steam" fix** [77] which resolved a glitch where creatures would get caught in an endless cycle of complaining about crowding. In our highly social, breeding-oriented Norns, we don't want a crowding feedback loop to paralyze them. The fix makes the act of "expressing" (complaining) not erroneously *increase* crowdedness, so they won't get stuck constantly moaning about overcrowding. [77] This way, in a colony setting, they can tolerate each other enough to mate and cooperate. - **Immune and health tweaks:** While not directly about intelligence or reproduction, a healthier creature can devote more energy to those. The **TWB (True Warmblood) genome** overhauls body temperature and immune response [60]. If feasible, we'd like to incorporate at least partial TWB features: for instance, use TWB's method for fever response (so sick creatures get elevated temperature and seek rest). Also, TWB introduced a more realistic **organ deterioration** model and an **immune system** that fights antigens better [78]. Adopting these would reduce random death by illness, giving Norns more opportunity to reach old age and reproduce multiple times. These are extensive changes, so we will ensure any TWB gene added doesn't conflict with our other modifications. GEN Patcher's CFF patch does not include TWB features by default (TWB is a separate base), but we could apply TWB v3 genes separately if needed. At minimum, we'll incorporate TWB's *coldness and heat* adjustments (so our creatures seek appropriate environments and don't lose fertility due to temperature stress) and the **"body heat from activity"** feature (creatures burn calories and warm up when active, preventing an exploit where they could become obese without consequences) [74].

In summary, our biochemical plan creates **positive feedback loops for learning and mating** (ample reward for good behavior, strong drive to seek mates) and **negative feedback loops for harmful situations** (pain, fear and hunger properly push the creature away from danger or towards food). We'll be mindful of not pushing any value beyond engine limits – e.g., chemical concentrations above what organs can handle – and we'll test with the Biochemistry Set to watch that, for instance, a highly active brain doesn't deplete **ATP** (energy) to fatal levels. (Notably, brain organ failure occurs if ATP drops too low [79], so we'll confirm our Norns produce enough ATP via respiration genes to supply the bigger brain's needs.)

3. Behavioral Instincts and Tuning

To maximize success, we need to hard-code some wise instincts and ensure the learned behaviors don't stray. We will revise the **Instinct genes** and stimuli responses:

Proactive Foraging and Exploration: Boredom in Norns can sometimes lead them to useless actions. We will adopt the approach of the 2017 community genome which *weakened or silenced certain unhelpful instincts (especially those related to boredom)* and *added productive ones* [80]. For example, some original instincts would prompt a Norn to periodically play with toys when bored – fun, but maybe not optimal if hungry. We will reduce the strength of such boredom-relief instincts [81] so that when our Norn is bored it is more likely to translate that into exploring the world (which might lead to finding food or mates) rather than just fiddling idly. In place, we strengthen instincts for **foraging**: e.g. an instinct that *when bored and not hungry, wander left/right* to discover new areas. Also, ensure strong instincts remain for *seeking food when hungry* and *drinking when thirsty*. The Extended Feeding Instincts from CFF add multiple instincts covering all food types [81] – meaning the Norn is innately driven to eat not just “food” objects but also fruit, seeds, etc. when hungry. We’ll include these so they won’t starve even if one food category is absent.

Mating and Social Instincts: We will introduce or amplify instincts that nudge creatures toward mating behaviors. For example, *if male and high sex drive and sees a female, then “pull” (kisspop) the female* with an innate reward. Likewise for females, an instinct to approach males when ready to mate. The **CFF genome already improved social instincts** – it added a **Nesting instinct** where pregnant females seek the Norn Home to lay eggs safely [82]. We’ll include that, as it indirectly boosts reproductive success by protecting eggs/offspring. Also, post-birth, creatures have an instinct to **drop an egg when at home** [82], ensuring eggs don’t remain hidden in inventory. We might add a mild **parental instinct**: though Norns generally aren’t great parents, we can add a gentle instinct for adults to occasionally *“approach younger creature”* when they hear a baby cry, or to *tickle (encourage)* crying babies. This could improve survival of offspring (teaching them or moving them to safety).

We will ensure our Norns have the **“friendly” instincts** from CFE: those edits made decision-making less random and more survival-oriented [83]. For instance, CFE Norns have better sense not to endlessly push grendels or smack things that hurt them. Our genome, with the ForF lobe, will naturally give them some of that sense (they remember bad encounters), but we’ll double-check all original instincts for any illogical ones and remove any that conflict with our goals.

