

*This sheet contains references to all relevant workshop-related enabling tools as well as how to use them. Please read the table of contents and the T2007 section below if you are beginning modeling.*

[illegible][illegible]



[illegible]

















[illegible]

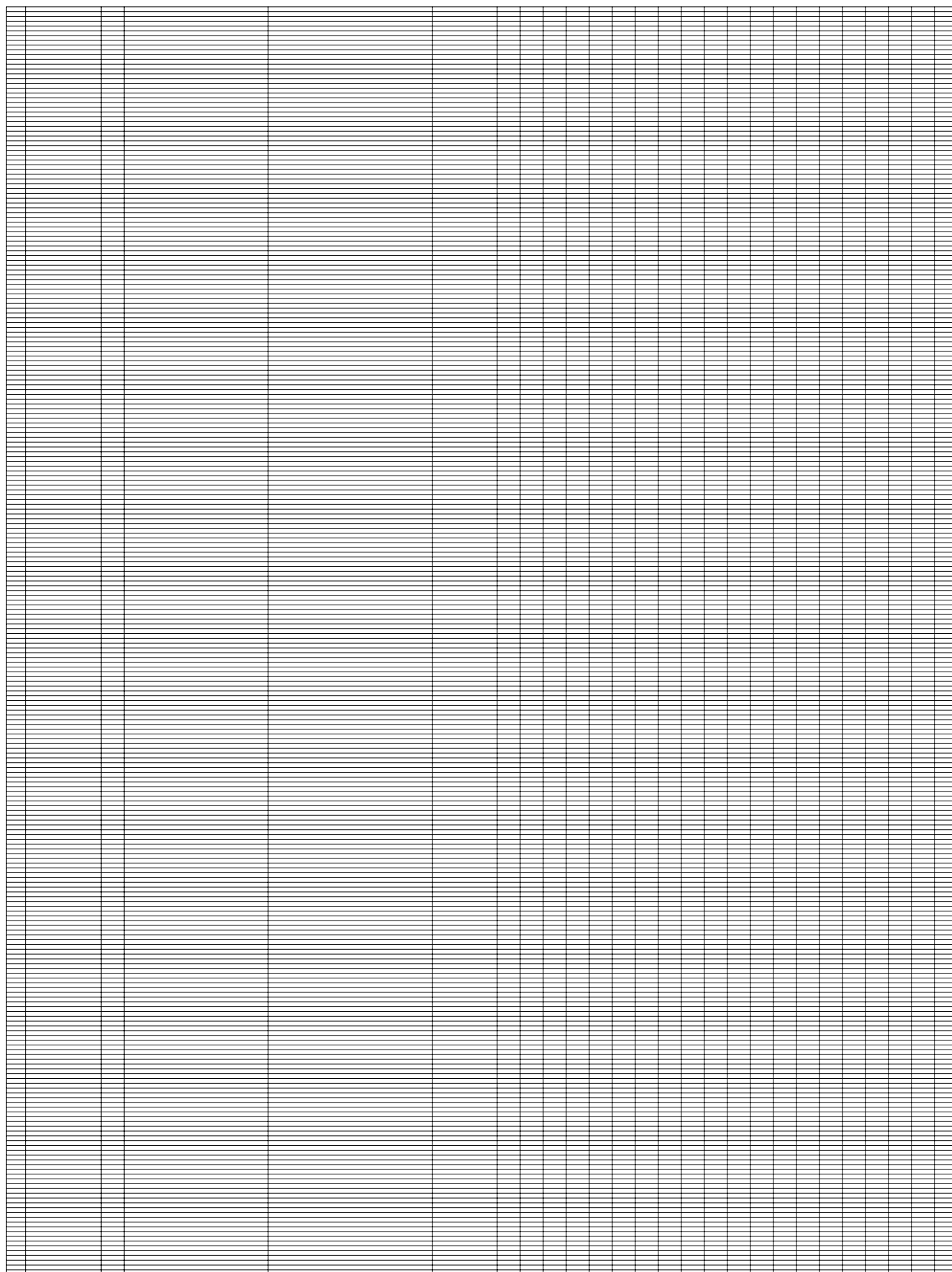












[illegible]

*This sheet is for an old version, see sheet tabs below.*

Might be missing updates try stable version: [https://docs.google.com/spreadsheets/d/7vEku\\_80](https://docs.google.com/spreadsheets/d/7vEku_80)

Might be missing updates try stable version: [https://docs.google.com/spreadsheets/d/7J3u\\_8DwgsTzB5oMS\\_IDUjHfhzZVESeSrkyrS4tnrk/edit#gid=412721476](https://docs.google.com/spreadsheets/d/7J3u_8DwgsTzB5oMS_IDUjHfhzZVESeSrkyrS4tnrk/edit#gid=412721476)

[illegible]



1. **Introduction**  
 2. **Background**  
 3. **Methodology**  
 4. **Results**  
 5. **Discussion**  
 6. **Conclusion**  
 7. **References**  
 8. **Appendix**  
 9. **Figure 1**  
 10. **Figure 2**  
 11. **Figure 3**  
 12. **Figure 4**  
 13. **Figure 5**  
 14. **Figure 6**  
 15. **Figure 7**  
 16. **Figure 8**  
 17. **Figure 9**  
 18. **Figure 10**  
 19. **Figure 11**  
 20. **Figure 12**  
 21. **Figure 13**  
 22. **Figure 14**  
 23. **Figure 15**  
 24. **Figure 16**  
 25. **Figure 17**  
 26. **Figure 18**  
 27. **Figure 19**  
 28. **Figure 20**  
 29. **Figure 21**  
 30. **Figure 22**  
 31. **Figure 23**  
 32. **Figure 24**  
 33. **Figure 25**  
 34. **Figure 26**  
 35. **Figure 27**  
 36. **Figure 28**  
 37. **Figure 29**  
 38. **Figure 30**  
 39. **Figure 31**  
 40. **Figure 32**  
 41. **Figure 33**  
 42. **Figure 34**  
 43. **Figure 35**  
 44. **Figure 36**  
 45. **Figure 37**  
 46. **Figure 38**  
 47. **Figure 39**  
 48. **Figure 40**  
 49. **Figure 41**  
 50. **Figure 42**  
 51. **Figure 43**  
 52. **Figure 44**  
 53. **Figure 45**  
 54. **Figure 46**  
 55. **Figure 47**  
 56. **Figure 48**  
 57. **Figure 49**  
 58. **Figure 50**  
 59. **Figure 51**  
 60. **Figure 52**  
 61. **Figure 53**  
 62. **Figure 54**  
 63. **Figure 55**  
 64. **Figure 56**  
 65. **Figure 57**  
 66. **Figure 58**  
 67. **Figure 59**  
 68. **Figure 60**  
 69. **Figure 61**  
 70. **Figure 62**  
 71. **Figure 63**  
 72. **Figure 64**  
 73. **Figure 65**  
 74. **Figure 66**  
 75. **Figure 67**  
 76. **Figure 68**  
 77. **Figure 69**  
 78. **Figure 70**  
 79. **Figure 71**  
 80. **Figure 72**  
 81. **Figure 73**  
 82. **Figure 74**  
 83. **Figure 75**  
 84. **Figure 76**  
 85. **Figure 77**  
 86. **Figure 78**  
 87. **Figure 79**  
 88. **Figure 80**  
 89. **Figure 81**  
 90. **Figure 82**  
 91. **Figure 83**  
 92. **Figure 84**  
 93. **Figure 85**  
 94. **Figure 86**  
 95. **Figure 87**  
 96. **Figure 88**  
 97. **Figure 89**  
 98. **Figure 90**  
 99. **Figure 91**  
 100. **Figure 92**  
 101. **Figure 93**  
 102. **Figure 94**  
 103. **Figure 95**  
 104. **Figure 96**  
 105. **Figure 97**  
 106. **Figure 98**  
 107. **Figure 99**  
 108. **Figure 100**  
 109. **Figure 101**  
 110. **Figure 102**  
 111. **Figure 103**  
 112. **Figure 104**  
 113. **Figure 105**  
 114. **Figure 106**  
 115. **Figure 107**  
 116. **Figure 108**  
 117. **Figure 109**  
 118. **Figure 110**  
 119. **Figure 111**  
 120. **Figure 112**  
 121. **Figure 113**  
 122. **Figure 114**  
 123. **Figure 115**  
 124. **Figure 116**  
 125. **Figure 117**  
 126. **Figure 118**  
 127. **Figure 119**  
 128. **Figure 120**  
 129. **Figure 121**  
 130. **Figure 122**  
 131. **Figure 123**  
 132. **Figure 124**  
 133. **Figure 125**  
 134. **Figure 126**  
 135. **Figure 127**  
 136. **Figure 128**  
 137. **Figure 129**  
 138. **Figure 130**  
 139. **Figure 131**  
 140. **Figure 132**  
 141. **Figure 133**  
 142. **Figure 134**  
 143. **Figure 135**  
 144. **Figure 136**  
 145. **Figure 137**  
 146. **Figure 138**  
 147. **Figure 139**  
 148. **Figure 140**  
 149. **Figure 141**  
 150. **Figure 142**  
 151. **Figure 143**  
 152. **Figure 144**  
 153. **Figure 145**  
 154. **Figure 146**  
 155. **Figure 147**  
 156. **Figure 148**  
 157. **Figure 149**  
 158. **Figure 150**  
 159. **Figure 151**  
 160. **Figure 152**  
 161. **Figure 153**  
 162. **Figure 154**  
 163. **Figure 155**  
 164. **Figure 156**  
 165. **Figure 157**  
 166. **Figure 158**  
 167. **Figure 159**  
 168. **Figure 160**  
 169. **Figure 161**  
 170. **Figure 162**  
 171. **Figure 163**  
 172. **Figure 164**  
 173. **Figure 165**  
 174. **Figure 166**  
 175. **Figure 167**  
 176. **Figure 168**  
 177. **Figure 169**  
 178. **Figure 170**  
 179. **Figure 171**  
 180. **Figure 172**  
 181. **Figure 173**  
 182. **Figure 174**  
 183. **Figure 175**  
 184. **Figure 176**  
 185. **Figure 177**  
 186. **Figure 178**  
 187. **Figure 179**  
 188. **Figure 180**  
 189. **Figure 181**  
 190. **Figure 182**  
 191. **Figure 183**  
 192. **Figure 184**  
 193. **Figure 185**  
 194. **Figure 186**  
 195. **Figure 187**  
 196. **Figure 188**  
 197. **Figure 189**  
 198. **Figure 190**  
 199. **Figure 191**  
 200. **Figure 192**  
 201. **Figure 193**  
 202. **Figure 194**  
 203. **Figure 195**  
 204. **Figure 196**  
 205. **Figure 197**  
 206. **Figure 198**  
 207. **Figure 199**  
 208. **Figure 200**  
 209. **Figure 201**  
 210. **Figure 202**  
 211. **Figure 203**  
 212. **Figure 204**  
 213. **Figure 205**  
 214. **Figure 206**  
 215. **Figure 207**  
 216. **Figure 208**  
 217. **Figure 209**





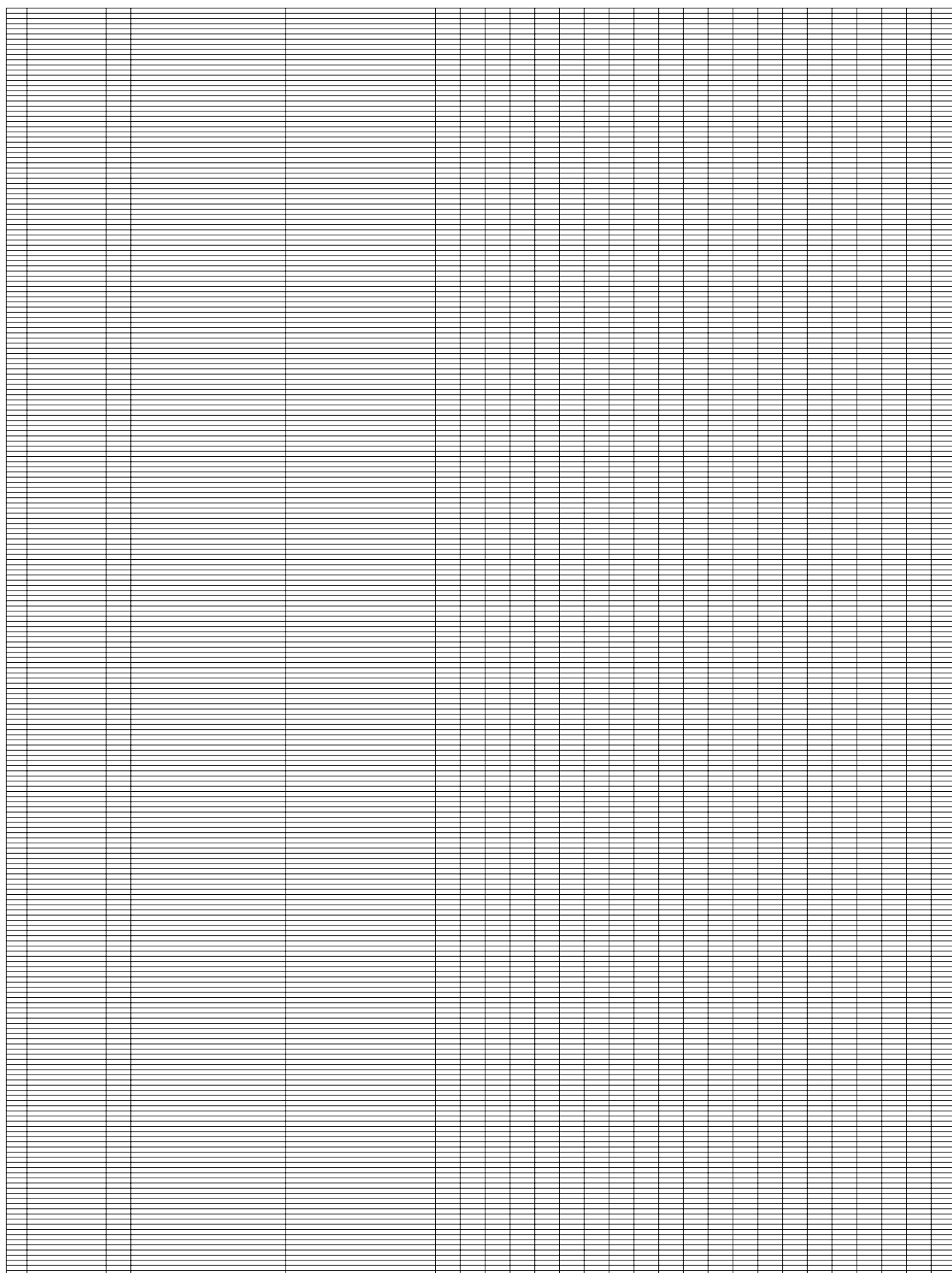




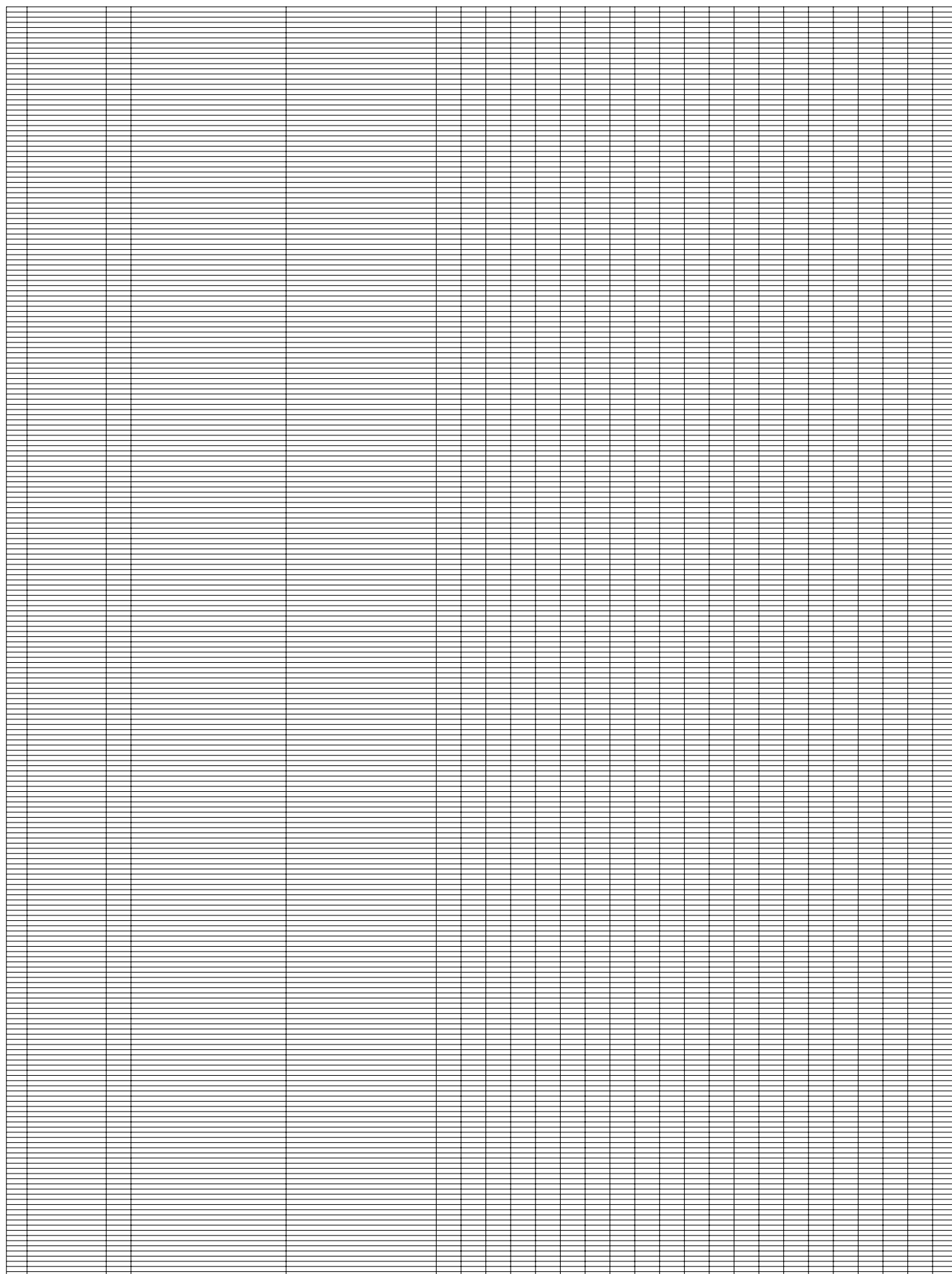












[illegible]

This sheet is for an old version, see sheet tabs below.

