01 | Randomizing Language Features in Tamgwa

by **armytag**

When and how to roll the dice

Insert an introduction here. 250-800 characters, including spaces.

Why use randomization?

At first glance, randomization might seem at odds with the very spirit of conlanging. Whether we make languages to flesh out our fictional worlds, to explore the edges of linguistic and philosophical expression, to further the noble cause of international communication, or simply for the glee this process brings us, we never want what we do to be a random mess. There is almost always a *purpose* behind our endeavor, an idea pushing forward all the effort we inevitably dedicate to this pursuit. If all we cared about were the phonologies, inflection tables, and word lists, we might spend a few hours with Vulgarlang or ChatGPT seeing what combinations they spit out.[1] But that kind of activity is never going to sustain our interest for very long, certainly not for the months and years that many conlangers put into their creations. Clearly there is something to this process fuels the passion of a conlanger, which randomization along can never fulfill.

So is there any place for randomization in the conlanging process? The answer, of course, is yes! Randomization works best as a tool in a conlanger's toolbelt, augmenting their creation process without dominating it. This can happen in several ways. First, randomization can help a conlanger generate content like phonemes, words, grammar structures, and more. This benefit should be obvious to anyone who winces remembering the half-finished dictionary they haven't touched in months. Second, randomization can counter a conlanger's linguistic biases. Few of us are familiar with more than a handful of languages, and that limited exposure can likewise limit the form our conlangs take. If you can't help but make Englishy conlangs, chance can steer you elsewhere. Lastly, randomization can help overcome the indecision we sometimes feel in the face of endless linguistic possibility. There

[1] If this actually is what you're interested in, I recommend looking at real languages anyway. The diversity in natural languages far outshines anything these tools can offer.

are so many ways that human language conveys ideas, and sometimes it can be almost impossible to wade through the myriad options to settle on the **one** this language will use.

All these situations and more are excellent times to use a bit of randomization. Just remember that randomization is only a tool to assist with the actual goal: creating a language.

Laying the foundation

Now that I've convinced you of the merits of a bit of randomization, you still need to know how to use it. To that end, I will walk through the process I used for starting **Tamgwa** with almost entirely randomized features. The simplest way to randomize a feature is to list a few variations of how that feature could look, and then toss a coin, roll some dice, or otherwise randomly choose which variation to move forward with. For example, we could take the list of the types of writing systems from Wikipedia (abjad, abugida, alphabetical, and logographic/syllabic) and roll a 4-sided die to decide which system to implement. The result might be at odds with the structure or phonology of the language, such as a syllabary for language with complex syllable structure, but that is precisely how randomization challenges our biases in language construction. We can always put the random result aside if it simply isn't working with the rest of the language.

For **Tamgwa**, rather than using Wikipedia I instead used the invaluable World Atlas of Langauge Structures (WALS).[2] WALS has a collection of chapters and associated maps showing the distribution of a variety of language features, from phonology to morphology to syntax and more. Each chapter lists various realizations of the language feature along with the number of languages using that variation. These numbers give an idea of how widespread each variation is and allow us to weight the randomization in a more naturalistic way. In other words, we can use the chapters in WALS to randomize an array of language features in a way that mimics the tendencies of natural languages.

To illustrate this process, we'll randomize how **Tamgwa** encodes noun plurality using WALS chapters 33 & 34. Chapter 33 covers how plurality is encoded, such as using affixes like English's *-s*. The authors list the following possibilities and number of languages which employ that option.

Language Feature	Language Count	Randomizer Range
Plural prefix	**126**	**1 - 126**
Plural suffix	**513**	**127 - 639**
Plural stem change	**6**	**640 - 645**
Plural tone	**4**	**646 - 649**
Plural by complete stem reduplication	**8**	**650 - 657**
Morphological, but no method primary	**60**	**658 - 717**
Plural word	**170**	**718 - 887**
Plural clitic	**81**	**888 - 968**
No plural	**98**	**969 - 1066**

Table 01.1: WALS Chapter 33: Coding of Nominal Plurality

I have added the Randomizer Range column, which simply shows how our random number should map back to the feature options.[3] Now all we have to do is generate a number

[2] https://wals.info

[3] To create the Randomizer Ranges yourself, simply take the high end of the previous range, add 1 to it to get the low end of the new range, and add the number of languages to it to get the high end of the new range

between 1 and 1066 (inclusive) and see which range it falls within. For **Tamgwa** I generated the number 814, so **Tamgwa** will encode plurality with an independant word rather than some kind of morphological inflection. We can use the same process on Chapter 34: Occurrence of Nominal Plurality to randomize *when* this plural word should be used. Hopefully we don't randomly select the option "No nominal plural", since we've already established that **Tamgwa** will use a plural word; we could either not include this option or re-randomize if we get it. Cutting to the end, I ultimately generated the random number 118 for this feature which corresponds to the option "Plural in all nouns, always optional". Thus in **Tamgwa** all nouns can be optionally marked for plurality by adding a plural word somewhere in the sentence.