Safety and Avoidance: Key safety instincts include *“if very scared, run away”* and *“if in pain, retreat”*. We will add an instinct like: *When drowning (underwater and oxygen low), then activate the “up” drive (to surface)* – essentially hardcoding a last-resort behavior for drowning beyond the brain tract we added. We also want them to avoid dangerous areas: e.g., an instinct *“if sees a lift (elevator) and life force low, use lift”* could be added to encourage escaping to a safer level, but that might be too specific. At minimum, we use the standard *“if sees Grendel, then feel fear and run”* instinct that might have been missing or weak in the original. CFF/TWB did fix a *Grendel fear* gene that wasn’t working [75]; we ensure it’s active.

Learning Adjustments: We might adjust the global **learning rate (reinforcement susceptibility)** for certain tracts. If the genome allows, increasing the *“susceptibility”* of decision neurons to reward might make learning faster. However, too high and they could overcorrect behaviors. We’ll likely use standard CFF values here, as those were tuned for better learning without erratic swings.

Finally, all these instincts and behaviors will be checked for **compatibility with our tools**. The GEN Patcher’s CFF module will automatically add many of the above (friend/foe, hunger overwhelmsion, etc.), and its behavior packs can inject some of these behaviors dynamically. For example, the **“Ambition++” (Goal-Seeking) pack** [84] likely increases the tendency of creatures to act to satisfy drives rather than remaining idle. We will use Ambition++ to ensure our Norns don’t laze around – they will actively pursue what they need (food, mates, toys) with higher frequency and focus. Similarly, if we want a bit of aggression (maybe to ensure they compete or defend themselves), the *Khan Mode* could be applied but probably at a low level for Norns (we don’t want them harming each other too much; that’s more for Grendel design).

To summarize, our genome design will produce Norns that **innately know how to survive and reproduce**: they're born with the knowledge to eat when hungry, seek others when lonely, and flee danger when scared. Then, their enhanced brains and well-tuned chemistry allow them to build on those instincts – learning more quickly and remembering more. The end result should be a population of Norns that are noticeably more intelligent (solving their needs in fewer tries, exploring more of the world, recognizing friend from foe) and more prolific (reaching reproductive maturity at a healthy pace, actively seeking mates, and having many offspring with fewer losses).

Implementation Steps

Designing the genome is half the battle; implementing and verifying it is the other half. Here is our step-by-step plan to turn the above design into reality:

Step 1: Set a Base Genome: We will begin with a known stable genome as a starting template. The **CFF 1.1 ChiChi Norn genome** (the improved version of the standard Docking Station Norn) is an ideal foundation, as it already contains many fixes (friend-or-foe lobe, hunger organ, etc.) [85] [86]. Starting from CFF ensures we don't have to re-implement all those baseline fixes from scratch. Alternatively, we could start from the 2017 genome or TWB genome if those are available – but CFF is widely used and well-understood, making cross-breeding easier (note: mixing radically different base genomes can lead to infertile offspring [87], so we'll stick to one base for the whole world to maintain compatibility).