Section	[core]		
Code	Description	Example	Value Type
	<b>Common keys</b>		
name:	Defines the raw name for your unit, in which case the game uses to identify as a unique name. (This name is used for the unit's name in the game)	name: customTank_1	string
altNames:	comma separated list of names. Like name but lower priority, useful for multiple optional mods.	altNames: custTank1, customTank1, cTank1	string(s)
class:	Reserved for future use, must be CustomUnitMetadata	class: CustomUnitMetadata	string
price:	The cost of your unit from builders/buildings.	price: 500	int
mass:	The 'weight' of your unit, this will define how it collides with other units, a greater value means it is tougher	mass: 3000	int
techLevel:	Defines the Tech Level of your unit, there are 3 levels and each will appear in a different color in the GUI	techLevel: 1	int
buildSpeed:	Time it takes to build unit (may multiply with builder speed) Formulae for seconds: $10 \div 60 = 0.16$ (0.16 seconds)	buildSpeed: 0.16	float
radius:	Circular area around your unit that makes it selectable (mouse click/screen touch)	radius: 20	int
isBio:	Choose whether your unit is biological or not, will effect sound and splat (unless hideScorchMark is true)	isBio: false	bool
isBug:	Changes some death defaults, and sort order in sandbox		bool
isBuilder	Normally required if this unit places buildings. Defaults to [ai]useAsBuilder		
	<b>Unit stats</b>		
maxHp:	The max health for your unit. (will spawn with this value)	maxHp: 200	int
selfRegenRate:	Passive self repair rate		float
maxShield:	The max shield hitpoints of your unit. Can start with 0 hitpoints if startShieldAtZero is set to true	maxShield: 500	int
startShieldAtZero:	Sets the unit's shield to 0 HP when it is first built.	startShieldAtZero: true	bool
shieldRegen:	Passive shield regen rate	shieldRegen: 0.15	float
energyMax:	Defaults to 0. Energy can be used as ammo for turrets or for laser defense	energyMax: 1	float
energyRegen:			float
energyStartingPercentage:	Sets the percentage of charged energy when the unit is first built.	energyStartingPercentage: 0.5	float
energyNeedsToRechargeToFull:	Disable weapons using energy after reaching zero till fully recharged		float
armour	Damage taken away from each hit (not currently used in any core units)		
armourMinDamageToKeep	Min damage to keep. Defaults to 1		
generation_credits:	Income unit creates	generation_credits: 2	int
generation_delay:	How often generation_credits is added. Defaults to 40 (changing not recommended)	generation_delay: 40	int
	<b>UI and Graphics related keys</b>		
showInEditor:	Set to false to hide unit in sandbox editor (Default true)	showInEditor: true	bool
displayText:	The name of your unit that the game shows to the player.	displayText: Custom Tank	string
displayText_(LANG)	LANG=ISO 639-1 Code to show this text instead when game is in this language	displayText_es: Hola	
displayDescription	Description of your unit that the game shows to the player.	displayDescription: - Fast movement\n- Light damage.	string
displayDescription_(LANG)	LANG=ISO 639-1 Code to show this text instead when game is in this language	displayDescription_es: -Movimiento rapido\n- Daño ligero	
displayLocaleKey:	Translation file key for unit name and description	displayLocaleKey: units.mechArtillery	string
displayRadius:	Defaults to radius value. Set to show a larger or smaller selection box ui on units.	displayRadius: 20	int
shieldRenderRadius			
shieldDisplayOnlyDeflection:	Hide shield (if active) unless deflecting shot	shieldDisplayOnlyDeflection: false	bool
shieldDeflectionDisplayRate:	Defaults to 4. High value causes shield deflection to fade disappear faster	shieldDeflectionDisplayRate: 3	float
isUnselectable	Cannot be selected		
	<b>Building only keys</b>		
isBuilding:			bool
footprint	Only applies to buildings, tiles taken up which block unit movement. Defaults to 0,0,0,0 = 1 center tile	footprint: -1,-1,1,1	
constructionFootprint	Only applies to buildings, tiles taken up for placement of other buildings. Defaults to 0,0,0,0	constructionFootprint: -1,-1,1,3	
buildingSelectionOffset:	Defaults to 0. Adds or removes padding on the drawn selection rect in UI		int
buildingToFootprintOffsetX	Defaults to 10		float
buildingToFootprintOffsetY	Defaults to 10		float
placeOnlyOnResPool:	Normally used for extractors		bool
selfBuildRate:	Rate unit builds it's self when placed without a builder		float
	<b>Misc Keys</b>		
copyFrom:	Use unit data from another ini file as base	copyFrom: customTank_Default	file (ini)
dont_load:	Do not load unit, and don't error on missing data. Can be useful when used with copyFrom	dont_load: true	bool
overrideAndReplace:	Override another unit with this unit. Build links pointing to target unit will be replaced with this unit. And	overrideAndReplace: builder, combatEngineer	string(s)
globalScale:	Defaults to 1. Changing not recommend	globalScale: 1	float
isLocked:	Disallow building of this unit. Can be used with overrideAndReplace to restrict units the player is allowed to build	isLocked: false	bool
isLockedIfGameModeNoNuke	Disallows building of this unit if nukes are disabled during match setup.	isLockedIfGameModeNoNuke: false	
experimental:	Tag unit as experimental. Affects zoomed out icon and end game stats.	experimental: false	bool
stayNeutral:	Set to false to disable capture when unit is on the neutral team	stayNeutral: false	bool
createNeutral			
resourceRate:	Used with canReclaimResources. Experimental feature that is not yet finished.		float
tags	List of any comma separated strings. Only used for transportUnitsRequireTag right now, will be used for other things in future		
fogOfWarSightRange:	Set number of tiles this unit can see. Defaults to 15	fogOfWarSightRange: 15	int
softCollisionOnAll:			int
disableAllUnitCollisions			
isUnrepairableUnit	No unit can repair this unit		
isPickableStartingUnit	If true, unit is added to dropdowns for starting unit in game setup menus.		
startFallingWhenStartingUnit			
soundOnAttackOrder	List of sounds. Only one will be played each attack order		sound
soundOnMoveOrder	List of sounds. Only one will be played each move order		sound
canNotBeDirectlyAttacked	No unit can directly target this unit. If true this will also skip this unit in victory/defeat checks		
	<b>Transport related keys</b>		
transportSlotsNeeded:	Defaults to 1. Number of slots this unit uses up in a transport, Experimentals are often set to 5 to stop small transports holding them.		int
maxTransportingUnits	Number of slots this units as for transporting other units.		int
transportUnitsRequireTag	Only allow trasport of units that have one of these tags.		tags
transportUnitsRequireMovementType	Only allow trasport of units that have one of these movement types.		movementTypes
transportUnitsBlockAirAndWaterUnits	Defaults to true if transportUnitsRequireMovementType is empty		
transportUnitsKeepBuiltUnits	Makes built units stay inside transport instead of exiting it once ready		LogicBoolean
transportUnitsCanUnloadUnits	Defaults to: if not self.isOverLiquid() and not self.isMoving()		LogicBoolean
transportUnitsAddUnloadOption	Should unload button be added to the menu		LogicBoolean
transportUnitsKillOnDeath	Default true. If false transporting units don't die when transport dies	eg: if self.isOverLiquid()	LogicBoolean
transportUnitsHealBy	Rate to heal units being transported		float
	<b>Construction and factory related keys</b>		
canRepairBuildings:	isBuilder:true is required as well right now)		bool
canRepairUnits:	isBuilder:true is required as well right now)		bool
canReclaimResources:	Used with resourceRate. Experimental feature that is not yet finished.		bool
autoRepair:	Automatically try and repair damaged units in range ([ai]useAsBuilder:true is required as well right now)		bool
nanoRange:	Defaults to 85		int
nanoRepairSpeed:	Defaults to 0.2		float
nanoBuildSpeed:	Defaults to 1		float
nanoFactorySpeed:	Defaults to 1		float
extraBuildRangeWhenBuildingThis	Temporary add extra build range to builders to build this unit/building. Useful for water based buildings.		

builtFrom_#_name:	canBuild and builtFrom have the same effect just in the opposite direction. BuiltFrom is useful if adding new units to existing buildings		string	
builtFrom_#_pos:	Order action appears in UI.		string	
builtFrom_#_forceNano:	Build as if this was a building (even if it's a unit)		bool	
canBuild_#_name:	List of units this unit can create. Can be buildings or units. (isBuilder:true and canRepairBuildings:true is recommended when adding buildings)		units	
canBuild_#_pos:	Order action appears in UI		string	
canBuild_#_tech:	Tech level. Mostly just effects action colour in UI. Defaults to 1		int	
canBuild_#_forceNano:	Build target as if it was a building (even if it's a unit)		bool	
canBuild_#_isVisible:	Hide this build option if true			
canBuild_#_isLocked:	Dynamically lock this build option and show isLockedMessage if true.			
canBuild_#_isLockedMessage:	To tell the player why a unit is locked			
canBuild_#_isLockedMessage_(LANG):	Optional. For other languages.			
exit_x:	Where created or unloaded units appear. Defaults to 0	exit_x: 0	float	
exit_y:	Where created or unloaded units appear. Defaults to 5.	exit_x: 5	float	
exit_dirOffset : float	Defaults to 180 for units. 0 for buildings			1.13
exit_heightOffset : float	Defaults to 0			1.13
exit_moveAwayAmount : float	Defaults to 70			1.13
	<b>Death related keys</b>			
dieOnConstruct:	Delete this unit when it starts making a building. Target building likely will need selfBuildRate set to be created without a builder.		bool	
dieOnZeroEnergy	Kill this unit if energy level reaches zero.	dieOnZeroEnergy: false	bool	1.13
numBitsOnDeath	A display text when you select your unit's action, used to explain it's purpose. (All your sub actions will		int	
nukeOnDeath:			bool	
nukeOnDeathRange:			float	
nukeOnDeathDamage:			float	
nukeOnDeathDisableWhenNoNuke:	Defaults to false		bool	
fireTurretXAtSelfOnDeath				
explodeOnDeath	defaults to true		boolean	1.13
explodeOnDeathGroundCollision	defaults to true		boolean	1.13
effectOnDeath			effect ref	1.13
effectOnDeathGroundCollision			effect ref	1.13
unitsSpawnedOnDeath	unit list	unitsSpawnedOnDeath: tank*5, hoverTank	units	1.13
unitsSpawnedOnDeath_setToTeamOfLastAttacker			bool	
hideScorchMark:			bool	
	<b>Deprecated Keys (can be used but there are better ways)</b>			
action_#_convertTo:	Deprecated in 1.13, use [action_x] sections instead	action_1_convertTo: customTank_2	string	deprecated 1.13
action_#_pos:	Order action appears in UI		float	deprecated 1.13
action_#_price:	The price of your action for the unit. (All your sub actions will be linked to the # you use)	action_1_price: 1000	int	deprecated 1.13
action_#_text:	A display text when you select your unit's action, used to explain it's purpose. (All your sub actions will	action_1_text: Upgrade to Custom Tank 2	string	deprecated 1.13
action_#_description:			string	deprecated 1.13
action_#_addEnergy:	Adds energy to unit. Has no effect unless energyMax is set.		float	deprecated 1.13
action_#_whenBuilding_cannotMove:	Stops unit moving while action is being applied. Useful for deploy like actions.		bool	deprecated 1.13
<b>Section</b>	<b>[graphics]</b>			
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
image:	File path to png image.		file (image)	
image_back:	An optional image drawn behind other units. Useful for factories that units exit		file (image)	
image_wreak:	Image to use when unit dies. Can be NONE to leave no wreak		file (image)	
image_offsetX:			int	
image_offsetY:			int	
teamColorsUseHue:	False: Green pixels on unit gets converted to team color. True: Whole unit is tinted the team colour. Defaults to false		bool	
imageSmoothing:			bool	
scaleImagesTo:	Resize image to fit this value in pixels. Effects leg, and shadow images as well.		float	
imageScale:	Resize image. Defaults to 1. Effects leg, and shadow images as well.		float	
drawLayer:	Set to experimentals, or leave line out		string	
whenBeingBuiltMakeTransparentTill			float	1.13
icon_zoomed_out			file (image)	
	<b>Turrets (images can also be set on each turret)</b>			
image_turret:	Default image for all turrets, can also be set per turret		file (image)	
teamColorsOnTurret	Defaults false. Apply team colours on turret as well. Also effects pre-turret images		bool	
scaleTurretImagesTo:			float	
lock_body_rotation_with_main_turret:	Locks body image locked to first turret's direction		bool	
lock_leg_rotation_with_main_turret			bool	1.13
	<b>Shadow</b>			
image_shadow:	Image file, NONE, or AUTO. (AUTO will use image and make it transparent black only.)		file (image)	
shadowOffsetX:			float	
shadowOffsetY:			float	
lock_shadow_rotation_with_main_turre	Locks body image shadow locked to first turret's direction		bool	
	<b>Effects and animation</b>			
total_frames	Defaults to 1. Animations require this.		int	
frame_width:	Calculated for you if total frames is set, but can be overridden		int	
frame_height:	Defaults to image height		int	
splastEffect:	True to create a water wave effect when over water. Default false		bool	
dustEffect:	True to create a dust effect when over land. Default false		bool	
splastEffectReverse:	True to also create effect when unit is reversing		bool	
dustEffectReverse:	True to also create effect when unit is reversing		bool	
movementEffect	Custom movement effect, can be anything	eg. movementEffect: smoke, CUSTOM:fastDust*2, CUSTOM:pop*5	effect	1.13
movementEffectReverse			effect	1.13
movementEffectRate			float	1.13
movementEffectReverseFlipEffects	Create effect as if unit has rotated 180 when reversing		bool	1.13
repairEffect	Custom movement effect, can be anything. Replaces default effect from builders			
repairEffectAtTarget				
repairEffectRate	Defaults to 5			
rotate_with_direction:	Defaults to true. Makes unit body image locked to 0 degrees when false. Often used with animation_direction_*		bool	
animation_direction_units:	45 for 8 directions, 90 for 4 direction animation. Used with rotate_with_direction:false		float	
animation_direction_strideX:	Animation frames to offset on direction change.		int	
animation_direction_strideY:	Animation frames to offset on direction change. Used with frame_height.		int	
animation_direction_starting:	Direction for first frame		float	
	<b>Deprecated Keys (can be used but there are better ways)</b>			
animation_TYPE_start:	Deprecated, use [animation] section instead	animation_moving_start: 0	int	deprecated 1.13
animation_TYPE_end:	End frame, must be larger then start	animation_moving_end: 3	int	deprecated 1.13
animation_TYPE_scale_start:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float	deprecated 1.13
animation_TYPE_scale_end:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float	deprecated 1.13
animation_TYPE_speed:	Delay for each frame of animation. Larger values cause slower animation		float	deprecated 1.13
animation_TYPE_pingPong:	Play animation in reverse before repeating. Useful with scale_start/scale_end		bool	deprecated 1.13
<b>Section</b>	<b>[attack]</b>			
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
maxAttackRange:	(multiplied by globalScale)		float	

canAttack:	If set to false, can not attack any unit. Regards of other canAttack options below.		bool	
canAttackFlyingUnits:	can also be narrowed per turret		LogicBoolean	
canAttackLandUnits:	can also be narrowed per turret		LogicBoolean	
canAttackUnderwaterUnits:	can also be narrowed per turret		LogicBoolean	
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos. (can also be set per turret)		LogicBoolean	
turretMultiTargeting	Allow each turrets to fire at a different target at the same time. Very useful if [turret]limitingAngle is used			1.13
isMelee:	Used with a low attack range (like maxAttackRange: 9) makes src and target radius get added to range, and effects AI.		bool	
turretRotateWithBody	Are all turrets rotated when body rotates. Defaults to true		bool	
attackMovement:	normal/bomber. bomber attack movement will retreat when energy runs out		string	
dieOnAttack:			bool	
isFixedFiring:	Must aim body at target to shoot. Will often make the unit need to stop before it can aim and shoot.		bool	
aimOffsetSpread:	Offset each shot multiplied by target radius. Defaults to 0.6	aimOffsetSpread:0 will make unit always attack center	bool	
stopTargetingAfterFiring	Unit stops targeting after firing a shot. Rarely used or needed.			1.13
disablePassiveTargeting:	Unit only attacks manually ordered target. Rarely used or needed.		bool	
	<b>Deprecated Keys - can be used but better to set these per turret</b>			
turretSize:	(multiplied-by-globalScale)		float	
turretTurnSpeed:			float	
shootDelay:	Global delay-can also use delay-on-each-turret		float	
<b>Section</b>		<b>[turret_NAME]</b>		
Code	Description	Example	Value Type	
x:			float	
y:			float	
copyFrom:	Copy all values from another turret as default values for this turret	copyFrom: 1	turret ref	
projectile	Projectile fired from this turret.	eg: projectile: torpedo	projectile ref	
barrelX	Defaults to 0			1.13
barrelY	Defaults to size. Note: size and barrelY have the same meaning			1.13
size:	Controls the distance between the center of the turret and the point from where projectiles spawn.	size: 5	float	
turnSpeed	Max turn speed of the turret			
turnSpeedAcceleration	Defaults to disabled, and full turn speed is used.			
turnSpeedDeceleration	Defaults to turnSpeedAcceleration. Setting this higher than turn acceleration might allow faster targets to be hit			
idleDir	Defaults to 0		float	
idleDirReversing	Defaults to idleDir+180 unless attached to another turret (as attached turret will often be rotating when reversing)		float	1.13
shouldResetTurret:	Defaults true. False to disable the resetting turret angle when idle		bool	
attachedTo:	Id of another turret to attach to, will be positioned relative to it, and rotate with it.		turret ref	
slave:	Locks this turret's direction and shot cooldown to attached turret. Often used with warmup for multiple barrel guns		bool	
isMainNanoTurret:	Defaults to false. Turret to use for creating buildings, etc. should only be true on one turret, and should have canShoot set to false		bool	
energyUsage:	Required energy to fire weapon. Same as resourceUsage: energy=X		float	
resourceUsage	can be in credits/energy/hp/shield/ammo. Stops firing if not met	resourceUsage: credits=5, energy=5, hp=100, shield=5, ammo=1	price	
	<b>Timing</b>			
delay:	Override global shootDelay for this turret		float	
linkDelayWithTurret	When this other turret fires the cooldown delay on this turret will be reset/removed		turret ref	
warmup:	Delay before firing a shot.		float	
warmupCallDownRate	Rate to reduce warmup when turret is not ready to fire at any targets		float	
warmupNoReset	Defaults to false. When true warmup is not reset after firing a shot. Used with warmupCallDownRate		bool	
warmupShootDelayTransfer	Defaults to 0, a multiplier which reduces the next shot delay by the warmup value. When used with warmupNoReset, can make a each shot faster.		float	
	<b>Targeting control</b>			
canShoot:	Defaults to true		bool	
canAttackFlyingUnits:	Narrows targeting for this turret, note targeting for the whole unit in [attack] is applied first. (so you can only use this to target less not more)		LogicBoolean	
canAttackLandUnits:			LogicBoolean	
canAttackUnderwaterUnits:			LogicBoolean	
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos.		LogicBoolean	
canAttackCondition:	Normally, used to optionally disable a turret based on a LogicBoolean. Eg: this unit's height	canAttackCondition: if not self.flying	LogicBoolean	
clearTurretTargetAfterFiring				1.13
limitingRange:	Make this turret have less range than the maxAttackRange. Do not apply this to all turrets change maxAttackRange instead.		float	
limitingAngle	Linked with idleDir. Turret will only be able to fire at units +/- this angle.			1.13
limitingMinRange	Sets minimum range for turret.	limitingMinRange: 200		1.13
interceptProjectiles_withTags	Currently used with anti-nuke units.			1.13
interceptProjectiles_andTargetingGroundUnderDistance				
interceptProjectiles_andUnderDistance	Defaults to 2000, distance inflight before firing			
interceptProjectiles_andOverHeight	Defaults to 0			
laserDefenceEnergyUse:	Set to enable a projectile laser defence from this turret. Should also set the energyMax in core.		float	
	<b>Graphics and effects</b>			
invisible:	Don't render this turret, but still can shoot, etc.		LogicBoolean	
image:	Use custom image. Overrides unit's main turret image		file (image)	
image_applyTeamColors			bool	
image_drawOffsetX			float	
image_drawOffsetY			float	
chargeEffectImage:	Used with warmup. Shows a scaling effect image on turret barrel when charging.		file (image)	
warmupStartEffect			effects	
shoot_sound:			string	
shoot_sound_vol:			float	
shoot_flame:	Current types are: small, large, smoke, shockwave, or CUSTOM: effectSectionName	eg: shoot_flame: smoke, CUSTOM:lightFade, CUSTOM:pop*5	effects	
shoot_light			color	
idleSpin:	Spin rate when idle, used on missile turrets		float	
onShoot_playAnimation	Play a custom animation from an [animation] section after firing this turret		animation ref	
recoilOffset	Push turret forward or back after firing for a recoil effect. Value in pixels.		float	
recoilOutTime	Time to get to offset position after firing		float	
recoilReturnTime	Time to return to default position		float	
<b>Section</b>		<b>[projectile_NAME]</b>		
Code	Description	Example	Value Type	
life:	How long till this projectile gets removed if it hasn't hit a target, 300 might be a good starting point, change depending on speed and range		int	
deflectionPower:	Defaults to 1. Energy needed for laser defence to deflect. -1 to disable deflection (only disable for special weapons like flames)		float	
explodeOnEndOfLife	Default to false. True to explode at end of life with all side effects and area damage instead of disappearing		bool	1.13
autoTargetingOnDeadTarget:	Retarget to nearby unit if target dies while in transit		bool	
unloadUpToXUnitsFromSource	Unload X units from source unit, to projectile explode location		int	1.13
teleportSource	Move unit that shot this projectile to projectile explode location		bool	1.13
spawnUnit	Spawn new units of this type at projectile explode location	eg: spawnUnit: heavyTank, tank*5, hoverTank	unit types	1.13
tags			tags	1.13
flameWeapon:	Generates small flames on hit (only cosmetic)		bool	
	<b>Damage</b>			
directDamage:	Damage to target unit on hit. Does not work with targetGround:true as we don't have a unit target		int	
areaDamage:	directDamage or areaDamage is required		int	
areaRadius:	How wide areaDamage effects. Note this drops off.		float	1.13
areaExpandTime			float	1.13
areaHitAirAndLandAtSameTime			bool	1.13