For **Tamgwa** I used this process to randomize features for most of the WALS chapters. This provides a wide grammatical basis upon which to build the rest of the language. I've listed some of the randomized features below. Of course some contradictions are likely to arise from these completely random features, which can be seen in this list as well. These contradictions are not always obstacles however, and they can sometimes lead to developing interesting new features to resolve them. I will go into an example of this next.

- 1
- 2

Personal disagreement

With features randomized for most of the WALS chapters, there was a surprising level of consistency across the features. **Tamgwa** is shaping up to be an analytic language with very little inflectional morphology. However, there's a glaring contradiction among these features: verbs have "little to no inflectional morphology" (chapter 26) but verbs also have "person marking of both A and P arguments" (chapter 102), in other words polypersonal agreement. The definition for minimal inflection in Chapter 26 is that verbs don't have an affix index above 2, with person marking for A adding 2 points and person marking of P adding 1 point to the affix index. If we marked both A and P with simple affixes, the affix index would be 3 and thus violate this feature.

A simple solution might be to throw out one or the other of these features and either randomly generate a new value or simply choose one myself. However, there are some details in the WALS definitions that can also help us resolve this contradiction. In Chapter 26, Prefixing vs Suffixing in Inflectional Morphology, the authors clarify that for verbal person markers they did not include clitics that can also attach to non-verbs as part of the affix index calculation. On the other hand Chapter 102, Verbal Person Marking, does include clitics as part of the analysis as long as the clitics sometimes attach to the verb. Therefore, if at least one of the person markers is a clitic then the affix index won't go above 2 and both of the features can be satisfied.

In order to implement this in **Tamgwa**, it would be nice to have an example of how this works in real-world languages. A quick internet search produced the paper *The clitic status of person markers in Sorani Kurdish*(Gharib & Pye 2023).[4] This paper goes into depth about person markers in the Sorani dialect of Kurdish, where one of the persons is marked by an affix on the verb and the other person is marked by a clitic on either the verb or a preceding adverb. Whether the clitic is marking the agent (A) or the patient (P) of the action depends on whether it is using the Perfective or Imperfective aspect.

[4]Gharib, H., & Pye, C. (2023) The clitic status of person markers in Sorani Kurdish. Kansas Working Papers in Linguistics, 39, 57-65. https://doi.org/10.17161/1808.27692

(1) **dat** **benim** Sorani Kurdish
da =t beni -m
IPFV =OBL.2SG *see*.IPFV -1SG
"I see you."

(2) **benemeet**
beni =m -eet
see.PFV =OBL.1SG -2SG
"I saw you."

(3) **benetm**
bene =t -m
see.PFV =OBL.2SG -1SG
"You saw me." (Gharib & Pye 2023)

We can see that in Sorani the oblique clitic form is used to mark the patient of imperfective verbs and the agent of perfective verbs. This is extremely interesting, but perhaps too much to consider at this early stage of **Tamgwa**. For now we can keep things simple in **Tamgwa** and implement a basic clitic system that attaches to the first verb/adverb in the verb phrase regardless of aspect. Thus for our purposes, it is sufficient to see that examples (1) and (3) show the OBL.2SG clitic attaching to the aspect marker if it is present, and the verb if not. We can now implement this system in **Tamgwa** to satisfy our conflicting features.

Producing Pronouns

Now that we have an idea of how verbal person marking will work in **Tamgwa**, we're ready to pick some pronouns and put the system to the test. Of course this process will involve some randomization too! There are plenty of ways to generate a list of words and syllables, such as the online tool Vulgarlang[5], but I've written a simple Python script that will give me a certain number of words based on the phonology and phonotactics of **Tamgwa**.