Step 2: Apply Automated Patches with GEN Patcher: Using **GEN Patcher 4.6.4**, we load the base genome `.gen` file. First, we run the **"One-Click Pipeline"**, which according to the interface will *"Apply Donor-less CFF → Behavior Packs → Scale to target"* automatically [88]. We will configure the pipeline options as follows: - Enable **CFF Safe Fixes**: check the options to *Add CFF instincts & stimuli*, *Add hunger emergency*, etc., as these will import the CFF changes in case our base lacked any [89]. (If starting from an already CFF genome, this may just ensure nothing is missing or could be skipped to avoid duplication – but it's safe to apply as it won't double-add genes due to donor-less logic.) - Check **Pro-breeding boost (mild)** while leaving "Import libido-lowerers" off (since we want short female recovery, we won't import the extended libido-lowerer from original CFF) [90]. This will apply a slight boost to reproductive drive as noted. - Set **Behavior Pack Sliders**: We'll dial *Ambition*++ to a moderate value (e.g. 140 out of 300, which is the default the tool shows [91]) to encourage goal-seeking, and *Libido*++ to around 150 (default) or maybe a bit higher if testing shows they still aren't mating enough [92]. *Khan Mode* we can leave at default low (120) or zero, since Norns aren't meant to be very aggressive [93]. These values can later be fine-tuned by re-patching if needed – one advantage of this approach is we can iteratively adjust and reapply. - Set **Target Neurons** to the desired count. We might start with 2400 as a conservative increase (which is already almost 3× the original ~900) [66]. We click *"Apply Augment (brain)"* to let GEN Patcher resize the lobes. Then perhaps try *"Max to grid"* which fills the brain grid to the 3072 maximum [56], and use the Brain Viewer to inspect the layout. We will verify no two lobes overlap – if the auto-scaler placed lobes too tightly, we might manually adjust a few positions in the Lobe Editor table (for example, separating any that ended up adjacent to avoid interference) [36]. We then ensure to hit "Apply" in the Lobe Editor for any manual tweaks so they reflect in the genome data. - Use the **Render Brain** button to generate a fresh visualization [35]. This should now show an image of the brain's 64×48 grid with differently colored rectangles for each lobe, allowing us to see the new neuron allocations. We check that the friend/foe lobe (if added) and fullness lobe appear on this map (they should, if the CFF patch added them, likely as small lobes in unused areas of the grid). The tract viewer table will list each tract gene; we scroll it to confirm key tracts (Drive->Decision, etc.) are present [63]. We might note the offsets and lengths for advanced reference, but mainly we want to ensure none say “0 length” (which would indicate a broken connection).

Step 3: Manual Genome Editing: Next, for any fine adjustments not covered by one-click: - Open the genome in the **Genetics Kit** or an editor like liveGMS. We will edit specific genes such as Half-lives: increase *Chemical X* half-life (e.g. Life chemical) by selecting the Half-life gene and editing the entry for that chemical. If the Genetics Kit displays time in minutes, we convert our desired lifespan to the appropriate value (the Discover Albia resource has tables mapping half-life values to real hours [94]). - Edit Receptor/Emitter genes: for example, find the *Reward* emitter gene and slightly raise its emission quantity, or set its threshold lower so it triggers a bit more easily. Likewise, ensure the *Punishment* emitter isn't too high. We can also adjust the *thresholds* of drive receptors that control when certain behaviors kick in (e.g. at what hunger level the hunger overwhelm organ triggers). - Verify new genes added: The CFF patch likely added about a dozen genes; we use the Genetics Kit's gene list (or the .gno notes file) to identify them (they might be marked with author comments). We particularly check the *Instincts* section: ensure instincts like “eat food when hungry” exist for all food types (fruit, seed, etc.), and add any missing ones. For instance, if we want a “approach mate when lonely and fertile” instinct, we can add a new Instinct gene: set the stimulus to *pheromone smell*, action to *approach* creature, and give it a reward. The Genetics Kit makes adding an instinct straightforward via drop-downs for stimuli and actions. - Increase **Brain plasticity** if possible: In the Dendrite genes, there are parameters for *learning rate* and *pregnancy impact on brain*. We could lower the penalty for pregnancy on learning (if any) so pregnant females remain as sharp as males mentally. Also, some genomes include *“Brain training rewards”* – e.g. rewarding the Norn for paying attention to the hand. We can insert an emitter gene to release a bit of reward when the Norn looks at the Hand and hears a tickle, reinforcing human teaching. - Save the modified genome with a new name (e.g. “SuperNorn.gen”). The Genetics Kit will compile it, checking for errors.

Step 4: Hatch Test Subjects: We will use the **Egg Tools** in GEN Patcher or a manual method to hatch some test Norns with the new genome. GEN Patcher can generate a **.cos file** that, when injected via the CAOS Command Line or Magic Words, will create an egg of our breed [95] . We might also build a **PRAY .agent egg** for convenience, using GEN Patcher's *Build .agent (EGGS)* option [96] to package the genome, sprites (we'll just reuse an existing sprite set like ChiChi Norn), and an egg laying script. We then inject this agent in Docking Station, which should place a new egg of our breed in the incubator. We hatch a few (at least one male and one female) to begin evaluations.