areaHitUnderwaterAlways	Defaults to false			
buildingDamageMultiplier	Defaults to 1		float	1.13
shieldDamageMultiplier	Defaults to 1.	eg: 0 to do no damage to shields and 2 to do double damage	float	1.13
shieldDeflectionMultiplier	Defaults to 1.	eg: 0 to ignore shields and directly damage hull	float	1.13
armourIgnoreAmount	Amount of armour to ignore on target and do damage as if this armour was not there			
friendlyFire	Lets area effect projectiles damage own team units (can't damage allies). Useful for nuke-like weapons	friendlyFire: false / friendlyFire: true / friendlyFire: only-ignoreEnemy	bool/string	1.13
	<b>Movement</b>			
targetGround	Target ground, and don't home in on target. Note: only areaDamage is applied if targeting ground.		bool	
speed:			float	
targetSpeed:	Accelerate to this speed		float	
targetSpeedAcceleration			float	1.13
ballistic:			bool	
ballistic_delaymove_height:			float	
ballistic_height:			float	
targetGroundSpread:	Randomly makes the shot inaccurate by this amount. Also used by weapons like the flamethrower		float	
speedSpread:	Randomly change the starting projectile speed by this amount		float	
instant	Hit target instantly		bool	
instantReuseLast:	Recycles last projectile fired, only one projectile ever exists. Can turn lasers into beam weapons by using lower rate of fire and setting this to true		bool	
disableLeadTargeting	Disable the lead targeting calculations when aiming at a moving target. Defaults false.		bool	1.13
leadTargetingSpeedCalculation	The expected speed of this projectile for targetGround lead target calculation. Defaults to 'targetSpeed' if set otherwise 'speed'.		float	1.13
	<b>Graphics and effects</b>			1.13
color		color: #bebe50	color	
image:	Use custom image. Overrides drawType and frame		file (image)	
drawType	Built-in image to use. 0:projectiles.png 1:projectiles_large.png 2:projectiles2.png	drawType:1	int	
drawSize:	Scale image. Defaults to 1		float	
frame	Built-in image frame to use, starts at zero.		int	
hitSound:	Default true		bool	
explodeEffect		explodeEffect: smallExplosion, CUSTOM.myExplodeEffect	effect ref list	1.13
explodeEffectOnShield	Use this effect if shield is active on target		effect ref list	1.13
teamColorRatio	Mix 0-1 of team colour into color field		float	1.13
drawUnderUnits			bool	1.13
effectOnCreate			effect ref list	1.13
shouldRevealFog	Reveal fog to player on explode		bool	1.13
alwaysVisibleInFog			bool	1.13
nukeWeapon	Shows on mini-map when fired. Some other side effects as well.		bool	1.13
trailEffect	true for built-in defaults, but can also point to any custom effects		bool/effect	
trailEffectRate	Defaults to 3		float	1.13
lightSize:			float	
lightColor		lightColor: #ffe92b	color	
largeHitEffect:	Creates a large explosion and accompanying sound on hit (only cosmetic)		bool	
lightingEffect:	Draw as lighting works best with instant:true		bool	
laserEffect:	Draw as laser works best with instant:true		bool	
<b>Section [movement]</b>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
<b>movementType:</b>	Defines what kind of terrain the unit will be able to move over, along with other properties related to unit	movementType: LAND	string	
slowDeathFall:	Used with large aircraft. Makes the unit fall slowly while maintaining its speed at the time of death.	slowDeathFall: true	bool	
moveSpeed:	Maximum movement speed of the unit.	moveSpeed: 1.2	float	
moveAccelerationSpeed:	Defines how fast units accelerate to max speed.	moveAccelerationSpeed: 0.07	float	
moveDecelerationSpeed:	Don't make this too low or units will have trouble stopping at waypoints	moveDecelerationSpeed: 0.17	float	
reverseSpeedPercentage:	0.6 default. Over 0.4 will reverse for short distances (at 40% speed). If set to 1 will drive in reverse same	reverseSpeedPercentage: 0	float	
landOnGround:	Should flying unit land when idle.	landOnGround: false	bool	
targetHeight:	Defaults to 0 but if AIR movementType default is 35	targetHeight: 25	float	
targetHeightDrift:	Smooth animated height change. Defaults to 0 but if AIR movementType default is 1.5	targetHeightDrift: 1	float	
startingHeightOffset				
<b>maxTurnSpeed:</b>			float	
<b>turnAcceleration:</b>			float	
moveSlidingMode:			bool	
moveIgnoringBody:			bool	
moveSlidingDir:			int	
joinsGroupFormations:	Defaults to true. Changing not recommended		bool	
<b>Section [ai]</b>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
useAsBuilder:	Set to true if unit can build or repair buildings. Defaults to [core]jsBuilder.		bool	
disableUse:	Disallow AI building this unit or building		bool	
ai_upgradePriority	Defaults to 0.06. Set between 0-1, higher means AI is more likely to upgrade this unit before others		float	
	<b>Buildings only</b>			
buildPriority:	0-1. AI uses 0.8 for first land factory, 0.48 for air factory, 0.47 for first turret.		float	
noneInBaseExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the AIs base		float	
noneGlobalExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the any where on the map		float	
upgradedFrom:	Create link to another unit to preserve max counts for upgraded and non-upgraded types in same base.		string	
maxGlobal:			int	
maxEachBase:			int	
<b>Section [leg_#] / [arm_#]</b>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
<b>x:</b>	Sets position of the foot on the X axis.		float	
<b>y:</b>	Sets position of the foot on the Y axis.		float	
copyFrom:	Copy from another leg. Useful to only need to set leg values once	copyFrom: 1	int	
attach_x:	Sets the leg's attach point on the X axis.		float	
attach_y:	Sets the leg's attach point on the Y axis.		float	
rotateSpeed:			float	
endDirOffset	Target foot/end rotation relative to body			
lockMovement	Lock to unit body. Useful if walking unit converted to a flying unit.		bool	
heightSpeed:			float	
moveSpeed				
moveWarmUp				
holdDisMin:	Defaults to 7. Reposition leg at this distance if neighbor legs are not already repositioning.		float	
holdDisMax:	Defaults to 16. Force reposition of leg at this distance.		float	
holdDisMin_maxMovingLegs				
hold_moveOnlyIfFurthest				
holdDisMin_checkNeighbours				
hardLimit:	Defaults to 50. Force leg to never go this far. Better to not be reached.		float	
estimatingPositionMultiplier	defaults to 1. Predicts where unit will be for leg placement based on unit speed.		float	
	<b>Graphics and effects</b>			
hidden:			logic boolean	



image_end			image	1.13
image_end_shadow			image	1.13
image_end_teamColors				1.13
image_foot	same as image_end		image	1.13
image_foot_shadow			image	
image_middle			image	1.13
image_leg	same as image_middle		image	
draw_foot_on_top			bool	1.13
drawOverBody	Draw over the body of unit, defaults of false		bool	
drawUnderAllUnits	Draw over all units		bool	
drawDirOffset			float	
dust_effect:	Spawns dust particles on each step.		bool	
spinRate	Makes arm/leg spin, like idleSpin for turrets		float	
favourOppositeSideNeighbours	calculate neighbours with X 10 times closer than Y			
drawLegWhenZoomedOut	For performance, defaults changes based on unit size			
drawFootWhenZoomedOut	For performance, defaults changes based on unit size			
resetAngle:	Unused		float	
<b>Section</b> <i>[effect_NAME]</i>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
life	Defaults 200. Time till effect is removed. Set low as possible to reduce effect overhead.		float	1.13
alsoEmitEffects	Create more effects when created, useful for meta-effects. Note: other 'alsoEmitEffects' on created effects are ignored.		effect ref	1.13
ifSpawnFailsEmitEffects	If 'spawnChance' for this effects fails then emit these effects instead			
alsoPlaySound			sound ref	1.13
createWhenOffscreen	Defaults false.		bool	1.13
createWhenZoomedOut	Defaults true		bool	1.13
createWhenOverLiquid	Defaults true		bool	1.13
createWhenOverLand	Defaults true		bool	1.13
spawnChance	Default 1. If less than 1 effect only has a random chance of being created		float	1.13
showInFog	Default false		boolean	1.13
delayedStartTimer	Hide for x time before showing and updating effect.		float	1.13
liveAfterAttachedDies	Defaults false when attachedToUnit is being used		bool	1.13
priority	Defaults to high. verylow/low/high/veryhigh/critical. Takes effect when too many effects are being shown at once.		string	1.13
	<b>Movement</b>			
attachedToUnit	Attach to unit or projectile that created this effect. Will move with this object.		bool	1.13
alwaysStartDirAtZero	Ignore source/attached unit dir		bool	1.13
atmospheric	Apply drag to slow this effect down and add small wind effects		bool	1.13
physics	Fall to ground and bounces. Needs height to take effect.		bool	1.13
physicsGravity	Defaults to 1. height speed acceleration when physics: true		float	1.13
xOffsetRelative	Offset starting effect position. Relative to direction of attached turret, projectile, unit		float	1.13
yOffsetRelative	Offset starting effect position. Relative to direction of attached turret, projectile, unit		float	1.13
xOffsetRelativeRandom	Random offset by +/- this value		float	1.13
yOffsetRelativeRandom	Random offset by +/- this value		float	1.13
xOffsetAbsolute	Offset starting effect by position ignoring direction of attached turret, projectile, unit		float	1.13
yOffsetAbsolute	Offset starting effect by position ignoring direction of attached turret, projectile, unit		float	1.13
xOffsetAbsoluteRandom	Random offset by +/- this value		float	1.13
yOffsetAbsoluteRandom	Random offset by +/- this value		float	1.13
xSpeedRelative			float	1.13
ySpeedRelative			float	1.13
xSpeedRelativeRandom	Randomly change by -value to value		float	1.13
ySpeedRelativeRandom	Randomly change by -value to value		float	1.13
xSpeedAbsolute			float	1.13
ySpeedAbsolute			float	1.13
xSpeedAbsoluteRandom	Randomly change by -value to value		float	1.13
ySpeedAbsoluteRandom	Randomly change by -value to value		float	1.13
hOffset	height offset from source		float	1.13
hOffsetRandom	Randomly change by -value to value		float	1.13
hSpeed	height speed		float	1.13
hSpeedRandom	Randomly change by -value to value		float	1.13
dirOffset	rotation		float	1.13
dirOffsetRandom	Randomly change by -value to value		float	1.13
dirSpeed	rotation speed		float	1.13
dirSpeedRandom	Randomly change by -value to value		float	1.13
	<b>Graphics</b>			
frameIndex	frame of to use		int	1.13
stripIndex	A built-in image set to use. Cannot be used with custom image	effects/explode_big/light_50/flare/effects/effects2/projectiles/projectiles2/explode_b	int/string	1.13
Image	Custom image file to use. Cannot be used with stripIndex.		image	1.13
imageShadow	Custom image file to use for shadows		image	
scaleTo	Defaults to 1		float	1.13
scaleFrom	Defaults to 1		float	1.13
color	Defaults #FFFFFFF		color	1.13
teamColorRatio	0-1			
drawUnderUnits			bool	1.13
fadeInTime	Fade alpha from 0% to 100% for this time at start		float	1.13
fadeOut	Fade alpha from 100% to 0% based on life. Set alpha is higher than 1 to delay fade		bool	1.13
alpha	Capped between 0-1. Can be set higher than 1 to delay fadeOut effects			
shadow	True to draw a shadow. Forced true if imageShadow is used			
	<b>Animation</b>			
total_frames	Total frames of 'image', used with animation or frameIndex. Only needed with custom images		int	1.13
animateFrameStart			int	1.13
animateFrameEnd			int	1.13
animateFramePingPong			int	1.13
animateFrameSpeed			time	1.13
animateFrameSpeedRandom			time	1.13
<b>Section</b> <i>[animation_NAME]</i>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
onActions : Unknown	move, attack, idle, underConstruction, underConstructionWithLinkedBuiltTime, queuedUnits			1.13
onActionsQueuedUnitPlayAt : float	For onAction: queuedUnits. Amount queue needs to reach before starting, set between 0-1			1.13
blendIn : time	Blend with last animation for this time			1.13
blendOut : time	Blend with next animation for this time			1.13
pingPong	Play animation in reverse after it ends		bool	1.13
KeyframeTimeScale : float	Scales all keyframe times, useful to make an animation faster/slower without changing everything		float	1.13
	<b>Keyframes - create as many as needed</b>			
arm#_[time]	Adds a keyframe at time. Use multiple times to create animation.	eg: arm1_5s: {x: 5, dir: 90 }		1.13
leg#_[time]	Adds a keyframe at time. Use multiple times to create animation.			
body_[time]	Adds a keyframe at time for body. Only frame and scale allowed on body	eg: body_4s: {frame: 4, scale: 0.5}		

	Deprecated Keys (can be used but there are better ways)			
start--int	Start-image-frame--deprecated			1.13
end--int	End-image-frame--deprecated			1.13
scale_start--float	Start-scale--Deprecated--use-body-keyframes-instead-			1.13
scale_end--float	End-scale--Deprecated--use-body-keyframes-instead-			1.13
speed--float	Speed-smaller-is-faster-Only-effects-start-end-scale-start-scale-end			1.13
Section	[action_NAME]			
Code	Description	Example	Value Type	
text	Text shown in UI		string	
text_(LANG)				
description	A display text when you select your unit's action, used to explain it's purpose.		string	
description_(LANG)				
displayType	none, rally, upgrade, queueUnit, building, action, infoOnly, infoOnlyNoBox			
displayRemainingStockpile	Queue is shown as number of times action can be triggered based on price		bool	
pos	Order action appears in UI		float	
iconImage				
	Requirements to trigger or enable			
price	The price of your action for the unit. Disables action if not available. Defaults to credits if unlabelled	price: credits=5, energy=5, hp=100, shield=5, ammo=1	resources	
isActive	Defaults true. If false then action is disabled and shown in red in UI.		LogicBoolean	
isVisible	Defaults true. If false action is hidden from UI and disabled.		LogicBoolean	
isLocked	Defaults false. If true action is disabled, and a lock icon is shown. Mostly used for no nuke game modes		LogicBoolean	
isLockedMessage			string	
onlyOneUnitAtATime	When action is picked in UI, only one unit selected with get this action. Defaults to false.		bool	
autoTrigger	When true triggers the effects of this action instantly (ignoring price, isActive, isVisible, buildSpeed, etc)	autoTrigger: if self.overWater()	LogicBoolean	
ai_isDisabled	Defaults false. Stop AI using this action.		LogicBoolean	
	While action is queued			
buildSpeed		buildSpeed: 5s	time	
highPriorityQueue	Defaults to false. If true this action skips all other low priority actions in queue. Useful for fireTurret actions.		bool	
canPlayerCancel			bool	
whenBuilding_cannotMove	Stops unit moving while action is being applied. Useful for deploy like actions.		bool	
whenBuilding_playAnimation	Play a custom animation from an [animation] section when in active queue		animation ref	
whenBuilding_rotateTo	Rotate unit body to this direction when action is in active queue		float	
whenBuilding_rotateTo_orBackwards	If true allow rotation in 180 degrees from whenBuilding_rotateTo when this is a smaller angle		bool	
whenBuilding_rotateTo_waitTillRotated	Pause action queue till rotation is finished		bool	
whenBuilding_temporarilyConvertTo	Convert to another unit while action is in active queue. Note: actions from the original unit will be kept		unit ref	
spawnEffectsOnQueue	Effects to spawn at unit when action is first added to queue		effect ref	
playSoundToPlayerOnQueue	Global sound to play to unit's player only when action is first added to queue		sound ref	
	Results (What happens) - Must be at least one result listed			
convertTo	Convert your unit into another unit. properties are preserved.		unit ref	
fireTurretXAIGround	When action finishes fire target turret at point on ground, bypasses canShoot rules in turret.	fireTurretXAIGround: nukeSilo	turret ref	
fireTurretXAIGround_withOffset	If not set player targets the ground with GUI, if a point is set this step is skipped	fireTurretXAIGround_withOffset: 0,0	point	
fireTurretXAIGround_withProjectile	Used with fireTurretXAIGround. Defaults to target turret's normal projectile.		projectile ref	
addEnergy	Adds energy to unit. Has no effect unless energyMax is set		float	
addResources	Add these resources when action finishes.	addResources: credits=5, energy=-5, hp=-100, shield=5, ammo=1	resources	
alsoTriggerAction	Trigger to results of another action as well. Ignores action's requirements		action ref	
alsoQueueAction	Adds another action into the normal unit's queue. Ignores action's requirements		action ref	
alsoTriggerOrQueueActionConditional	Defaults true. alsoTriggerAction and alsoQueueAction are ignored if this works out to be false.		LogicBoolean	
spawnEffects	Effects to spawn at unit		effect ref	
playSoundAtUnit	Local sound to play when action finishes		sound ref	
playSoundGlobally	Global sound to play to all players in game		sound ref	
playSoundToPlayer	Global sound to play to unit's player only		sound ref	
addUnitsIntoTransport	Add units into transport, use self.transportingCount() to check for space before adding	addUnitsIntoTransport: tank*3, heavyTank		
deleteNumUnitsFromTransport			int	
resetCustomTimer	Reset timer used with self.customTimer()		LogicBoolean	
Type	LogicBoolean			
Code	Description	Example	Targets	
true				
false				
if	Start all logic booleans with if, unless just using true/false			
and		if self.isInWater() and self.energy(greaterThan=1)		
or		if (self.energy(greaterThan=1) or self.ammo(greaterThan=1)) and self.isFlying()		
not		if not self.isOverLiquid()		
	Unit location and movement			
self.isUnderwater()				
self.isAtGroundHeight()				
self.isFlying()				
self.isMoving()				
self.isAtTopSpeed()				
self.isInWater()	Touching water			
self.isOverwater()	Touching or over a water tile			
self.isOverLiquid()				
self.isOverCliff()				
self.isOverPassableTile()	(parameters: type)			
self.isOverOpenLand()	shortcut for self.isOverPassableTile(type="LAND")			
	Unit stats			
self.hp()	(parameters: greaterThan, lessThan, empty, full)			
self.ammo()	(parameters: greaterThan, lessThan, empty, full)			
self.isAmmoEmpty()	shortcut for self.ammo(empty=true)			
self.ammoIncludingQueued()	Also includes ammo from actions still in queue (parameters: greaterThan, lessThan, empty, full)	if self.ammoIncludingQueued(lessThan=12)		
self.energy()	(parameters: greaterThan, lessThan, empty, full)			
self.energyIncludingQueued()	Also includes energy from actions still in queue (parameters: greaterThan, lessThan, empty, full)			
self.isEnergyFull()	shortcut for self.energy(full=true)			
self.isEnergyEmpty()	shortcut for self.energy(empty=true)			
self.shield()	(parameters: greaterThan, lessThan, empty, full)			
self.kills()	(parameters: greaterThan, lessThan)			
	Misc			
game.nukesEnabled()				
self.hasFlag()	Boolean flag saved into units for mods to use. (parameters: id=0-31)	Use addResources in action change this vaule		
self.tags()	(parameters: includes)			
self.transportingCount()	(parameters: greaterThan, lessThan, empty)			
self.isAttacking()				
self.transportingUnitWithTags()	(parameters: includes)	self.transportingUnitWithTags(includes="human")		
self.hasTakenDamage()	(parameters: withinSeconds=X, laterThanSeconds=X)	self.hasTakenDamage(withinSeconds=1)		
self.timeAlive()	(parameters: withinSeconds=X, laterThanSeconds=X)			

self.lastConverted()	(parameters: withinSeconds=X, laterThanSeconds=X)		
self.customTimer()	(parameters: withinSeconds=X, laterThanSeconds=X)		
self.isOnNeutralTeam()			
numberOfUnitsInTeam()	(parameters: withTag, greaterThan, lessThan, withinRange, incompleteBuildings, factoryQueue)	numberOfUnitsInTeam(withTag='techUnlockBuilding', greaterThan=0)	
Type	Resources - used by addResources, price, etc		
Code	Description	Example	Targets
credits	Global resource		
energy			
hp			
shield		price: hp=-100, shield=100	
ammo	Hidden value on each unit for use by mods		
setFlag	use with addResources, resourceUsage or price. 0-31. Flags are stored in each unit	setFlag=1	
unsetFlag	use with addResources, resourceUsage or price. 0-31		
hasFlag	use with price or resourceUsage		
hasMissingFlag	use with price or resourceUsage		

This sheet is for 1.13, for 1.13.2 see sheet tabs below.

