

## Configurations

The user should initialize *cfg.nVaryingDimensions* to an integer between 0 and 17. See the “key variables” table for descriptions of each stim similarity level or the “Example Quaddles” table for image references.

Values ‘16’ and ‘17’ in *nVaryingDimensions* enable similarity score condition checking (control randomness mode), such that the Quaddles will become more similar and harder to distinguish. In general, as a *nVaryingDimensions* value increases, ease of distinguishing Quaddles decreases. Control randomness mode further increases similarity by ensuring that Quaddles share certain features.

## Tips to avoid errors

If you generated a set using a certain *num\_quaddles* value and now want to generate a set using a smaller *num\_quaddles* value than before, you may need to clear your *stimtxt\_path* folder to avoid issues with StimDef file writing and Quaddle generation.

Additionally, the user should be aware that the higher *num\_quaddles*, the longer the runtime will be. Quaddle generation takes a relatively long time, runtimes can be several minutes long.

*num\_quaddles* should be set to the desired number of Quaddles to be produced, but it should not exceed the maximum number of unique Quaddles able to be generated given the values in *nVaryingDimensions*. For example, if ‘1’ is in *nVaryingDimensions*, the user should not set *num\_quaddles* to be greater than 7, since there are only 7 available Quaddles to be generated if using just one feature dimension.

## Control randomness mode

A while loop is used to check the similarity score between all Quaddles and the first Quaddle in the set. An initial seed Quaddle sample is generated, then all subsequent Quaddles are added to the group of Quaddles in that set only if they pass the specified similarity condition in relation to the initial seed Quaddle sample. If *ss\_loBound* and *ss\_upBound* are both 0 for a given trial, then inter-Quaddle similarity will be randomly non-shared across all dimensions. These values will generate the most dissimilar set using this mechanism, as there will be no inherent cohesion between Quaddles. However, as similarity score bounds are set higher, the similarity between all Quaddles in the set increases. Since all Quaddles in a set need to maintain a certain level of similarity with the first Quaddle in the set when the specified similarity bounds are higher, the set of dimension values they can choose from is smaller, leading to increased similarity amongst the group as a whole. Each Quaddle is required to be of a certain similarity to the seed Quaddle, and therefore indirectly required to be of a certain similarity to all other Quaddles in its set.

## Control features mode

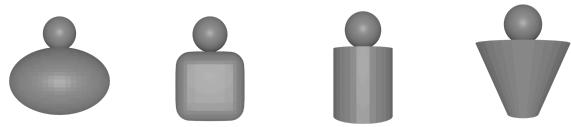
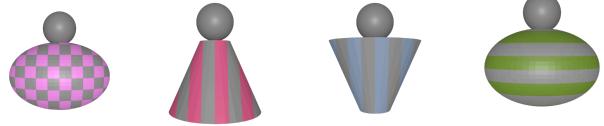
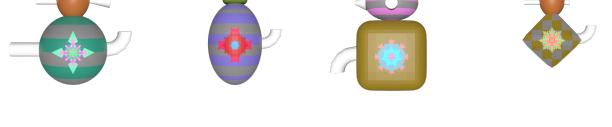
Higher similarity between Quaddles should be used in more difficult tasks, as it is more difficult to discriminate between similar Quaddles than dissimilar Quaddles. If the user requires Quaddles that are easier to discriminate than the random set of Quaddles produced by setting *nVaryingDimensions* for that set to 1-15, they activate the “control features mode.” Selecting low values in *nVaryingDimensions* allows for easy discrimination between stimuli. The number 1-15 corresponds to the number of varying feature dimensions. No similarity score conditions are enforced in this mode.

In control features mode, Quaddles are standardized in every dimension but those selected, allowing the user to restrict feature variance to a small number of dimensions and making discrimination between Quaddles simpler for the participant. The rationale behind having two modes for Quaddle generation is as follows: if highly distinguishable stimuli are required, it is better to limit the amount of varying features using control features mode; if highly similar stimuli are required, all features should be up for variation, so the user can enable control randomness mode and choose the extent of the variation.

## Quaddle 1.0 Mode

Because we wanted to provide access to Quaddles with the same dimensions that are offered by Quaddle 1.0, we assigned a special value of *nVaryingDimensions* to produce Quaddle 1.0-like stimuli. When *nVaryingDimensions* is set to 0, the program will produce Quaddles that vary in the four original dimensions: body shape, body color, pattern, and arms. These Quaddles do not have heads, fractals, ears, or beaks.