Step 5: Verify Basic Functionality: Once hatched, we immediately check a few things: - Using the DS **Science Kit**, open the **Genetics (dDNA) tab** for our creature and scan that all our genes are present and expressed. This is a quick way to ensure nothing is catastrophically wrong (e.g. a lobe gene failing to express would show a missing brain region). - Check the **Health Kit**: all organs should be green (no organ is dead on arrival), and life force ~100%. If any organ shows failure, we investigate which gene might be mis-specified. - With the Neuroscience Kit's **Brain Activity** window [40] , confirm neurons are firing and the creature isn't brain-dead. We can poke a neuron in the Perception lobe (the kit allows firing a neuron) and see if it propagates expectedly to concept or decision lobes. If our brain enlargement or new lobes caused an issue (like a lobe not updating), we might see a blank lobe or abnormal behavior here. Given we followed known parameters, this is unlikely, but it's a necessary sanity check.

Step 6: Behavioral Testing: Now the fun part – observe the creatures in various scenarios to see if our design goals are met: - **Learning test:** We present a simple challenge, e.g. put the Norn in front of a food vendor when it's hungry. Does it figure out to press the vendor faster than a normal Norn would? Ideally, with the hunger drive high and our strong "press food dispenser" instinct, it should perhaps do it on first try or with minimal coaching. We can measure this by counting how many tickles (positive reinforcement from the Hand) it needs compared to an unmodified Norn. - **Memory test:** We introduce a friendly Norn and a nasty Grendel. Our Norn should approach the Norn (due to social drive and friend memory) and flee or fight the Grendel. We can even let the Grendel hit our Norn once – the **ForF lobe** neuron for that Grendel should light up negative. Later, if the Grendel comes near again, we expect to see the creature's fear response engage before it even gets hit (memory of foe). We watch for adrenaline and fear chemical spikes on the Biochemistry Panel when the Grendel appears [21] . If that works, it's a success: the Norn recalls a foe. - **Reproduction test:** We monitor a male-female pair in an enclosed area with ample food. With elevated libido and social instincts, they should progress to kissing and conceiving on their own relatively quickly (perhaps as soon as the female is fertile). We will use the **Breeder's Kit Reproductive monitor** [47] to see each creature's **fertility and sex drive** levels. We want to see: sex drive rises, they seek each other, "kisspop" happens, and then one becomes pregnant. If nothing happens for a while, we may need to adjust libido or ensure the "pheromone emitter" gene is functioning. If the male seems willing but the female not, perhaps our instinct for female approach needs boosting. These observations will guide minor tweaks. - **Lifespan and energy:** We accelerate time (or just use the **Wolf Control** if available for hands-off running) to see that our Norns can live to breed multiple times. Do they age gracefully, or do they all die of hypoglycemia from that big brain? If we see many sudden deaths in youth, that's a red flag – likely ATP deficiency in the brain organ. We'd address that by increasing glucose -> ATP conversion or reducing brain update frequency slightly. If they live long but then maybe get illness, we check if our immune tweaks worked (for instance, a Death due to organ failure might show in logs – if heavy metals or antigen killed them, we might incorporate a missing antidote or vitamin gene). The **HoverDoc** or X-Ray could be used if available to inspect internal states periodically. - Throughout, we also ensure **no unintended behaviors** emerged: e.g., sometimes boosting one drive too much can cause obsessive behavior (a Norn that mates but forgets to eat, or vice versa). With our hunger overwhelm in place, they shouldn't starve from mating obsession, but we watch to be sure they balance activities (this is where the *Ambition++* tuning matters: it helps them cycle through needs).

Step 7: Iteration: Based on testing, we loop back. Using GEN Patcher and/or Genetics Kit, we adjust any parameters that were off: - If they aren't as smart as hoped (say they still hit a wall repeatedly), we might further enlarge the concept lobe or adjust neuron gain on pain->decision pathways so that "wall equals pain" is learned more strongly. Or perhaps add an instinct "if hit wall, turn around" to give a hint. - If reproduction is lacking, increase the *Libido++* slider and maybe shorten the refractory period (lower the "sexual exhaustion" chemical longevity). - If they overbreed to the point of harm, we might *reintroduce a bit of brake*: e.g., ensure the female's **pregnancy hormone** actually suppresses her drive until she lays the egg (so she doesn't immediately conceive again within seconds). Also ensure *laying an egg injects some exhaustion* so she rests a moment (the game normally does this via lactate and recovery drives). - We also ensure **cross-compatibility**: if these Norns are to live with other breeds (Ettins, etc.), do our changes cause any issues? They should be fine, but if a feature like the friend/foe lobe only Norns have, Ettins won't have that lobe – that's okay; the Norn might still treat an Ettin neutrally/default as "friend" or "foe" based on species smell. This is more of a note that we wouldn't breed our genome with non-CFF genomes to avoid infertile hybrids [87] .