Section	[core]		
Code	Description	Example	Value Type
	<b>Common keys</b>		
name:	Defines the raw name for your unit, in which case the game uses to identify as a unique name. (This name is used for the unit's name in the game)	name: customTank_1	string
altNames:	comma separated list of names. Like name but lower priority, useful for multiple optional mods.	altNames: custTank1, customTank1, cTank1	string(s)
class:	Reserved for future use, must be CustomUnitMetadata	class: CustomUnitMetadata	string
price:	The cost of your unit from builders/buildings.	price: 500	int
mass:	The 'weight' of your unit, this will define how it collides with other units, a greater value means it is tougher	mass: 3000	int
techLevel:	Defines the Tech Level of your unit, there are 3 levels and each will appear in a different color in the GUI	techLevel: 1	int
buildSpeed:	Time it takes to build unit (may multiply with builder speed) Formulae for seconds: 10 * 60 = 0.16 (0.16 seconds)	buildSpeed: 0.16	float
radius:	Circular area around your unit that makes it selectable (mouse click/screen touch)	radius: 20	int
isBio:	Choose whether your unit is biological or not, will effect sound and splat (unless hideScorchMark is true)	isBio: false	bool
isBug:	Changes some death defaults, and sort order in sandbox		bool
isBuilder	Normally required if this unit places buildings. Defaults to [ai]useAsBuilder		
	<b>Unit stats</b>		
maxHp:	The max health for your unit. (will spawn with this value)	maxHp: 200	int
selfRegenRate:	Passive self repair rate		float
maxShield:	The max shield hitpoints of your unit. Can start with 0 hitpoints if startShieldAtZero is set to true	maxShield: 500	int
startShieldAtZero:	Sets the unit's shield to 0 HP when it is first built.	startShieldAtZero: true	bool
shieldRegen:	Passive shield regen rate	shieldRegen: 0.15	float
energyMax:	Defaults to 0. Energy can be used as ammo for turrets or for laser defense	energyMax: 1	float
energyRegen:			float
energyStartingPercentage:	Sets the percentage of charged energy when the unit is first built.	energyStartingPercentage: 0.5	float
energyNeedsToRechargeToFull:	Disable weapons using energy after reaching zero till fully recharged		float
armour	Damage taken away from each hit (not currently used in any core units)		
armourMinDamageToKeep	Min damage to keep. Defaults to 1		
generation_credits:	Income unit creates	generation_credits: 2	int
generation_delay:	How often generation_credits is added. Defaults to 40 (changing not recommended)	generation_delay: 40	int
	<b>UI and Graphics related keys</b>		
showInEditor:	Set to false to hide unit in sandbox editor (Default true)	showInEditor: true	bool
displayText:	The name of your unit that the game shows to the player.	displayText: Custom Tank	string
displayText_(LANG)	LANG=ISO 639-1 Code to show this text instead when game is in this language	displayText_es: Hola	
displayDescription	Description of your unit that the game shows to the player.	displayDescription: - Fast movement\n- Light damage.	string
displayDescription_(LANG)	LANG=ISO 639-1 Code to show this text instead when game is in this language	displayDescription_es: -Movimiento rapido\n-Daño ligero	
displayLocaleKey:	Translation file key for unit name and description	displayLocaleKey: units.mechArtillery	string
displayRadius:	Defaults to radius value. Set to show a larger or smaller selection box ui on units.	displayRadius: 20	int
shieldRenderRadius			
shieldDisplayOnlyDeflection:	Hide shield (if active) unless deflecting shot	shieldDisplayOnlyDeflection: false	bool
shieldDeflectionDisplayRate:	Defaults to 4. High value causes shield deflection to fade disappear faster	shieldDeflectionDisplayRate: 3	float
	<b>Building only keys</b>		
isBuilding:			bool
footprint	Only applies to buildings, tiles taken up which block unit movement. Defaults to 0,0,0,0 = 1 center tile	footprint: -1,-1,1,1	
constructionFootprint	Only applies to buildings, tiles taken up for placement of other buildings. Defaults to 0,0,0,0	constructionFootprint: -1,-1,1,3	
buildingSelectionOffset:	Defaults to 0. Adds or removes padding on the drawn selection rect in UI		int
buildingToFootprintOffsetX	Defaults to 10		float
buildingToFootprintOffsetY	Defaults to 10		float
placeOnlyOnResPool:	Normally used for extractors		bool
selfBuildRate:	Rate unit builds it's self when placed without a builder		float
	<b>Misc Keys</b>		
copyFrom:	Use unit data from another ini file as base	copyFrom: customTank_Default	file (ini)
dont_load:	Do not load unit, and don't error on missing data. Can be useful when used with copyFrom	dont_load: true	bool
overrideAndReplace:	Override another unit with this unit. Build links pointing to target unit will be replaced with this unit. And	overrideAndReplace: builder, combatEngineer	string(s)
globalScale:	Defaults to 1. Changing not recommended	globalScale: 1	float
isLocked:	Disallow building of this unit. Can be used with overrideAndReplace to restrict units the player is allowed to build	isLocked: false	bool
isLockedIfGameModeNoNuke	Disallows building of this unit if nukes are disabled during match setup.	isLockedIfGameModeNoNuke: false	
experimental:	Tag unit as experimental. Affects zoomed out icon and end game stats.	experimental: false	bool
stayNeutral:	Set to false to disable capture when unit is on the neutral team	stayNeutral: false	bool
createNeutral			
resourceRate:	Used with canReclaimResources. Experimental feature that is not yet finished.		float
tags	List of any comma separated strings. Only used for transportUnitsRequireTag right now, will be used for other things in future		
fogOfWarSightRange:	Set number of tiles this unit can see. Defaults to 15	fogOfWarSightRange: 15	int
softCollisionOnAll:			int
isUnrepairableUnit	No unit can repair this unit		
isPickableStartingUnit	If true, unit is added to dropdowns for starting unit in game setup menus.		
startFallingWhenStartingUnit			
soundOnAttackOrder	List of sounds. Only one will be played each attack order		sound
soundOnMoveOrder	List of sounds. Only one will be played each move order		sound
	<b>Transport related keys</b>		
transportSlotsNeeded:	Defaults to 1. Number of slots this unit uses up in a transport, Experimentals are often set to 5 to stop small transports holding them.		int
maxTransportingUnits	Number of slots this units as for transporting other units.		int
transportUnitsRequireTag	Only allow trasport of units that have one of these tags.		tags
transportUnitsRequireMovementType	Only allow trasport of units that have one of these movement types.		movementTypes
transportUnitsBlockAirAndWaterUnits	Defaults to true if transportUnitsRequireMovementType is empty		
transportUnitsKeepBuiltUnits	Makes built units stay inside transport instead of exiting it once ready		LogicBoolean
transportUnitsCanUnloadUnits	Defaults to: if not self.isOverLiquid() and not self.isMoving()		LogicBoolean
transportUnitsAddUnloadOption	Should unload button be added to the menu		LogicBoolean
transportUnitsKillOnDeath	Default true. If false transporting units don't die when transport dies	eg: if self.isOverLiquid()	LogicBoolean
transportUnitsHealBy	Rate to heal units being transported		float
	<b>Construction and factory related keys</b>		
canRepairBuildings:	isBuilder:true is required as well right now)		bool
canRepairUnits:	isBuilder:true is required as well right now)		bool
canReclaimResources:	Used with resourceRate. Experimental feature that is not yet finished.		bool
autoRepair:	Automatically try and repair damaged units in range ([ai]useAsBuilder:true is required as well right now)		bool
nanoRange:	Defaults to 85		int
nanoRepairSpeed:	Defaults to 0.2		float
nanoBuildSpeed:	Defaults to 1		float
nanoFactorySpeed:	Defaults to 1		float
builtFrom_#_name:	canBuild and builtFrom have the same effect just in the opposite direction. BuiltFrom is useful if adding new units to existing buildings		string
builtFrom_#_pos:	Order action appears in UI.		string
builtFrom_#_forceNano:	Build as if this was a building (even if it's a unit)		bool
canBuild_#_name:	List of units this unit can create. Can be buildings or units. (isBuilder:true and canRepairBuildings:true is recommended when adding buildings)		units

canBuild_#_pos:	Order action appears in UI		string	
canBuild_#_tech:	Tech level. Mostly just effects action colour in UI. Defaults to 1		int	
canBuild_#_forceNano:	Build target as if it was a building (even if it's a unit)		bool	
exit_x:	Where created or unloaded units appear. Defaults to 0	exit_x: 0	float	
exit_y:	Where created or unloaded units appear. Defaults to 5.	exit_x: 5	float	
exit_dirOffset : float	Defaults to 180 for units. 0 for buildings			1.13
exit_heightOffset : float	Defaults to 0			1.13
exit_moveAwayAmount : float	Defaults to 70			1.13
	<b>Death related keys</b>			
dieOnConstruct:	Delete this unit when it starts making a building. Target building likely will need selfBuildRate set to be created without a builder.		bool	
dieOnZeroEnergy	Kill this unit if energy level reaches zero.	dieOnZeroEnergy: false	bool	1.13
numBitsOnDeath			int	
nukeOnDeath:			bool	
nukeOnDeathRange:			float	
nukeOnDeathDamage:			float	
nukeOnDeathDisableWhenNoNuke:	Defaults to false		bool	
fireTurretXAtSelfOnDeath				
explodeOnDeath	defaults to true		boolean	1.13
explodeOnDeathGroundCollision	defaults to true		boolean	1.13
effectOnDeath			effect ref	1.13
effectOnDeathGroundCollision			effect ref	1.13
unitsSpawnedOnDeath	unit list	unitsSpawnedOnDeath: tank*5, hoverTank	units	1.13
unitsSpawnedOnDeath_setToTeamOfLastAttacker			bool	
hideScorchMark:			bool	
	<b>Deprecated Keys (can be used but there are better ways)</b>			
action_#_convertTo:	Deprecated in 1.13, use [action_x] sections instead	action_1_convertTo: customTank_2	string	deprecated 1.13
action_#_pos:	Order action appears in UI		float	deprecated 1.13
action_#_price:	The price of your action for the unit. (All your sub actions will be linked to the # you use)	action_1_price: 1000	int	deprecated 1.13
action_#_text:	A display text when you select your unit's action, used to explain it's purpose. (All your sub actions will be linked to the # you use)	action_1_text: Upgrade to Custom Tank 2	string	deprecated 1.13
action_#_description:			string	deprecated 1.13
action_#_addEnergy:	Adds energy to unit. Has no effect unless energyMax is set		float	deprecated 1.13
action_#_whenBuilding_cannotMove:	Stops unit moving while action is being applied. Useful for deploy like actions.		bool	deprecated 1.13
<b>Section</b>	<b>[graphics]</b>			
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
image:	File path to png image.		file (image)	
image_back:	An optional image drawn behind other units. Useful for factories that units exit		file (image)	
image_wreak:	Image to use when unit dies. Can be NONE to leave no wreck		file (image)	
image_offsetX:			int	
image_offsetY:			int	
teamColorsUseHue:	False: Green pixels on unit gets converted to team color. True: Whole unit is tinted the team colour. Defaults to false		bool	
imageSmoothing:			bool	
scaleImagesTo:	Resize image to fit this value in pixels. Effects leg, and shadow images as well.		float	
imageScale:	Resize image. Defaults to 1. Effects leg, and shadow images as well.		float	
drawLayer:	Set to experimentals, or leave line out		string	
whenBeingBuiltMakeTransparentTill			float	1.13
	<b>Turrets (images can also be set on each turret)</b>			
image_turret:	Default image for all turrets. can also be set per turret		file (image)	
teamColorsOnTurret	Defaults false. Apply team colours on turret as well. Also effects pre-turret images		bool	
scaleTurretImagesTo:			float	
lock_body_rotation_with_main_turret:	Locks body image locked to first turret's direction		bool	
lock_leg_rotation_with_main_turret			bool	1.13
	<b>Shadow</b>			
image_shadow:	Image file, NONE, or AUTO. (AUTO will use image and make it transparent black only.)		file (image)	
shadowOffsetX:			float	
shadowOffsetY:			float	
lock_shadow_rotation_with_main_turre	Locks body image shadow locked to first turret's direction		bool	
	<b>Effects and animation</b>			
total_frames	Defaults to 1. Animations require this.		int	
frame_width:	Calculated for you if total frames is set, but can be overridden		int	
frame_height:	Defaults to image height		int	
splstEffect:	True to create a water wave effect when over water. Default false		bool	
dustEffect:	True to create a dust effect when over land. Default false		bool	
splstEffectReverse:	True to also create effect when unit is reversing		bool	
dustEffectReverse:	True to also create effect when unit is reversing		bool	
movementEffect	Custom movement effect, can be anything	eg: movementEffect: smoke, CUSTOM:fastDust*2, CUSTOM:pop*5	effect	1.13
movementEffectReverse			effect	1.13
movementEffectRate			float	1.13
movementEffectReverseFlipEffects	Create effect as if unit has rotated 180 when reversing		bool	1.13
repairEffect	Custom movement effect, can be anything. Replaces default effect from builders			
repairEffectAtTarget				
repairEffectRate	Defaults to 5			
rotate_with_direction:	Defaults to true. Makes unit body image locked to 0 degrees when false. Often used with animation_direction_*		bool	
animation_direction_units:	45 for 8 directions, 90 for 4 direction animation. Used with rotate_with_direction:false		float	
animation_direction_strideX:	Animation frames to offset on direction change.		int	
animation_direction_strideY:	Animation frames to offset on direction change. Used with frame_height.		int	
animation_direction_starting:	Direction for first frame		float	
	<b>Deprecated Keys (can be used but there are better ways)</b>			
animation_#_TYPE_start:	Deprecated, use [animation] section instead	animation_moving_start: 0	int	deprecated 1.13
animation_#_TYPE_end:	End frame, must be larger then start	animation_moving_end: 3	int	deprecated 1.13
animation_#_TYPE_scale_start:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float	deprecated 1.13
animation_#_TYPE_scale_end:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float	deprecated 1.13
animation_#_TYPE_speed:	Delay for each frame of animation. Larger values cause slower animation		float	deprecated 1.13
animation_#_TYPE_pingPong:	Play animation in reverse before repeating. Useful with scale_start/scale_end		bool	deprecated 1.13
<b>Section</b>	<b>[attack]</b>			
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
maxAttackRange:	(multiplied by globalScale)		float	
canAttack:	If set to false, can not attack any unit. Regards of other canAttack options below.		bool	
canAttackFlyingUnits:	can also be narrowed per turret		LogicBoolean	
canAttackLandUnits:	can also be narrowed per turret		LogicBoolean	
canAttackUnderwaterUnits:	can also be narrowed per turret		LogicBoolean	
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos. (can also be set per turret)		LogicBoolean	
turretMultiTargeting	Allow each turrets to fire at a different target at the same time. Very useful if [turret]limitingAngle is used			1.13
isMelee:	Used with a low attack range (like maxAttackRange: 9) makes src and target radius get added to range, and effects AI.		bool	
turretRotateWithBody	Are all turrets rotated when body rotates. Defaults to true		bool	
attackMovement:	normal/bomber. bomber attack movement will retreat when energy runs out		string	

dieOnAttack:			bool	
isFixedFiring:	Must aim body at target to shoot. Will often make the unit need to stop before it can aim and shoot.		bool	
aimOffsetSpread:	Offset each shot multiplied by target radius. Defaults to 0.6	aimOffsetSpread:0 will make unit always attack center	bool	
stopTargetingAfterFiring	Unit stops targeting after firing a shot. Rarely used or needed.			1.13
disablePassiveTargeting:	Unit only attacks manually ordered target. Rarely used or needed.		bool	
	<b>Deprecated Keys - can be used but better to set these per turret</b>			
turretSize:	(multiplied-by-globalScale)		float	
turretTurnSpeed:			float	
shootDelay:	Global delay; can also use delay on each turret		float	
<b>Section</b> <i>[turret_NAME]</i>				
Code	Description	Example	Value Type	
x:			float	
y:			float	
copyFrom:	Copy all values from another turret as default values for this turret	copyFrom: 1	turret ref	
projectile	Projectile fired from this turret.	eg. projectile: torpedo	projectile ref	
barrelX	Defaults to 0			1.13
barrelY	Defaults to size. Note: size and barrelY have the same meaning			1.13
size:	Controls the distance between the center of the turret and the point from where projectiles spawn.	size: 5	float	
turnSpeed	Max turn speed of the turret			
turnSpeedAcceleration	Defaults to disabled, and full turn speed is used.			
turnSpeedDeceleration	Defaults to turnSpeedAcceleration. Setting this higher than turn acceleration might allow faster targets to be hit			
idleDir	Defaults to 0		float	
idleDirReversing	Defaults to idleDir+180 unless attached to another turret (as attached turret will often be rotating when reversing)		float	1.13
shouldResetTurret:	Defaults true. False to disable the resetting turret angle when idle		bool	
attachedTo:	Id of another turret to attach to, will be positioned relative to it, and rotate with it.		turret ref	
slave:	Locks this turret's direction and shot cooldown to attached turret. Often used with warmup for multiple barrel guns		bool	
isMainNanoTurret:	Defaults to false. Turret to use for creating buildings, etc. should only be true on one turret, and should have canShoot set to false		bool	
energyUsage:	Required energy to fire weapon. Same as resourceUsage: energy=X		float	
resourceUsage	can be in credits/energy/hp/shield/ammo. Stops firing if not met	resourceUsage: credits=5, energy=5, hp=100, shield=5, ammo=1	price	
	<b>Timing</b>			
delay:	Override global shootDelay for this turret		float	
linkDelayWithTurret	When this other turret fires the cooldown delay on this turret will be reset/removed		turret ref	
warmup:	Delay before firing a shot.		float	
warmupCallDownRate	Rate to reduce warmup when turret is not ready to fire at any targets		float	
warmupNoReset	Defaults to false. When true warmup is not reset after firing a shot. Used with warmupCallDownRate		bool	
warmupShootDelayTransfer	Defaults to 0, a multiplier which reduces the next shot delay by the warmup value. When used with warmupNoReset, can make a each shot faster.		float	
	<b>Targeting control</b>			
canShoot:	Defaults to true		bool	
canAttackFlyingUnits:	Narrows targeting for this turret, note targeting for the whole unit in [attack] is applied first. (so you can only use this to target less not more)		LogicBoolean	
canAttackLandUnits:			LogicBoolean	
canAttackUnderwaterUnits:			LogicBoolean	
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos.		LogicBoolean	
canAttackCondition:	Normally, used to optionally disable a turret based on a LogicBoolean. Eg: this unit's height	canAttackCondition: if not self flying	LogicBoolean	
clearTurretTargetAfterFiring				1.13
limitingRange:	Make this turret have less range than the maxAttackRange. Do not apply this to all turrets change maxAttackRange instead.		float	
limitingAngle	Linked with idleDir. Turret will only be able to fire at units +/- this angle.			1.13
limitingMinRange	Sets minimum range for turret.	limitingMinRange: 200		1.13
interceptProjectiles_withTags	Currently used with anti-nuke units.			1.13
<b>interceptProjectiles_andTargetingGroundUnderDistance</b>				
interceptProjectiles_andUnderDistance	Defaults to 2000, distance inflight before firing			
interceptProjectiles_andOverHeight	Defaults to 0			
laserDefenceEnergyUse:	Set to enable a projectile laser defence from this turret. Should also set the energyMax in core.		float	
	<b>Graphics and effects</b>			
invisible:	Don't render this turret, but still can shoot, etc.		bool	
image:	Use custom image. Overrides unit's main turret image		file (image)	
chargeEffectImage:	Used with warmup. Shows a scaling effect image on turret barrel when charging.		file (image)	
warmupStartEffect			effects	
shoot_sound:			string	
shoot_sound_vol:			float	
shoot_flame:	Current types are: small, large, smoke, shockwave, or CUSTOM: effectSectionName	eg: shoot_flame: smoke, CUSTOM:lightFade, CUSTOM:pop*5	effects	
shoot_light			color	
idleSpin:	Spin rate when idle, used on missile turrets		float	
onShoot_playAnimation	Play a custom animation from an [animation] section after firing this turret		animation ref	
recoilOffset	Push turret forward or back after firing for a recoil effect. Value in pixels.		float	
recoilOutTime	Time to get to offset position after firing		float	
recoilReturnTime	Time to return to default position		float	
<b>Section</b> <i>[projectile_NAME]</i>				
Code	Description	Example	Value Type	
life:	How long till this projectile gets removed if it hasn't hit a target, 300 might be a good starting point, change depending on speed and range		int	
deflectionPower:	Defaults to 1. Energy needed for laser defence to deflect. -1 to disable deflection (only disable for special weapons like flames)		float	
explodeOnEndOfLife	Default to false. True to explode at end of life with all side effects and area damage instead of disappearing		bool	1.13
autoTargetingOnDeadTarget:	Retarget to nearby unit if target dies while in transit		bool	
unloadUpToXUnitsFromSource	Unload X units from source unit, to projectile explode location		int	1.13
teleportSource	Move unit that shot this projectile to projectile explode location		bool	1.13
spawnUnit	Spawn new units of this type at projectile explode location	eg: spawnUnit: heavyTank, tank*5, hoverTank	unit types	1.13
tags			tags	1.13
flameWeapon:	Generates small flames on hit (only cosmetic)		bool	
	<b>Damage</b>			
directDamage:	Damage to target unit on hit. Does not work with targetGround:true as we don't have a unit target		int	
areaDamage:	directDamage or areaDamage is required		int	
areaRadius:	How wide areaDamage effects. Note this drops off.		float	1.13
areaExpandTime			float	1.13
areaHitAirAndLandAtSameTime			bool	1.13
areaHitUnderwaterAlways	Defaults to false			
buildingDamageMultiplier	Defaults to 1		float	1.13
shieldDamageMultiplier	Defaults to 1.	eg: 0 to do no damage to shields and 2 to do double damage	float	1.13
shieldDeflectionMultiplier	Defaults to 1.	eg: 0 to ignore shields and directly damage hull	float	1.13
armourIgnoreAmount	Amount of armour to ignore on target and do damage as if this armour was not there			
friendlyFire	Lets area effect projectiles damage own team units (can't damage allies). Useful for nuke-like weapons	friendlyFire: false / friendlyFire: true / friendlyFire: only-ignoreEnemy	bool/string	1.13
	<b>Movement</b>			
targetGround	Target ground, and don't home in on target. Note: only areaDamage is applied if targeting ground.		bool	
speed:			float	
targetSpeed:	Accelerate to this speed		float	
targetSpeedAcceleration			float	1.13
ballistic:			bool	