## Example Quaddles

nVarying Dimensions	Mode used	Similarity Specifications	Quaddles
0	“Quaddle 1” mode	quaddle 1 features only	
1	control features mode	body	
3	control features mode	body, primary/secondary body color, body pattern	
4	control features mode	body, primary/secondary body color, body pattern, head	
6	control features mode	body, primary/secondary body color, body pattern, primary/ secondary head color, head pattern	
15	control features mode	all features vary	
16	control randomness mode	all features vary; ss = [6.5, 8]	

17	control randomness mode	all features vary; ss = [9.25, 13]	
----	-------------------------	------------------------------------	--

## Key variables

Variable name	Description
<i>nVaryingDimensions</i>	Integer representing the number of feature dimensions that will vary in the Quaddles generated for that set. ‘1’ represents 1 varying dimensions, ‘2’ represents 2 varying dimensions, etc. ‘16’, and ‘17’ represent all 17 varying dimensions, but with similarity score bounds being completely random, [6.5 , 8], and [9.25, 13], respectively. Setting this to ‘0’ produces Quaddles that have the same dimensions as Quaddle 1.0.
<i>ss_loBound</i>	Integer representing current set’s similarity score lower bound. Relevant only when <i>nVaryingDimensions</i> of a given block is set to ‘16’ or ‘17’. The user may alter this value as desired.
<i>ss_upBound</i>	Integer representing current set’s similarity score upper bound. Relevant only when <i>nVaryingDimensions</i> of a given set is set to ‘16’ or ‘17’. The user may alter this value as desired.
<i>num_quaddles</i>	Number of Quaddles to be generated per set.
<i>gltf_files</i>	Intermediate struct used containing index and name of each Quaddle 2.0 generated in a session. Aids in writing to StimDef.
<i>samples</i>	List containing the dimension values of a Quaddle. Represents a single Quaddle.
<i>all_samples</i>	Matrix containing all accumulated <i>samples</i> in a set. Represents all Quaddles in a set.
<i>dimension_name</i>	Array of strings representing names of Quaddle dimensions (e.g. body, arm angle, head pattern, etc). There are 17 dimensions.
<i>dimension_value</i>	Dictionary representing each Quaddle dimension and the corresponding number of options for it.
<i>available_dimensions</i>	List containing all dimensions that are available to be varied. Different for each value of <i>nVaryingDimensions</i> .

## Key methods

Method name	Returns	Parameters	Description
<i>makeQuaddles2</i>	<i>cfg_out</i>	<i>cfg</i>	Creates <i>num_quaddles</i> Quaddles of specified similarity. Executes a loop to check similarity (calculated by <i>calculate_similarity_two_list</i> ) to the first sample in <i>all_samples</i> , calling <i>create_one_sample</i> to generate new <i>samples</i> with each iteration. Writes the <i>samples</i> to .txt files and calls <i>generate_object</i> to create Quaddles in Blender.
<i>init_dimensions</i>	<i>dimension_name,</i> <i>dimension_value,</i> <i>available_dimensions</i>	<i>controlRandomnessMode0_controlFeaturesMode1,</i> <i>nVaryingDimensions</i>	Initializes <i>dimension_name</i> , <i>dimension_value</i> , and <i>available_dimensions</i> . <i>Available_dimensions</i> will vary based on the value of <i>nVaryingDimensions</i> .
<i>create_one_sample</i>	<i>samples</i>	<i>dimension_name,</i> <i>dimension_value,</i> <i>available_dimensions</i>	Loops over all dimensions and randomly samples a value for each one to create a sample. Calls <i>modify_list</i> to help Blender with Quaddle generation.
<i>generate_object</i>	none	none	Reads sample .txt file and allows Blender to generate a corresponding .png, .gltf, or .fbx Quaddle.
<i>modify_list</i>	<i>modified_list</i>	<i>input_list</i>	function to help with similarity score and blender Quaddle generation
<i>calculate_similarity_two_list</i>	<i>score</i>	<i>list1, list2</i>	Compares two <i>samples</i> and outputs a similarity score. Similarity score is weighted based on feature relevance, but can be adjusted.