Step 8: Finalize and Distribute: Once satisfied, we use GEN Patcher's **Export tools** to compile the final genome into distributable form. We will create: - A **GEN file** (for advanced users who want to directly hatch via Genetics Kit). - A **breed agent** (`.agents`) that includes our genome and uses the PRAY template to inject eggs. GEN Patcher can build a *Full Breed Pack* agent, which would include not just eggs but also all necessary sprites and body data [38]. Since we are making a genome only (no new sprites; likely using an existing Norn sprite set like Bruin or ChiChi), we can make a *DSAG injector* agent that simply injects our genome as eggs into the world [96]. We test the agent in a clean Docking Station install to ensure it correctly spawns creatures.

We'll document the differences of our genome (perhaps using **gendiff** to list changes from CFF base) so other researchers can understand what was done. This also helps ensure we didn't accidentally omit a fix: if gendiff shows we removed a CFF gene unintentionally, we'd catch it.

Recommended Next Actions

With our "Intelligent Prolific Norn" genome implemented, there are several follow-up steps and possibilities:

- **Population Testing (Wolfling Runs):** We should run a long-term *wolfling run* (autonomous population) using this genome to see how they fare over many generations. This will test the stability of mutations: Did our expansions lead to any mutational hot-spots (e.g. a tendency for a certain lobe to mutate out of bounds)? Given the SVRules design, it's unlikely to crash, but behaviors could drift. We might discover that after 10 generations, they have become too aggressive or too docile, etc. If so, we might adjust mutation rates or make certain genes **immutable** (CFF already set some critical genes like pose and gait to immutable to prevent degeneration [97]). We can designate our new lobes as immutable if mutations there would cause issues, though ideally we allow some evolution.

- **Incorporate** TWB Features**: Our current focus was intelligence and breeding. The** True Warmblood (TWB) genome **improvements (thermal regulation, dynamic blood chemistry, etc.) can be integrated as a phase two. This could further improve survivability and open the door to more complex behaviors (like seeking warmth or producing vitamin C from fruit [98]). Since TWB v3 is largely compatible with CFF (it was built on CFF) [99], we could use GEN Patcher to apply TWB-specific genes or use the TWB genome as a donor for things like the** immune system overhaul** (which adds realistic antibody production). This would make our Norns not just smart and fertile, but also hardy against disease – an all-around superior creature.

- **Cross-Species Application:** The design principles here could be applied to Grendels and Ettins as well. Grendels could have an enlarged brain for cunning and a high aggression (perhaps using that Khan Mode at max for a truly dangerous predator), whereas Ettins might be made more curious and less hoarding. The GEN Patcher can patch those genomes similarly. We recommend creating equivalent *Super Grendel* and *Super Ettin* genomes so that all species in the world are balanced with advanced capabilities. Otherwise, our super Norns might overrun a world if Ettins remain stupid or Grendels remain easy to outsmart – though that's a fun experiment in its own right.

- **Monitoring and Tweaking:** Even after release, we encourage monitoring by players. Using the Neuroscience Kit and Biochemistry Kit in real player worlds can help catch any odd behaviors. For example, if many players report that these Norns tend to all crowd near the incubator and not spread out, we might introduce a stronger *"exploration"* drive or agent to encourage dispersion (sometimes a custom **"Navigator" agent** can periodically push creatures to explore). Our genome gives them tools, but environment design (metarooms with proper food distribution, etc.) also matters. So, recommended: along with the genome, provide guidance on ideal world setup (e.g.

ensure toys and food are available to satisfy their higher activity, maybe use a **population control agent** if needed since they breed a lot).

- **Genetic Documentation:** As a next action, produce a full documentation of the genome: list of lobes, neuron counts, chemicals, and new instincts. This helps the community understand and perhaps improve further. For instance, posting the **full gene list (in .gno format)** on the Creatures Wiki or forums would allow others to suggest refinements. Perhaps some community members will run *IQ tests* (there are scripted IQ tests in the community, like getting Norns to navigate a maze for food) to quantitatively measure intelligence gains.