ballistic_delaymove_height:			float	
ballistic_height:			float	
targetGroundSpread:	Randomly makes the shot inaccurate by this amount. Also used by weapons like the flamethrower		float	
speedSpread:	Randomly change the starting projectile speed by this amount		float	
instant	Hit target instantly		bool	
instantReuseLast:	Recycles last projectile fired, only one projectile ever exists. Can turn lasers into beam weapons by using lower rate of fire and setting this to true		bool	
disableLeadTargeting			bool	1.13
leadTargetingSpeedCalculation			float	1.13
Graphics and effects				1.13
color		color: #bebe50	color	
image:	Use custom image. Overrides drawType and frame		file (image)	
drawType	Built-in image to use. 0:projectiles.png 1:projectiles_large.png 2:projectiles2.png	drawType:1	int	
drawSize:	Scale image. Defaults to 1		float	
frame	Built-in image frame to use, starts at zero.		int	
hitSound:	Default true		bool	
explodeEffect		explodeEffect: smallExplosion, CUSTOM.myExplodeEffect	effect ref list	1.13
explodeEffectOnShield	Use this effect if shield is active on target		effect ref list	1.13
teamColorRatio	Mix 0-1 of team colour into color field		float	1.13
drawUnderUnits			bool	1.13
effectOnCreate			effect ref list	1.13
shouldRevealFog	Reveal fog to player on explode		bool	1.13
alwaysVisibleInFog			bool	1.13
nukeWeapon	Shows on mini-map when fired. Some other side effects as well.		bool	1.13
trailEffect	true for built-in defaults, but can also point to any custom effects		bool/effect	
trailEffectRate	Defaults to 3		float	1.13
lightSize:			float	
lightColor		lightColor: #ffe92b	color	
largeHitEffect:	Creates a large explosion and accompanying sound on hit (only cosmetic)		bool	
lightingEffect:	Draw as lighting works best with instant:true		bool	
laserEffect:	Draw as laser works best with instant:true		bool	
Section [movement]				
Code	Description	Example	Value Type	
movementType:	Defines what kind of terrain the unit will be able to move over, along with other properties related to unit	movementType: LAND	string	
slowDeathFall:	Used with large aircraft. Makes the unit fall slowly while maintaining its speed at the time of death.	slowDeathFall: true	bool	
moveSpeed:	Maximum movement speed of the unit.	moveSpeed: 1.2	float	
moveAccelerationSpeed:	Defines how fast units accelerate to max speed.	moveAccelerationSpeed: 0.07	float	
moveDecelerationSpeed:	Don't make this too low or units will have trouble stopping at waypoints	moveDecelerationSpeed: 0.17	float	
reverseSpeedPercentage:	0.6 default. Over 0.4 will reverse for short distances (at 40% speed). If set to 1 will drive in reverse same speed	reverseSpeedPercentage: 0	float	
landOnGround:	Should flying unit land when idle.	landOnGround: false	bool	
targetHeight:	Defaults to 0 but if AIR movementType default is 35	targetHeight: 25	float	
targetHeightDrift:	Smooth animated height change. Defaults to 0 but if AIR movementType default is 1.5	targetHeightDrift: 1	float	
startingHeightOffset				
maxTurnSpeed:			float	
turnAcceleration:			float	
moveSlidingMode:			bool	
moveIgnoringBody:			bool	
moveSlidingDir:			int	
joinsGroupFormations:	Defaults to true. Changing not recommended		bool	
Section [ai]				
Code	Description	Example	Value Type	
useAsBuilder:	Set to true if unit can build or repair buildings. Defaults to [core]sBuilder.		bool	
disableUse:	Disallow AI building this unit or building		bool	
ai_upgradePriority	Defaults to 0.06. Set between 0-1, higher means AI is more likely to upgrade this unit before others		float	
Buildings only				
buildPriority:	0-1. AI uses 0.8 for first land factory, 0.48 for air factory, 0.47 for first turret.		float	
noneInBaseExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the AIs base		float	
noneGlobalExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the any where on the map		float	
upgradedFrom:	Create link to another unit to preserve max counts for upgraded and non-upgraded types in same base.		string	
maxGlobal:			int	
maxEachBase:			int	
Section [leg_#] / [arm_#]				
Code	Description	Example	Value Type	
x:	Sets position of the foot on the X axis.		float	
y:	Sets position of the foot on the Y axis.		float	
copyFrom:	Copy from another leg. Useful to only need to set leg values once	copyFrom: 1	int	
attach_x:	Sets the leg's attach point on the X axis.		float	
attach_y:	Sets the leg's attach point on the Y axis.		float	
rotateSpeed:			float	
lockMovement	Lock to unit body. Useful if walking unit converted to a flying unit.		bool	
heightSpeed:			float	
moveSpeed				
moveWarmUp				
holdDisMin:	Defaults to 7. Reposition leg at this distance if neighbor legs are not already repositioning.		float	
holdDisMax:	Defaults to 16. Force reposition of leg at this distance.		float	
holdDisMin_maxMovingLegs				
hold_moveOnlyIfFurthest				
holdDisMin_checkNeighbours				
hardLimit:	Defaults to 50. Force leg to never go this far. Better to not be reached.		float	
estimatingPositionMultiplier	defaults to 1. Predicts where unit will be for leg placement based on unit speed.		float	
Graphics and effects				
hidden:			bool	
image_end			image	1.13
image_end_shadow			image	1.13
image_end_teamColors				1.13
image_foot	same as image_end		image	1.13
image_foot_shadow			image	
image_middle			image	1.13
image_leg	same as image_middle		image	
draw_foot_on_top			bool	1.13
drawOverBody	Draw over the body of unit, defaults of false		bool	
drawUnderAllUnits	Draw over all units		bool	
drawDirOffset			float	
dust_effect:	Spawns dust particles on each step.		bool	
spinRate	Makes arm/leg spin, like idleSpin for turrets		float	

favourOppositeSideNeighbours	calculate neighbours with X 10 times closer than Y			
resetAngles	Unused		float	
<b>Section</b> <i>[effect_NAME]</i>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
life	Defaults 200. Time till effect is removed. Set low as possible to reduce effect overhead.		float	1.13
alsoEmitEffects	Create more effects when created, useful for meta-effects. Note: other 'alsoEmitEffects' on created effects are ignored.		effect ref	1.13
alsoPlaySound			sound ref	1.13
createWhenOffscreen	Defaults false.		bool	1.13
createWhenZoomedOut	Defaults true		bool	1.13
createWhenOverLiquid	Defaults true		bool	1.13
createWhenOverLand	Defaults true		bool	1.13
spawnChance	Default 1. If less than 1 effect only has a random chance of being created		float	1.13
showInFog	Default false		boolean	1.13
delayedStartTimer	Hide for x time before showing and updating effect.		float	1.13
liveAfterAttachedDies	Defaults false when attachedToUnit is being used		bool	1.13
priority	Defaults to high. verylow/low/high/veryhigh/critical. Takes effect when too many effects are being shown at once.		string	1.13
	<b>Movement</b>			
attachedToUnit	Attach to unit or projectile that created this effect. Will move with this object.		bool	1.13
alwaysStartDirAtZero	Ignore source/attached unit dir		bool	1.13
atmospheric	Apply drag to slow this effect down and add small wind effects		bool	1.13
physics	Fall to ground and bounces. Needs height to take effect.		bool	1.13
physicsGravity	Defaults to 1. height speed acceleration when physics: true		float	1.13
xOffsetRelative	Offset starting effect position. Relative to direction of attached turret, projectile, unit		float	1.13
yOffsetRelative	Offset starting effect position. Relative to direction of attached turret, projectile, unit		float	1.13
xOffsetRelativeRandom	Random offset by +/- this value		float	1.13
yOffsetRelativeRandom	Random offset by +/- this value		float	1.13
xOffsetAbsolute	Offset starting effect by position ignoring direction of attached turret, projectile, unit		float	1.13
yOffsetAbsolute	Offset starting effect by position ignoring direction of attached turret, projectile, unit		float	1.13
xOffsetAbsoluteRandom	Random offset by +/- this value		float	1.13
yOffsetAbsoluteRandom	Random offset by +/- this value		float	1.13
xspeedRelative			float	1.13
yspeedRelative			float	1.13
xspeedRelativeRandom	Randomly change by -value to value		float	1.13
yspeedRelativeRandom	Randomly change by -value to value		float	1.13
xspeedAbsolute			float	1.13
yspeedAbsolute			float	1.13
xspeedAbsoluteRandom	Randomly change by -value to value		float	1.13
yspeedAbsoluteRandom	Randomly change by -value to value		float	1.13
hOffset	height offset from source		float	1.13
hOffsetRandom	Randomly change by -value to value		float	1.13
hSpeed	height speed		float	1.13
hSpeedRandom	Randomly change by -value to value		float	1.13
dirOffset	rotation		float	1.13
dirOffsetRandom	Randomly change by -value to value		float	1.13
dirSpeed	rotation speed		float	1.13
dirSpeedRandom	Randomly change by -value to value		float	1.13
	<b>Graphics</b>			
frameIndex	frame of to use		int	1.13
striplIndex	A built-in image set to use. Cannot be used with custom image	effects/explode_big/light_50/flame/effects/effects2/projectiles/projectiles2/explode_b	int/string	1.13
Image	Custom image file to use. Cannot be used with striplIndex.		image	1.13
imageShadow	Custom image file to use for shadows		image	
scaleTo	Defaults to 1		float	1.13
scaleFrom	Defaults to 1		float	1.13
color	Defaults #FFFFFFF		color	1.13
drawUnderUnits			bool	1.13
fadeInTime	Fade alpha from 0% to 100% for this time at start		float	1.13
fadeOut	Fade alpha from 100% to 0% based on life. Set alpha is higher than 1 to delay fade		bool	1.13
alpha	Capped between 0-1. Can be set higher than 1 to delay fadeOut effects			
shadow	True to draw a shadow. Forced true if imageShadow is used			
	<b>Animation</b>			
total_frames	Total frames of 'image', used with animation or frameIndex. Only needed with custom images		int	1.13
animateFrameStart			int	1.13
animateFrameEnd			int	1.13
animateFramePingPong			int	1.13
animateFrameSpeed			time	1.13
animateFrameSpeedRandom			time	1.13
<b>Section</b> <i>[animation_NAME]</i>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
onActions : Unknown	move, attack, idle, underConstruction, underConstructionWithLinkedBuiltTime, queuedUnits			1.13
onActionsQueuedUnitPlayAt : float	For onAction: queuedUnits. Amount queue needs to reach before starting, set between 0-1			1.13
blendIn : time	Blend with last animation for this time			1.13
blendOut : time	Blend with next animation for this time			1.13
pingPong	Play animation in reverse after it ends		bool	1.13
KeyframeTimeScale : float	Scales all keyframe times, useful to make an animation faster/slower without changing everything		float	1.13
	<b>Keyframes - create as many as needed</b>			
arm#_[time]	Adds a keyframe at time. Use multiple times to create animation.	eg: arm1_5s: {x: 5, dir: 90 }		1.13
leg#_[time]	Adds a keyframe at time. Use multiple times to create animation.			
body_[time]	Adds a keyframe at time for body. Only frame and scale allowed on body	eg: body_4s: {frame: 4, scale: 0.5}		
	<b>Deprecated Keys (can be used but there are better ways)</b>			
start--int	Start-image-frame--deprecated			1.13
end--int	End-image-frame--deprecated			1.13
scale_start--float	Start-scale--Deprecated--use-body-keyframes-instead-			1.13
scale_end--float	End-scale--Deprecated--use-body-keyframes-instead-			1.13
speed--float	Speed, smaller-is-faster. Only effects start, end, scale_start, scale_end			1.13
<b>Section</b> <i>[action_NAME]</i>				
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Value Type</b>	
text	Text shown in UI		string	
description	A display text when you select your unit's action, used to explain it's purpose.		string	
displayType	none, rally, upgrade, queueUnit, building, action, infoOnly, infoOnlyNoBox			
displayRemainingStockpile	Queue is shown as number of times action can be triggered based on price		bool	
pos	Order action appears in UI		float	
	<b>Requirements to trigger or enable</b>			



price	The price of your action for the unit. Disables action if not available. Defaults to credits if unlabelled	price: credits=5, energy=5, hp=100, shield=5, ammo=1	resources
isActive	Defaults true. If false then action is disabled and shown in red in UI.		LogicBoolean
isVisible	Defaults true. If false action is hidden from UI and disabled.		LogicBoolean
isLocked	Defaults false. If true action is disabled, and a lock icon is shown. Mostly used for no nuke game modes		LogicBoolean
isLockedMessage			string
onlyOneUnitAtATime	When action is picked in UI, only one unit selected with get this action. Defaults to false.		bool
autoTrigger	When true triggers the effects of this action instantly (ignoring price, isActive, isVisible, buildSpeed, etc)	autoTrigger: if self.overWater()	LogicBoolean
ai_isDisabled	Defaults false. Stop AI using this action.		LogicBoolean
	<b>While action is queued</b>		
buildSpeed		buildSpeed: 5s	time
highPriorityQueue	Defaults to false. If true this action skips all other low priority actions in queue. Useful for fireTurret actions.		bool
canPlayerCancel			bool
whenBuilding_cannotMove	Stops unit moving while action is being applied. Useful for deploy like actions.		bool
whenBuilding_playAnimation	Play a custom animation from an [animation] section when in active queue		animation ref
whenBuilding_rotateTo	Rotate unit body to this direction when action is in active queue		float
whenBuilding_rotateTo_orBackwards	If true allow rotation in 180 degrees from whenBuilding_rotateTo when this is a smaller angle		bool
whenBuilding_rotateTo_waitTillRotated	Pause action queue till rotation is finished		bool
whenBuilding_temporarilyConvertTo	Convert to another unit while action is in active queue. Note: actions from the original unit will be kept		unit ref
spawnEffectsOnQueue	Effects to spawn at unit when action is first added to queue		effect ref
playSoundToPlayerOnQueue	Global sound to play to unit's player only when action is first added to queue		sound ref
	<b>Results (What happens) - Must be at least one result listed</b>		
convertTo	Convert your unit into another unit. properties are preserved.		unit ref
fireTurretXAIGround	When action finishes fire target turret at point on ground, bypasses canShoot rules in turret.	fireTurretXAIGround: nukeSilo	turret ref
fireTurretXAIGround_withOffset	If not set player targets the ground with GUI, if a point is set this step is skipped	fireTurretXAIGround_withOffset: 0.0	point
fireTurretXAIGround_withProjectile	Used with fireTurretXAIGround. Defaults to target turret's normal projectile.		projectile ref
addEnergy	Adds energy to unit. Has no effect unless energyMax is set		float
addResources	Add these resources when action finishes.	addResources: credits=5, energy=-5, hp=-100, shield=5, ammo=1	resources
alsoTriggerAction	Trigger to results of another action as well. Ignores action's requirements		action ref
alsoQueueAction	Adds another action into the normal unit's queue. Ignores action's requirements		action ref
alsoTriggerOrQueueActionConditional	Defaults true. alsoTriggerAction and alsoQueueAction are ignored if this works out to be false.		LogicBoolean
spawnEffects	Effects to spawn at unit		effect ref
playSoundAtUnit	Local sound to play when action finishes		sound ref
playSoundGlobally	Global sound to play to all players in game		sound ref
playSoundToPlayer	Global sound to play to unit's player only		sound ref
addUnitsIntoTransport	Add units into transport, use self.transportingCount() to check for space before adding	addUnitsIntoTransport: tank*3, heavyTank	
resetCustomTimer	Reset timer used with self.customTimer()		LogicBoolean
<b>Type</b>	<b>LogicBoolean</b>		
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Targets</b>
true			
false			
if	Start all logic booleans with if, unless just using true/false		
and		if self.isInWater() and self.energy(greaterThan=1)	
or		if (self.energy(greaterThan=1) or self.ammo(greaterThan=1)) and self.isFlying()	
not		if not self.isOverLiquid()	
	<b>Unit location and movement</b>		
self.isUnderwater()			
self.isAtGroundHeight()			
self.isFlying()			
self.isMoving()			
self.isAtTopSpeed()			
self.isInWater()	Touching water		
self.isOverwater()	Touching or over a water tile		
self.isOverLiquid()			
self.isOverCliff()			
self.isOverPassableTile()	(parameters: type)		
self.isOverOpenLand()	shortcut for self.isOverPassableTile(type="LAND")		
	<b>Unit stats</b>		
self.hp()	(parameters: greaterThan, lessThan, empty, full)		
self.ammo()	(parameters: greaterThan, lessThan, empty, full)		
self.isAmmoEmpty()	shortcut for self.ammo(empty=true)		
self.ammoIncludingQueued()	Also includes ammo from actions still in queue (parameters: greaterThan, lessThan, empty, full)	if self.ammoIncludingQueued(lessThan=12)	
self.energy()	(parameters: greaterThan, lessThan, empty, full)		
self.energyIncludingQueued()	Also includes energy from actions still in queue (parameters: greaterThan, lessThan, empty, full)		
self.isEnergyFull()	shortcut for self.energy(full=true)		
self.isEnergyEmpty()	shortcut for self.energy(empty=true)		
self.shield()	(parameters: greaterThan, lessThan, empty, full)		
	<b>Misc</b>		
game.nukesEnabled()			
self.hasFlag()	Boolean flag saved into units for mods to use. (parameters: id=0-31)	Use addResources in action change this vaule	
self.tags()	(parameters: includes)		
self.transportingCount()	(parameters: greaterThan, lessThan, empty)		
self.isAttacking()			
self.transportingUnitWithTags()	(parameters: includes)	self.transportingUnitWithTags(includes="human")	
self.hasTakenDamage()	(parameters: withinSeconds=X, laterThanSeconds=X)	self.hasTakenDamage(withinSeconds=1)	
self.timeAlive()	(parameters: withinSeconds=X, laterThanSeconds=X)		
self.lastConverted()	(parameters: withinSeconds=X, laterThanSeconds=X)		
self.customTimer()	(parameters: withinSeconds=X, laterThanSeconds=X)		
<b>Type</b>	<b>Resources - used by addResources, price, etc</b>		
<b>Code</b>	<b>Description</b>	<b>Example</b>	<b>Targets</b>
credits	Global resource		
energy			
hp			
shield		price: hp=-100, shield=100	
ammo	Hidden value on each unit for use by mods		
setFlag	use with addResources, resourceUsage or price. 0-31. Flags are stored in each unit	setFlag=1	
unsetFlag	use with addResources, resourceUsage or price. 0-31		
hasFlag	use with price or resourceUsage		
hasMissingFlag	use with price or resourceUsage		

*This sheet is for 1.12, for 1.13.2 see sheet tabs below.*

## Section *[core]*

Code	Description	Example	Value Type
<b>name:</b>	Defines the raw name for your unit, in which case the game uses to identify as a unique name. (This name is not displayed in-game)	name: customTank_1	string
<b>class:</b>	Reserved for future use, must be CustomUnitMetadata	class: CustomUnitMetadata	string
<b>price:</b>	The cost of your unit from builders/buildings.	price: 500	int
<b>maxHp:</b>	The max health for your unit. (will spawn with this value)	maxHp: 200	int
<b>mass:</b>	The 'weight' of your unit, this will define how it collides with other units, a greater value means it is tougher to push.	mass: 3000	int
<b>techLevel:</b>	Defines the Tech Level of your unit, there are 3 levels and each will appear in a different color in the GUI.	techLevel: 1	int
<b>buildSpeed:</b>	Time it takes to build unit (may multiply with builder speed) Formulae for seconds: $10 + 60 = 0.16$ (0.16 will make it take 10 seconds to build)	buildSpeed: 0.16	float
<b>radius:</b>	Circular area around your unit that makes it selectable (mouse click/screen touch)	radius: 20	int
<b>isBio:</b>	Choose whether your unit is biological or not, will effect sound and splat (unless hideScorchMark is true)	isBio: false	bool
displayText:	The name of your unit that the game shows to the player.	displayText: Custom Tank	string
displayDescription:	Description of your unit that the game shows to the player.	displayDescription: - Fast movement.\n- Light damage.	string
copyFrom:	Use unit data from another ini file as base	copyFrom: customTank_Default	file (ini)
dont_load:	Do not load unit, and don't error on missing data. Can be useful when used with copyFrom	dont_load: true	bool
altNames:	comma separated list of names. Like name but lower priority, useful for multiple optional mods.	altNames: custTank1, customTank1, cTank1	string(s)
overrideAndReplace:	Override another unit with this unit. Build links pointing to target unit will be replaced with this unit. And this unit will replace target on maps.	overrideAndReplace: builder, combatEngineer	string(s)
globalScale:	Defaults to 1. Changing not recommend	globalScale: 1	float
displayLocaleKey:	Translation file key for unit name and description	displayLocaleKey: units.mechArtillery	string
showInEditor:	Set to false to hide unit in sandbox editor (Default true)	showInEditor: true	bool
isLocked:	Disallow building of this unit. Can be used with overrideAndReplace to restrict units the player is allowed to build.	isLocked: false	bool
experimental:	Tag unit as experimental. Affects zoomed out icon and end game stats.	experimental: false	bool
generation_credits:	Income unit creates	generation_credits: 2	int
generation_delay:	Defaults to 40, recommended only changing generation_credits	generation_delay: 40	int
stayNeutral:	Set to false to disable capture when unit is on the neutral team	stayNeutral: false	bool
resourceRate:	Used with canReclaimResources. Experimental feature that is not yet finished.		float
maxShield:			int
startShieldAtZero:			bool
shieldRegen:			float
shieldDisplayOnlyDeflection:	Hide shield (if active) unless deflecting shot		bool
shieldDeflectionDisplayRate:	Defaults to 4. High value causes shield deflection to fade disappear faster		float
energyMax:	Defaults to 0. Energy can be used as ammo for turrets or for laser defense		float
energyRegen:			float
energyStartingPercentage:			float
energyNeedsToRechargeToFull:	Disable weapons using energy after reaching zero till fully recharged		float
displayRadius:	Defaults to radius value. Set to show a larger or smaller selection box ui on units.	displayRadius: 20	int
buildingSelectionOffset:			int
buildingToFootprintOffsetX:			float
buildingToFootprintOffsetY:			float
fogOfWarSightRange:	Set number of tiles this unit can see. Defaults to 15	fogOfWarSightRange: 15	int
exit_x:	Defaults to 0	exit_x: 0	float
exit_y:	Defaults to 5. Controls were newly created units appear.	exit_x: 5	float
softCollisionOnAll:			int
isBug:			bool
hideScorchMark:			bool
isBuilding:			bool
placeOnlyOnResPool:	Normally used for extractors		bool
canRepairBuildings:	([ai]useAsBuilder:true is required as well right now)		bool
canRepairUnits:	([ai]useAsBuilder:true is required as well right now)		bool
autoRepair:	Automatically try and repair damaged units in range ([ai]useAsBuilder:true is required as well right now)		bool
nanoRange:			int
nanoRepairSpeed:			float
nanoBuildSpeed:			float
nanoFactorySpeed:			float
selfRegenRate:	Passive self repair rate		float
selfBuildRate:			float
dieOnConstruct:	Delete this unit when it starts making a building. Target building likely will need selfBuildRate set to be created without a builder.		bool
numBitsOnDeath:			int

nukeOnDeath:			bool
nukeOnDeathRange:			float
nukeOnDeathDamage:			float
nukeOnDeathDisableWhenNoNuke:	Defaults to false		bool
<del>fireOnDeath:</del>			int
canReclaimResources:	Used with resourceRate. Experimental feature that is not yet finished.		bool
maxTransportingUnits:	Number of slots this units as for transporting other units.		int
transportSlotsNeeded:	Defaults to 1. Number of slots this unit uses up in a transport, Experimentals are often set to 5 to stop small transports holding them.		int
builtFrom_#_name:			string
builtFrom_#_pos:	Order action appears in UI		string
builtFrom_#_forceNano:	Build as if it was a building even if it's a unit.		bool
canBuild_#_name:	(Note: canBuild and builtFrom have the same effect just in the opposite direction. Only one is needed to create a link.)		string
canBuild_#_pos:	Order action appears in UI		string
canBuild_#_tech:			int
canBuild_#_forceNano:	Build target as if it was a building even if it's a unit.		bool
canBuild_#_type:			string
action_#_convertTo:	Convert your unit into another unit. (All your sub actions will be linked to the # you use) (due to bug target must have the same number of legs)	action_1_convertTo: customTank_2	string
action_#_pos:	Order action appears in UI		float
action_#_price:	The price of your action for the unit. (All your sub actions will be linked to the # you use)	action_1_price: 1000	int
action_#_text:	A display text when you select your unit's action, used to explain it's purpose. (All your sub actions will be linked to the # you use)	action_1_text: Upgrade to Custom Tank 2	string
action_#_description:			string
action_#_addEnergy:	Adds energy to unit. Has no effect unless energyMax is set		float
action_#_whenBuilding_cannotMove:	Stops unit moving while action is being applied. Useful for deploy like actions.		bool