Based on the features we generated at the beginning, **Tamgwa** will need indepent pronoun words with person-number stems including a clusivity distinction in 1PL pronouns. Since we will be marking agent and patient on the verb, we may as well make separate nominative and accusative forms of the pronouns as well. Finally, we will also indicate possession with a specially inflected pronoun, so we also need some kind of genetive form. Altogether that makes 7 person-number combinations and 5 forms (3 cases, 1 affix, and 1 clitic form) for a total of 35 distinct "words". After telling the Python script to generate 35 words we can put the results into a table to better visualize the pronouns we've just created.

	NOM	Affix	ACC	Clitic	GEN
1SG	**hnja**	**ki**	**gya**	**hnja**	**hma**
2SG	**lam**	**man**	**ba**	**be**	**pu**
3SG	**le**	**na**	**bwa**	**nji**	**pe**
1PL.INC	**ga**	**fe**	**hmuhl**	**pix**	**re**
1PL.EXC	**pwa**	**tenj**	**hla**	**le**	**pu**
2PL	**bihl**	**tu**	**bya**	**bes**	**hnjanj**
3PL	**xa**	**ti**	**pwax**	**njar**	**bwa**

Table 01.2: Random pronoun table

[5]https://www.vulgarlang.com

This table is, obviously, totally random and nonsensical. Keep in mind, though, that the randomization is merely meant to give us a starting point. From here we can look for patterns and rearrange the table to fit it into something more reasonable. Personally, I immediately latched onto the two **hnja** in the 1SG row and decided to move all the words beginning with **hnj** or **nj** into that row. From there we continue forming other patters, such as voiceless labial onsets for 2SG and closed syllables for GEN. The arrangement I finally settled on is printed below, with obligatory plural marker **tyan**[6] added to NOM and ACC forms of plural pronouns.

	NOM	Affix	ACC	Clitic	GEN
1SG	hnja	-nji	hnja	=njar	hnjanj
2SG	fe	-pe	pwax	=pwa	pix
3SG	bes	-hla	hmuhl	=hma	man
1PL.INC	tu tyan	-ti	re tyan	=na	tenj
1PL.EXC	xa tyan	-ki	gya tyan	=ga	bya
2PL	bwa tyan	-be	bwa tyan	=ba	bihl
3PL	pu tyan	-le	pu tyan	=le	lam

Table 01.3: Rearranged pronoun table

Revealing Results

Now that we have a full set of personal pronouns generated, we just need some preverbal tense-aspects words for the clitics to attach to, and then we'll be able to fully test our new syntax. I've decided to have **Tamgwa** use the same set of aspect words as described in Timothy Ajani's 2001 paper.[7] According to this analysis, we will need 4 aspect words marking Realis Imperfective, Irrealis Perfective, and Irrealis Imperfective, and Relational aspects. The Realis Perfective is unmarked and thus assumed when none of these explicit aspect words occur. The aspect words can co-occur to create more nuanced aspectual distinctions, such as Realis Imperfective and Irrealis Imperfective together denoting a Habitual aspect. For more details on how this works, please see Ajani's analysis on the original Yoruba system.

Unlike when we generated the whole set of personal pronouns and reorganized them afterward, when I'm creating just a handful of words I like to generate 2-3 times the number of words I need and manually pick from the resulting list. The results are listed in the table below.

Aspect	Word
RLS.PFV	<none>
RLS.IPFV	kwa
IRR.PFV	hna
IRR.IPFV	pi
REL	gya

Table 01.4: Aspect words in Tamgwa

[6] *N.B.* **tyan** is an optional plural marker for all nouns, but is only obligatory for plural pronouns. The feature was randomly generated for Chapter 35 of WALS.

[7] Ajani, T. 2001. *Aspect in Yoruba and Nigerian English*

With both the personal pronouns and aspect words, we can finally test how the person-marking clitic interacts with the verbal aspect system. We've decided that the clitics will attach at the end of the first word in the aspect-verb structure. Below are some examples of what that looks like.

(4) **kyamunjipwa** Tamgwa
kyamu -nji =pwa
see -1SG.NOM =2SG.ACC
"I saw you."

(5) **kwapwa** **kyamunji**
kwa =pwa kyamu -nji
RLS.IPFV =2SG.ACC *see* -1SG.NOM
"I see you."

(6) **pipwa** **kyamunji**
pi =pwa kyamu -nji
IRR.IPFV =2SG.ACC *see* -1SG.NOM
"I will see you."

(7) **pipwa** **kwa** **kyamunji**
pi =pwa kwa kyamu -nji
IRR.IPFV =2SG.ACC RLS.IPFV *see* -1SG.NOM
"I generally (habitually) see you."

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Conclusion

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.