- **Compatibility with OpenC2E:** We should also test the genome in the open-source engine **OpenC2E** (if available in a usable state). OpenC2E aims to replicate the Creatures engine, and in some cases it might lift certain limits or at least help debug (it has its own brain viewer in development). If OpenC2E supports larger brains or multi-threading, our super Norns could truly shine there. It's worth ensuring that nothing in our genome exploits a quirk of the original engine that might differ in OpenC2E. If differences are found, we could create slight variant genomes for OpenC2E as needed.

- **Further Enhancement Ideas:** Future research could experiment with **new drives or needs** for even richer behavior. The engine allows up to 20 drives; we mostly use the standard ones, but one could, for example, introduce a **"Curiosity" drive** that goes up when a Norn hasn't learned something new in a while, pushing it to experiment. This is speculative, but our expanded brain might handle it. Also, utilizing unused chemicals, one could simulate, say, **"hormonal cycles"** to drive more complex social structures (imagine Norns that have a periodic mating season influenced by a chemical rhythm). These go beyond current community mods, but with our strong foundation, such additions could be the next frontier.

- **Ensure GEN Patcher Ongoing Support:** Since we heavily rely on GEN Patcher, we will collaborate with its maintainer if needed to report any issues encountered during our process. For instance, if we found that patching something caused a subtle bug, that feedback can improve the tool. The fact that our plan stays within GEN Patcher's feature set (v4.6.4 Lite's capabilities) means others can easily recreate or modify our genome using the same tool, which is beneficial for community uptake.

In conclusion, by combining the proven fixes from CFF/TWB with targeted enhancements for brain size, biochemistry, and behavior, we expect to create Norns that are **significantly more intelligent and prolific** than the originals – yet still balanced and *Creaturey*, not veering into "unrealistic" territory. Through careful testing and iteration, and with the rich toolset at our disposal [31] [24], this genome redesign can be successfully implemented. The next steps involve community testing and iterative refinement, but the groundwork laid out here provides a clear path to a new generation of Norns that truly push the limits of the Creatures simulation engine.

Sources:

- Creatures Wiki – *Brain anatomy and mutation design* [100] [101]; *Biochemical system overview* [15] [23]; *C3/DS genome enhancements (CFF, etc.)* [86] [69].
- Creatures Caves – *Official tool descriptions* (Genetics Kit, Biochemistry Kit, Neuroscience Kit) [43] [102]; *Community tool listings (GEN Patcher, GMS editors)* [31] [49].

- GEN Patcher v4.6.4 Lite – *Interface features for brain scaling and CFF integration* [66] [103]; *Behavior pack sliders and application* [33] [104]; *Brain viewer and lobe editor usage* [35] [36].

---

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [20] [21] [22] [26] [55] [79] [100] [101] Brain - Creatures Wiki
https://creatures.wiki/Brain

[14] [15] [16] [17] [18] [19] [23] [94] Biochemistry - Creatures Wiki
https://creatures.wiki/Biochemistry

[24] [25] [39] [40] [41] [44] [45] [46] [47] [102] Creatures Caves | Community | Resources
https://creaturescaves.com/community.php?section=Resources&view=53

[27] Genetics - Creatures Wiki
https://creatures.wiki/Genetics

[28] [29] [30] Genetics Kit - Creatures Wiki
https://creatures.wiki/Genetics_Kit

[31] [32] [33] [34] [35] [36] [37] [38] [56] [63] [66] [71] [84] [88] [89] [90] [91] [92] [93] [95] [96] [103] [104] gen-patcher-augment-v4.6.4-lite.html
file://file-78BEJevkcNC8cy7YBFVvvK

[42] [43] [48] [51] [53] Creatures Caves | Community | Resources
https://www.creaturescaves.com/community.php?section=Resources&view=102

[49] [50] [52] [54] Creatures Caves | Community | Resources
https://www.creaturescaves.com/community.php?section=Resources&view=14

[57] [58] [59] [60] [61] [62] [64] [65] [67] [68] [69] [70] [73] [74] [75] [76] [77] [78] [80] [81] [82] [85] [86] [87] [97] [98] [99] Base Genome - Creatures Wiki
https://creatures.wiki/Base_Genome

[72] [PDF] Docking Station
https://shared.fastly.steamstatic.com/store_item_assets/steam/apps/1659050/manuals/DSManual_final.pdf?t=1721031581

[83] C3/DS Genetic Breeds - The Realm
http://grendelman.blogspot.com/p/c3ds-genetic-breeds_10.html