## Section

[graphics]

Code	Description	Example	Value Type
total_frames	Defaults to 1. Animations require this.		int
frame_width:	Calculated for you if total frames is set, but can be overridden		int
frame_height:	Defaults to image height		int
image_offsetX:			int
image_offsetY:			int
teamColorsUseHue:	False: Green pixels on unit gets converted to team color. True: Whole unit is tinted the team colour. Defaults to false		bool
imageSmoothing:			bool
image:			file (image)
image_back:			file (image)
image_wreak:			file (image)
image_turret:			file (image)
image_shadow:	Image file, NONE, or AUTO. AUTO will use image and make it transparent black only.		file (image)
shadowOffsetX:			float
shadowOffsetY:			float
teamColorsOnTurret:	Defaults false. Apply team colours on turret as well. (Only works on image_turret right now, not image on each turret)		bool
scaleImagesTo:			float
imageScale:			float
scaleTurretImagesTo:			float
splstEffect:			bool
dustEffect:			bool
splstEffectReverse:			bool
dustEffectReverse:			bool
drawLayer:	Set to experimentals, or leave line out		string
lock_body_rotation_with_main_turret:	Locks body image locked to first turret's direction		bool
lock_shadow_rotation_with_main_turret	Locks body image shadow locked to first turret's direction		bool
rotate_with_direction:	Defaults to true. Makes unit body image locked to 0 degrees when false. Often used with animation_direction_*		bool
animation_direction_units:	45 for 8 directions, 90 for 4 direction animation. Used with rotate_with_direction:false		float
animation_direction_strideX:	Animation frames to offset on direction change.		int
animation_direction_strideY:	Animation frames to offset on direction change. Used with frame_height.		int
animation_direction_starting:	Direction for first frame		float
animation_TYPE_start:	Animation start frame, from 0. Set total_frames to use (TYPE is moving/idle/attack)	animation_moving_start: 0	int
animation_TYPE_end:	End frame, must be larger then start	animation_moving_end: 3	int
animation_TYPE_scale_start:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float
animation_TYPE_scale_end:	Scale unit image. Defaults to 1. Useful for bio units or breathing effects.		float
animation_TYPE_speed:	Delay for each frame of animation. Larger values cause slower animation		float
animation_TYPE_pingPong:	Play animation in reverse before repeating. Useful with scale_start/scale_end		bool

## Section

[attack]

Code	Description	Example	Value Type
turretSize:	(multiplied by globalScale)		float
turretTurnSpeed:			float
turretRotateWithBody:			bool
attackMovement:	normal/bomber		string
disablePassiveTargeting:	Unit only attacks manually ordered target. Rarely needed.		bool
attackMovementSpeed:	Unused		float
maxAttackRange:	(multiplied by globalScale)		float
shootDelay:	Global delay, can also use delay on each turret		float
isMelee:	Used with a low attack range (like maxAttackRange: 9) makes src and target radius get added to range, and effects AI.		bool
dieOnAttack:			bool
canAttack:	If set to false, can not attack any unit. Regards of other canAttack options below.		bool
canAttackFlyingUnits:	can also be narrowed per turret		LogicBoolean
canAttackLandUnits:	can also be narrowed per turret		LogicBoolean
canAttackUnderwaterUnits:	can also be narrowed per turret		LogicBoolean
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos. (can also be set per turret)		LogicBoolean
isFixedFiring:	Must aim body at target to shoot. Will often make the unit need to stop before it can aim and shoot.		bool
aimOffsetSpread:			bool

## Section *[turret\_#]*

Code	Description	Example	Value Type
x:			float
y:			float
copyFrom:	Copy all values from another turret as default values for this turret	copyFrom: 1	int
idleDir:			float
linkDelayWithTurret:			int
delay:	Override global shootDelay for this turret		float
warmup:			float
warmupCallDownRate:			float
warmupNoReset:			bool
warmupShootDelayTransfer:			float
recoilOffset:	Amount to offset turret after firing		float
recoilOutTime:	How long to get to offset value		float
recoilReturnTime:	How long to return back to default position.		float
energyUsage:	Required energy to fire weapon		float
attachedTo:	Id of another turret to attach to, will be positioned relative to it.		int
canShoot:	Defaults to true		bool
shoot_sound:			string
shoot_sound_vol:			float
shoot_flame:	Current types are: small, large, smoke, shockwave		string
shoot_light			color
slave:	Locks this turret's direction and shot cooldown to attached turret. Often used with warmup for multiple barrel guns		bool
invisible:	Don't render this turret, but still can shoot, etc		bool
canAttackFlyingUnits:	Narrows targeting for this turret, note targeting for the whole unit in [attack] is applied first. (so you can only use this to target less not more)		LogicBoolean
canAttackLandUnits:			LogicBoolean
canAttackUnderwaterUnits:			LogicBoolean
canAttackNotTouchingWaterUnits:	Default true. If false unit can only attack units in contact with the water. Used for units with torpedos.		LogicBoolean
canAttackCondition:	Normally, used to optionally disable a turret based on a LogicBoolean. Eg: this unit's height	canAttackCondition: if not self.flying	LogicBoolean
size:			float
limitingRange:	Make this turret have less range than the maxAttackRange. Do not apply this to all turrets change maxAttackRange instead.		float
laserDefenceEnergyUse:	Set to enable a projectile laser defence from this turret. Should also set the energyMax in core.		float
shouldResetTurret:	Defaults true. False to disable the resetting turret angle when idle		bool
idleSpin:	Spin rate when idle, used on missile turrets		float
isMainNanoTurret:	Defaults to false. Turret to use for creating buildings, etc. should only be true on one turret, and should have canShoot set to false		bool
image:	Use custom image. Overrides unit's main turret image		file (image)
chargeEffectImage:	Used with warmup. Shows a scaling effect image on turret barrel when charging.		file (image)

## Section *[projectile\_#]*

Code	Description	Example	Value Type
directDamage:	Damage to target unit on hit. Does not work with targetGround.		int
areaDamage:	directDamage or areaDamage is required		int
areaRadius:	How wide areaDamage effects. Note this drops off.		
deflectionPower:	Defaults to 1. Energy needed for laser defence to deflect. -1 to disable deflection (only disable for special weapons like flames)		float

targetGround:	Target ground, and don't home in on target. Only areaDamage is applied if targeting ground.		bool
life:	How long till this projectile gets removed if it hasn't hit a target, 300 might be a good starting point, change depending on speed and range		int
speed:			float
instant:	Hit target instantly		bool
instantReuseLast:	Recycles last projectile fired, only one projectile ever exists. Can turn lasers into beam weapons by using lower rate of fire and setting this to true		bool
ballistic:			bool
trailEffect:			bool
lightSize:			float
lightColor:		lightColor: #ffe92b	color
largeHitEffect:			bool
lightingEffect:			bool
laserEffect:			bool
ballistic_delaymove_height:			float
ballistic_height:			float
targetSpeed:	Accelerate to this speed		float
autoTargetingOnDeadTarget:	Retarget to nearby unit if target dies while in transit		bool
drawSize:	Scale image. Defaults to 1		float
color		color: #bebe50	color
image:	Use custom image. Overrides drawType and frame		file (image)
drawType	Built-in image to use. 0:projectiles.png 1:projectiles_large.png 2:projectiles2.png	drawType:1	int
frame	Built-in image frame to use, starts at zero.		int
flameWeapon:			bool
hitSound:			bool
targetGroundSpread:	Randomly makes the shot inaccurate by this amount. Also used by weapons like the flamethrower		float
speedSpread:	Randomly change the starting projectile speed by this amount		float

## Section *[movement]*

Code	Description	Example	Value Type
<b>movementType:</b>			string
slowDeathFall:	Used with large aircraft		bool
moveSpeed:			float
moveAccelerationSpeed:			float
moveDecelerationSpeed:	Don't make this too low or units will have trouble stopping at waypoints		float
reverseSpeedPercentage:	0.6 default. Over 0.4 will reverse for short distances (at 40% speed). If set to 1 will drive in reverse same as forwards. Useful if slow turning		float
landOnGround:	Should flying unit land when idle.		bool
targetHeight:	Defaults to 0 but if AIR movementType default is 35		float
targetHeightDrift:	Smooth animated height change. Defaults to 0 but if AIR movementType default is 1.5		float
maxTurnSpeed:			float
turnAcceleration:			float
moveSlidingMode:			bool
moveIgnoringBody:			bool
moveSlidingDir:			int
joinsGroupFormations:	Defaults to true. Changing not recommended		bool

## Section *[ai]*

Code	Description	Example	Value Type
useAsBuilder:	Set to true if unit can build or repair buildings.		bool
disableUse:	Disallow AI building this unit or building		bool
maxGlobal:			int
maxEachBase:			int
buildPriority:	0-1. AI uses 0.8 for first land factory, 0.48 for air factory, 0.47 for first turret.		float
noneInBaseExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the AIs base		float
noneGlobalExtraPriority:	Adds to buildPriority, if this unit doesn't exist in the any where on the map		float
upgradedFrom:	Create link to another unit to preserve max counts for upgraded and non-upgraded types in same base.		string

## Section *[leg\_#]*

Code	Description	Example	Value Type
x:			float
y:			float
copyFrom:	Copy from another leg. Useful to only need to set leg values once	copyFrom: 1	int
attach_x:			float
attach_y:			float

hidden:			bool
rotateSpeed:			float
resetAngle:	Unused		float
lockMovement	Lock to unit body. Useful if walking unit converted to a flying unit.		bool
heightSpeed:			float
draw_foot_on_top:			bool
dust_effect:			bool
holdDisMin:	Defaults to 7. Reposition leg at this distance if neighbor legs are not already repositioning.		float
holdDisMax:	Defaults to 16. Force reposition of leg at this distance.		float
hardLimit:	Defaults to 50. Force leg to never go this far. Better to not be reached.		float

The spreadsheets to the right of this one are tools designed to automate specific processes so that they are not as time consuming for development.

Table of Contents

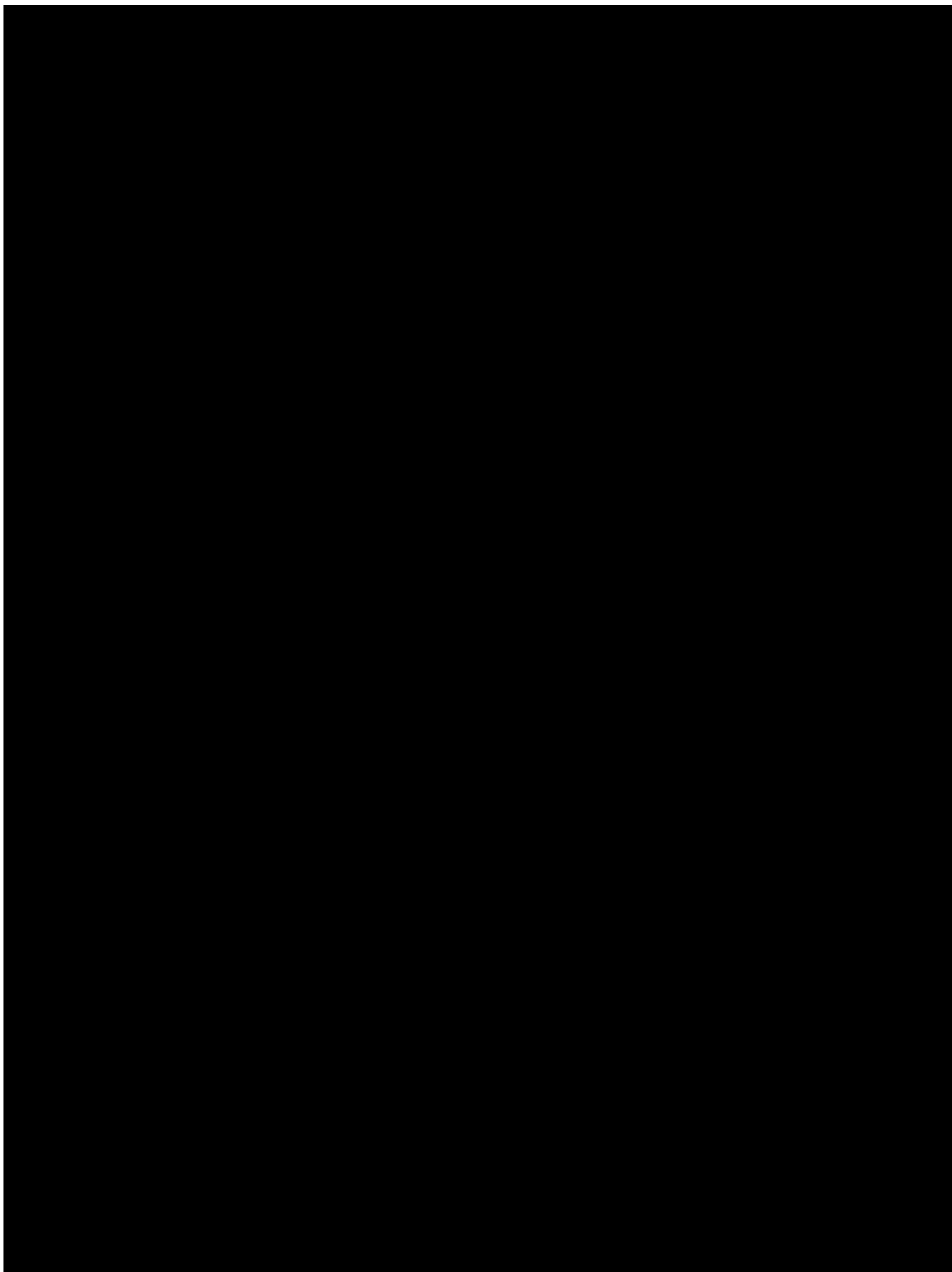
[AccessProjectList Tool](#)

A tool that automatically sets up infrastructure for Lambda.

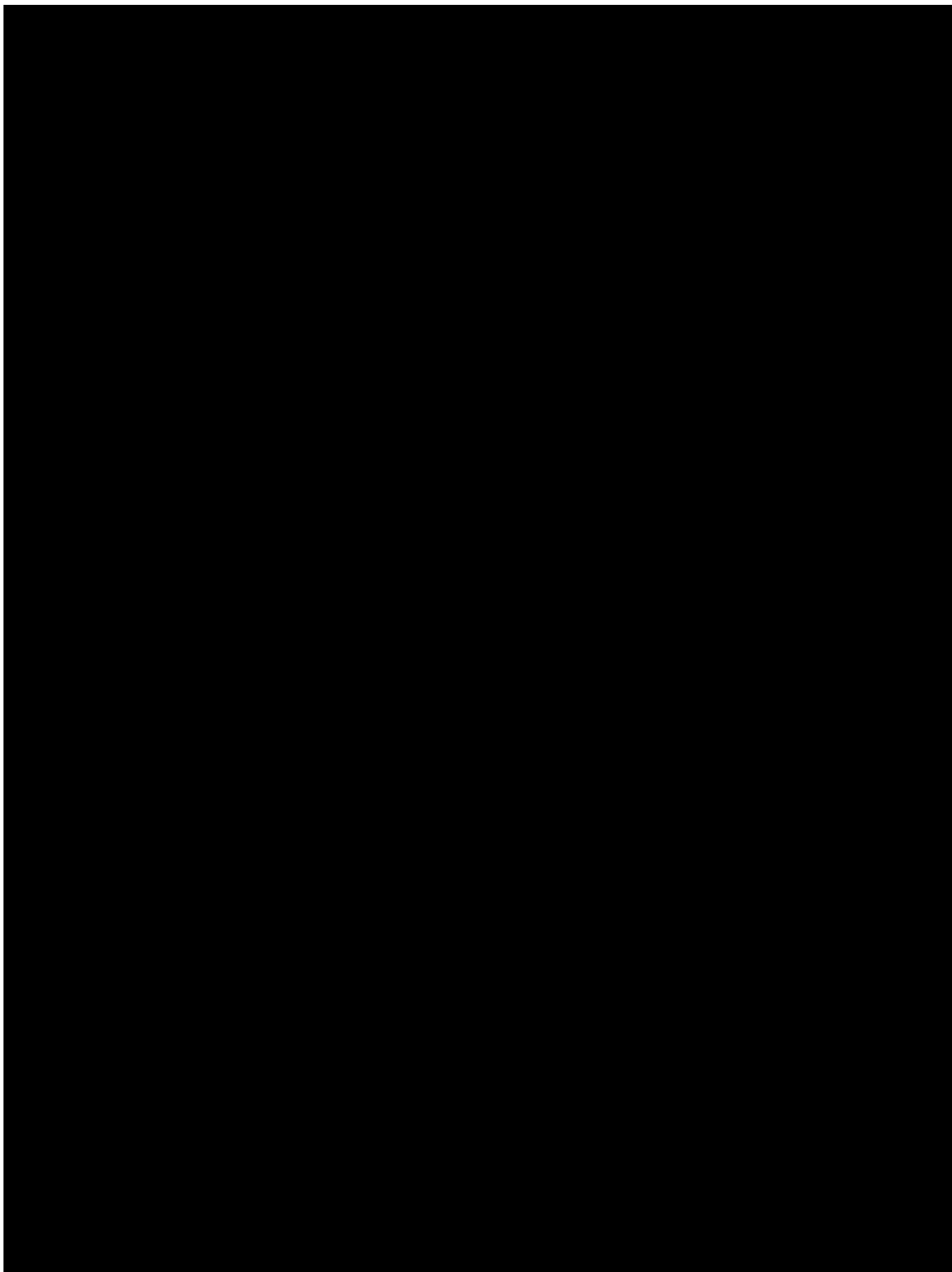
[Translation Tool](#)

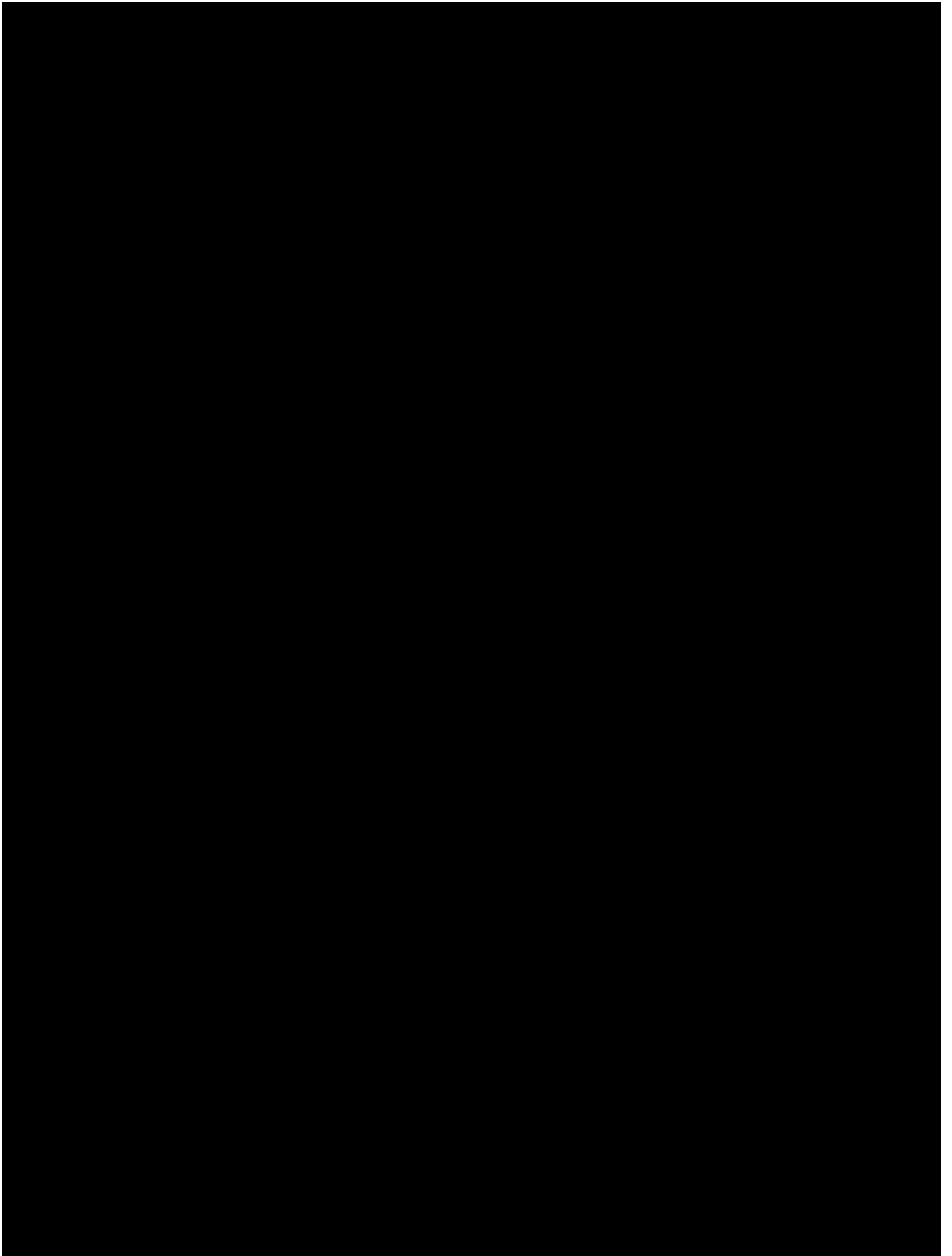
A tool that automatically compiles Transforms into Java to speed time on the...

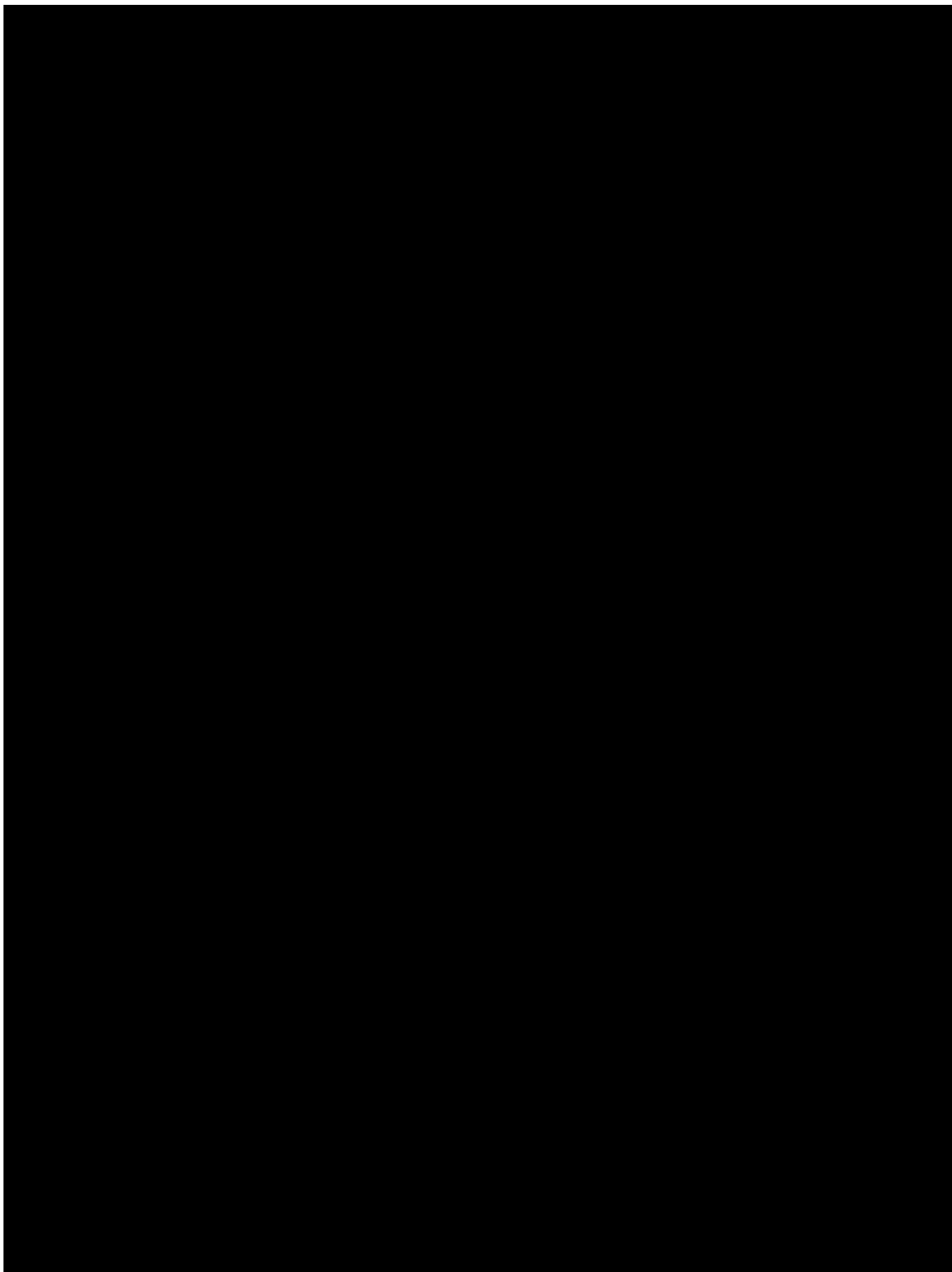
Feel free to copy and paste the spreadsheet into a new copy in order to edit it yourself, the tools are automated and will compile an Output based on the input values which can be changed in a sheet you own.

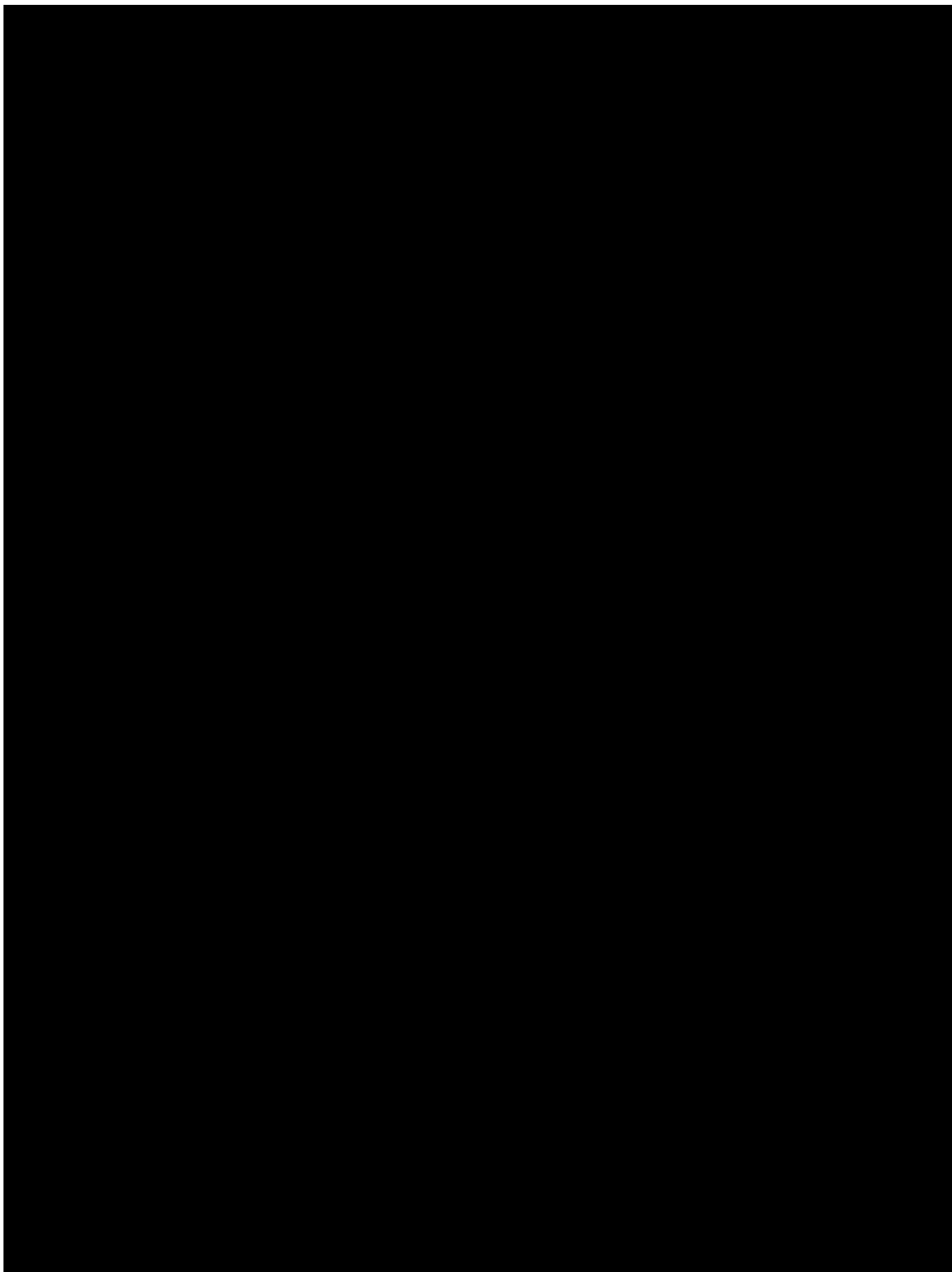












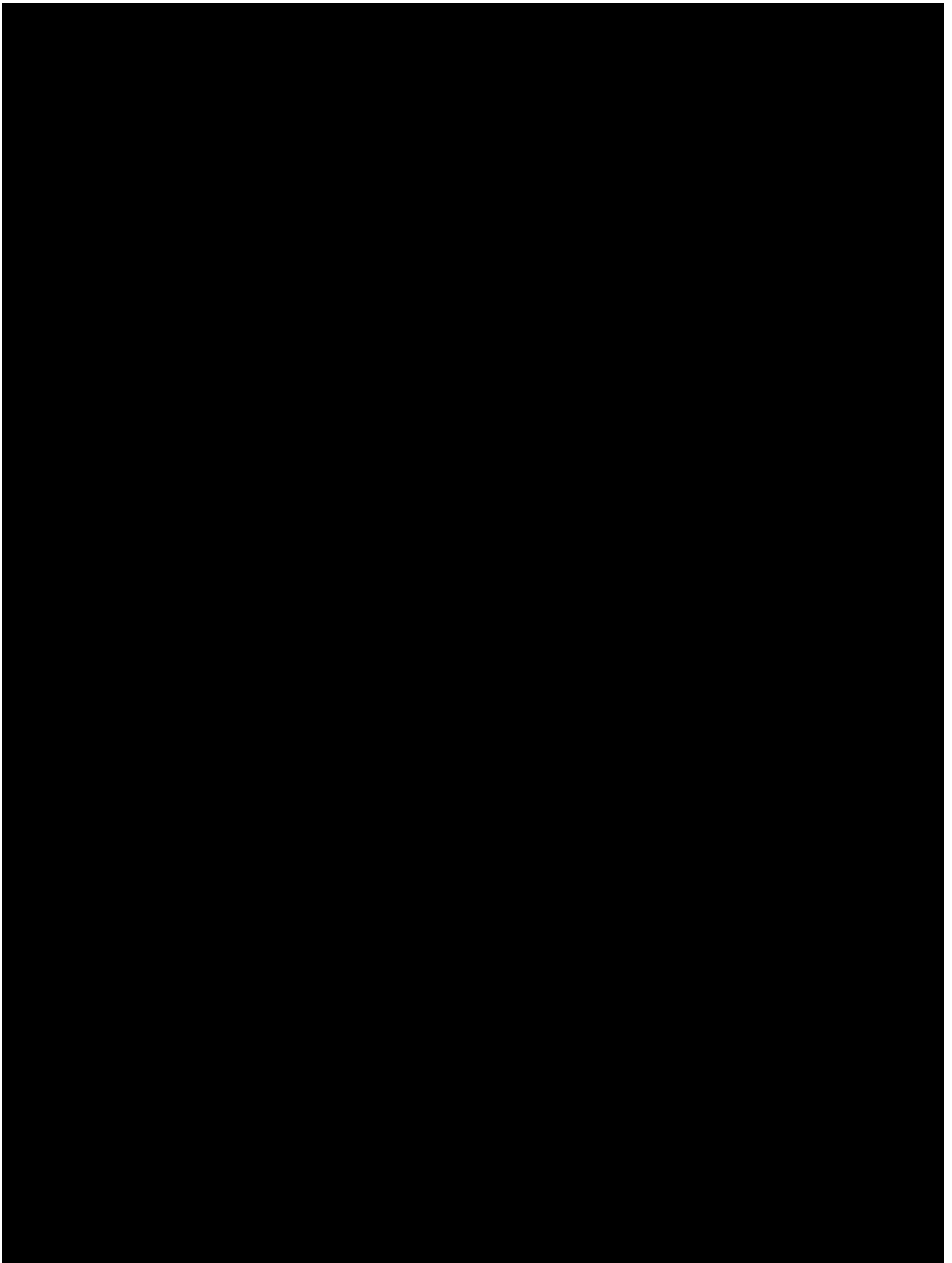
\_\_\_\_\_

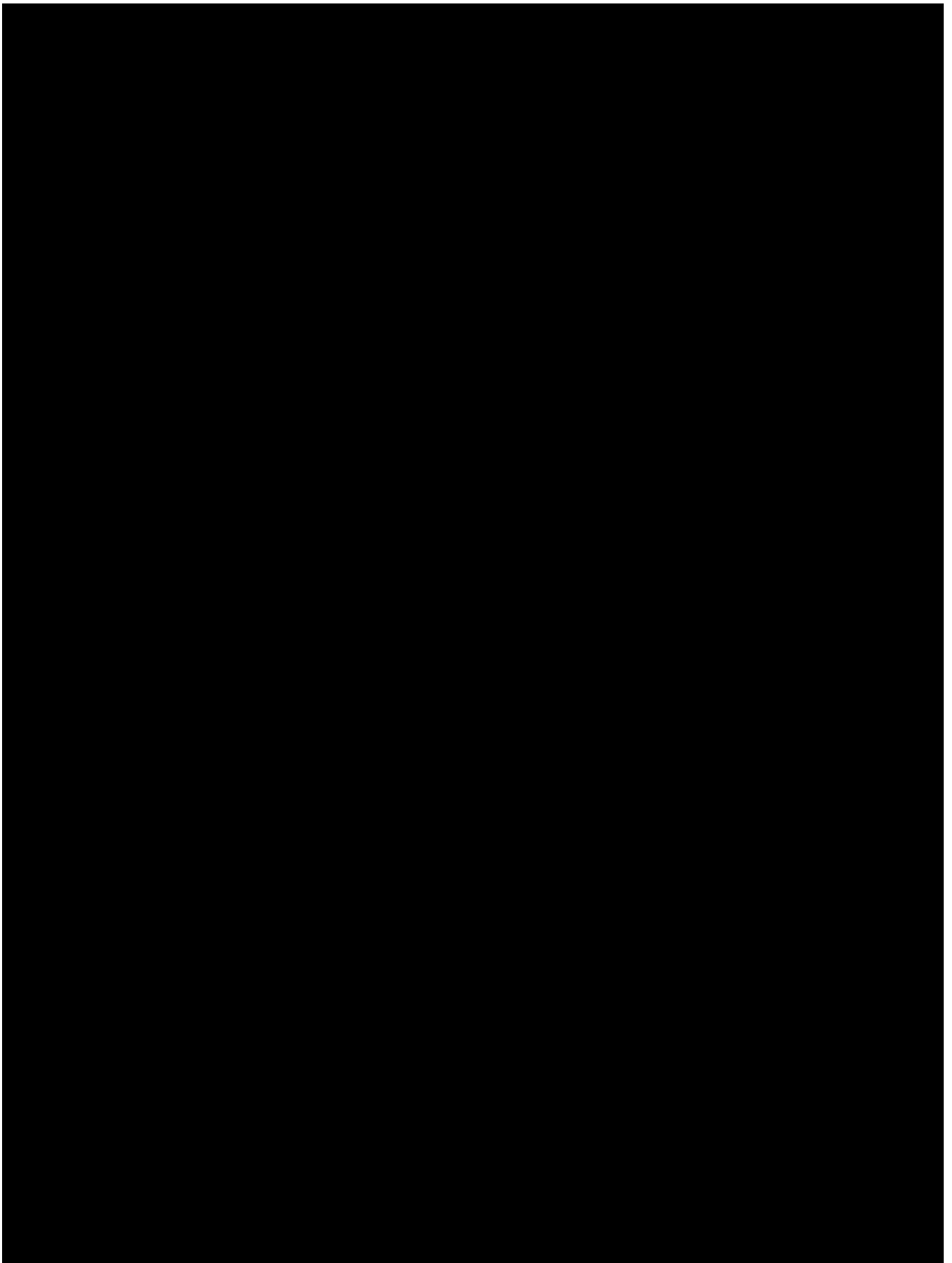
[illegible]

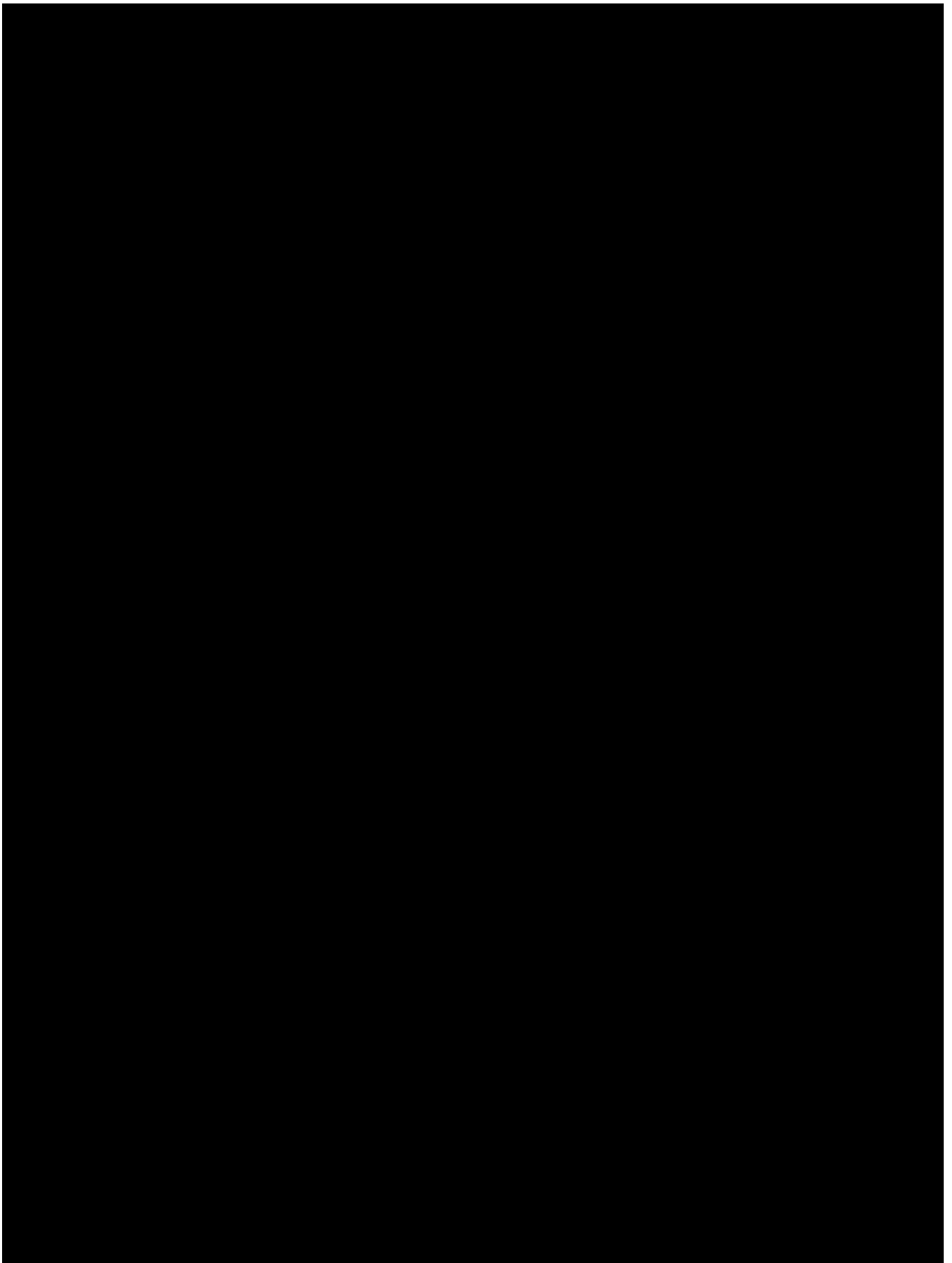


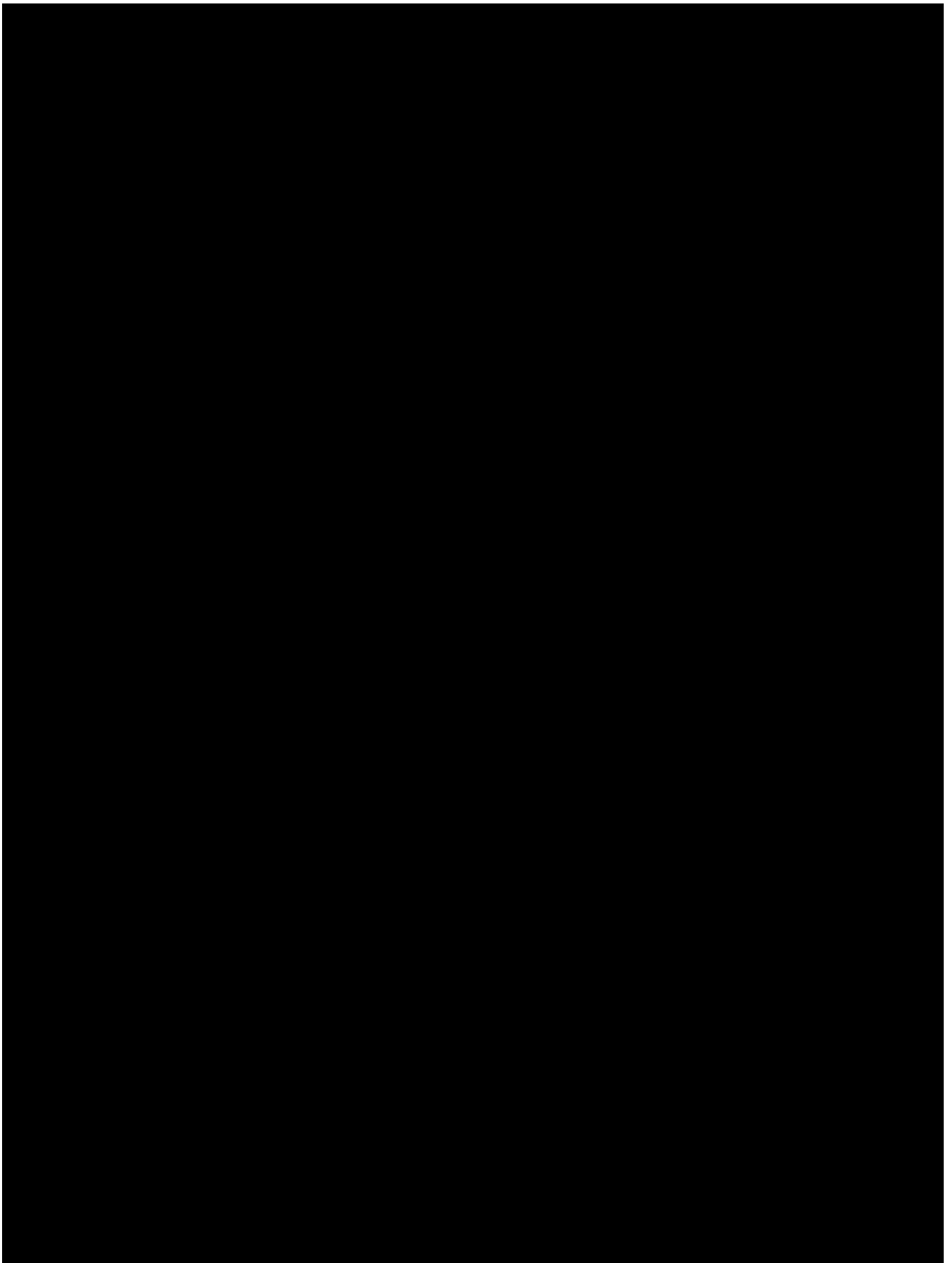
[illegible]

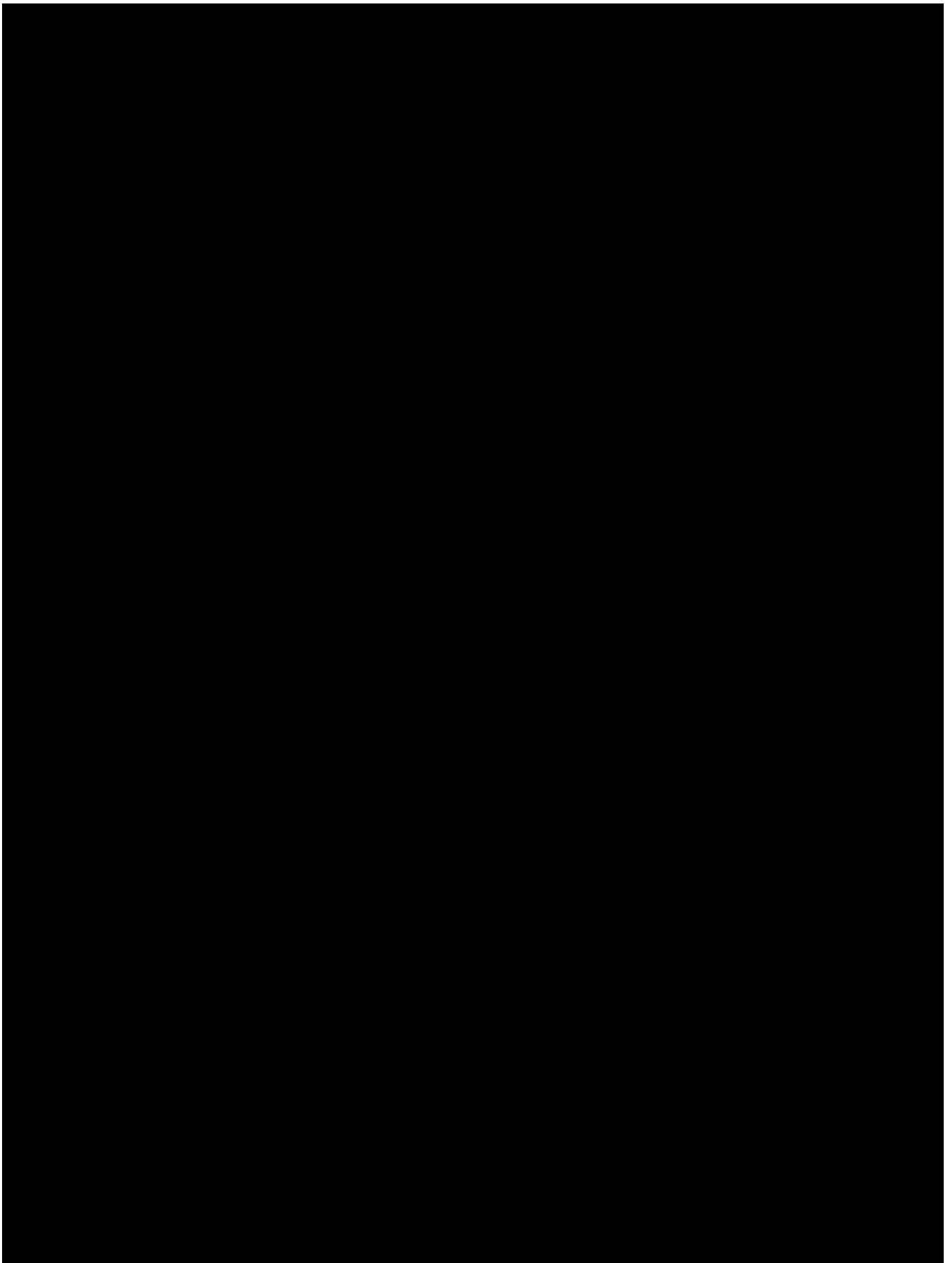


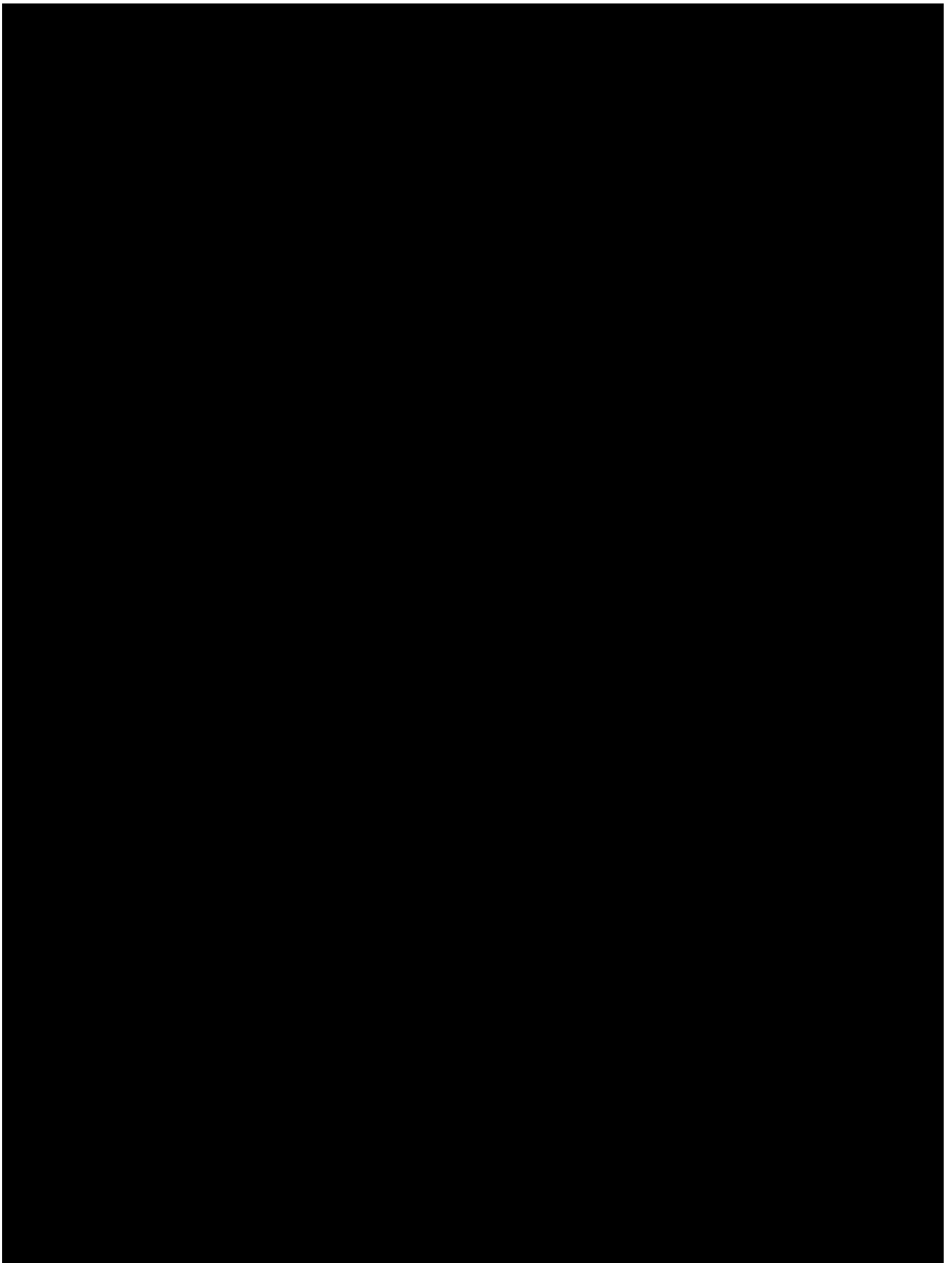


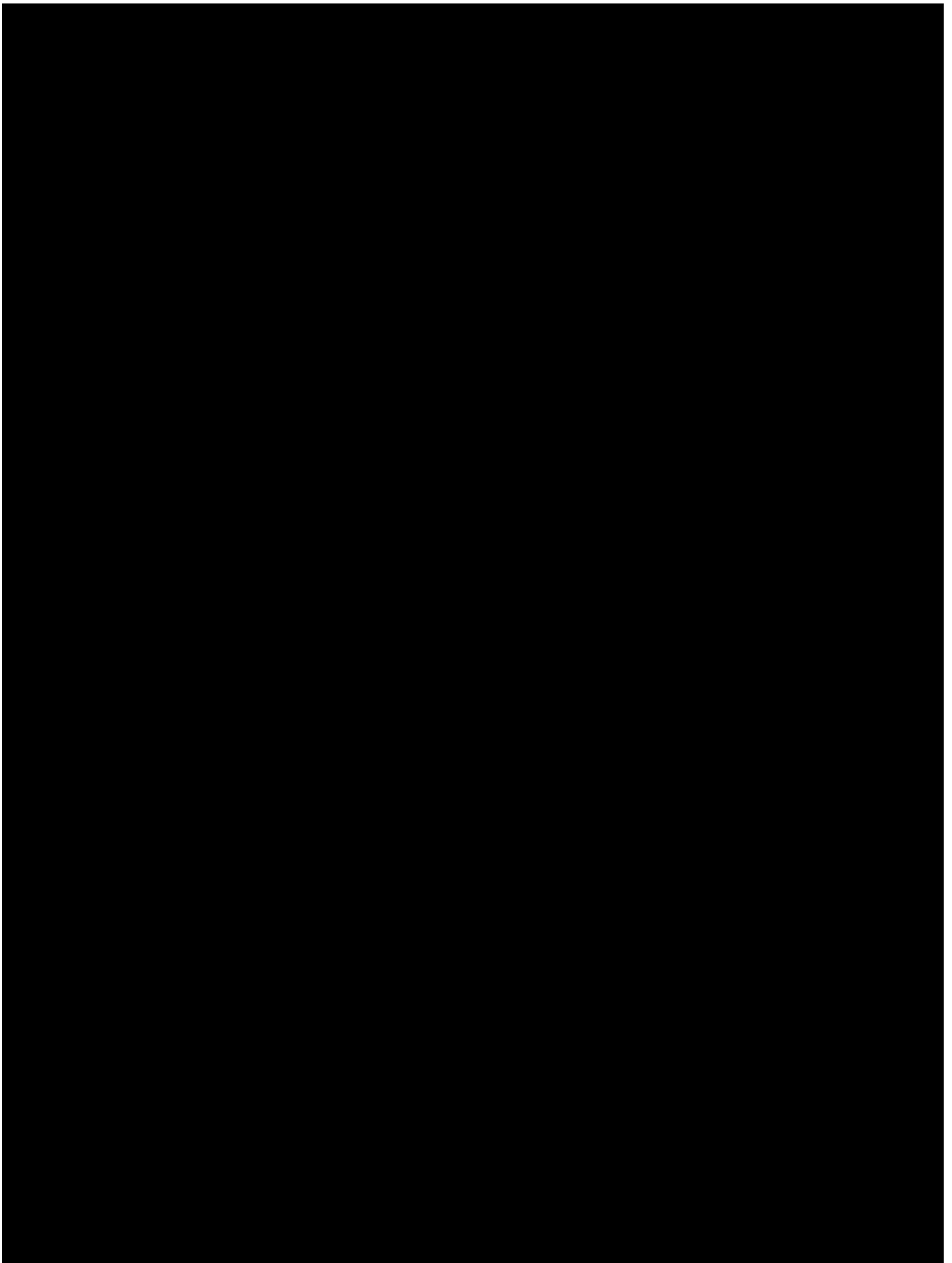


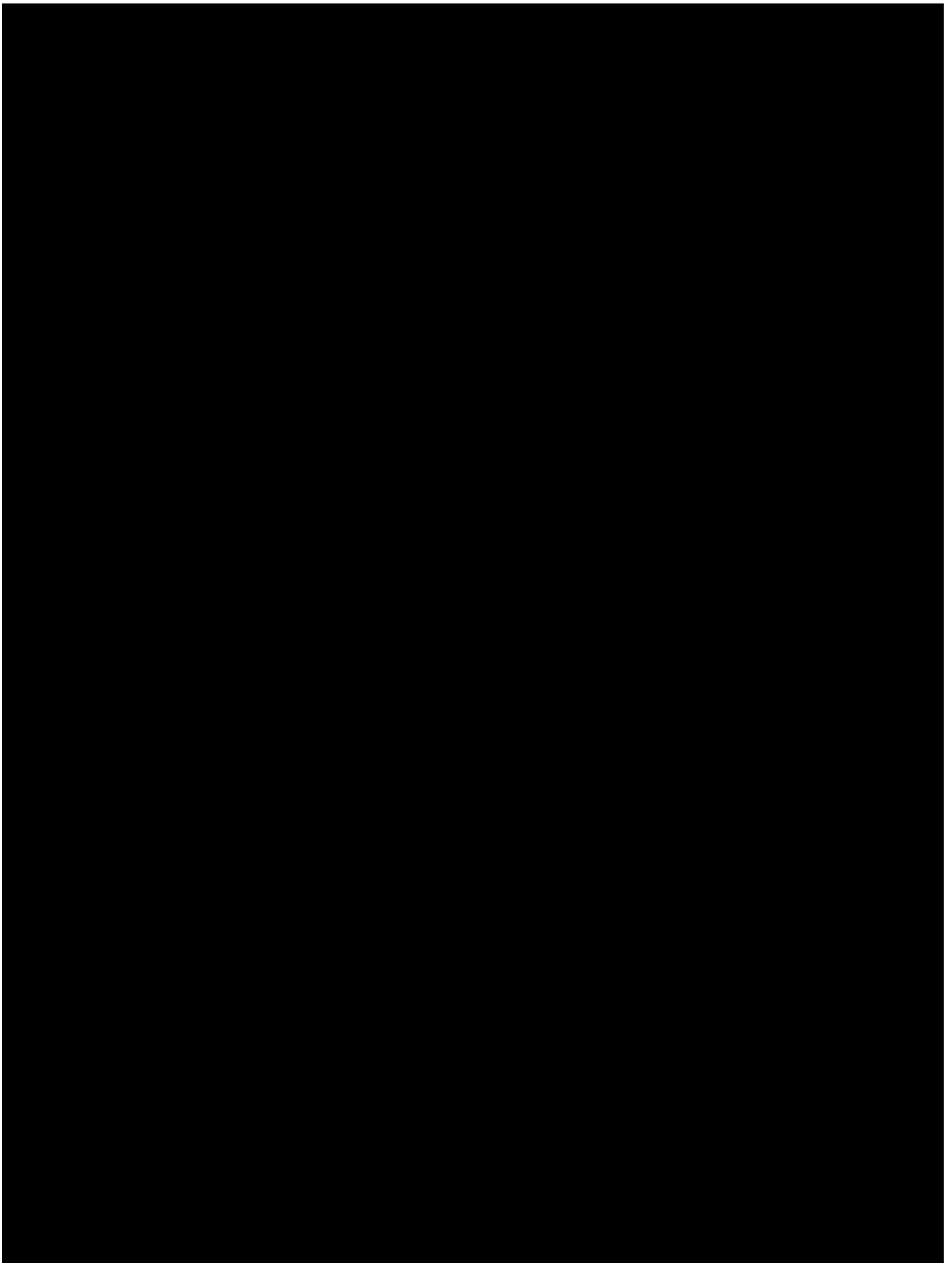




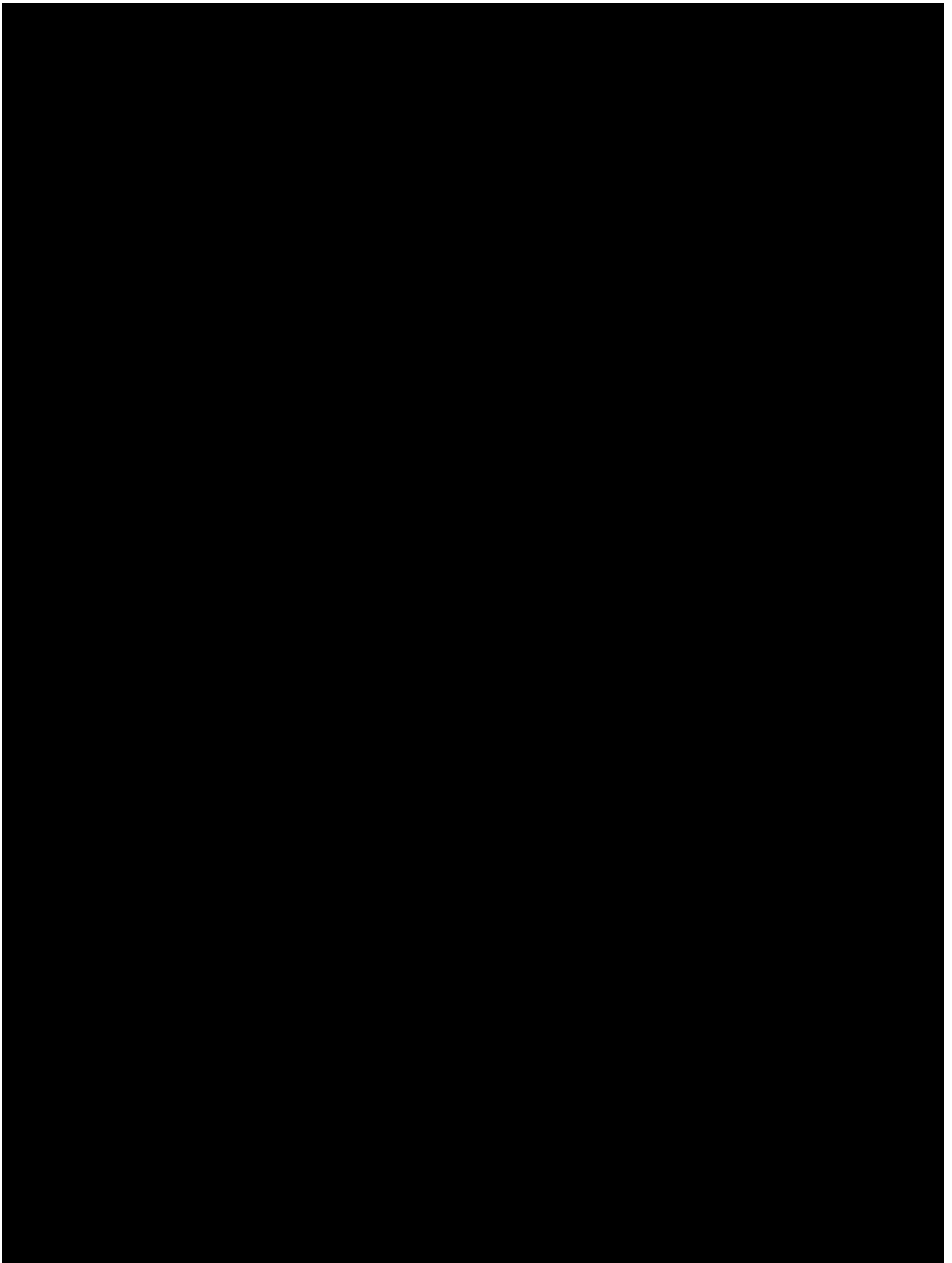


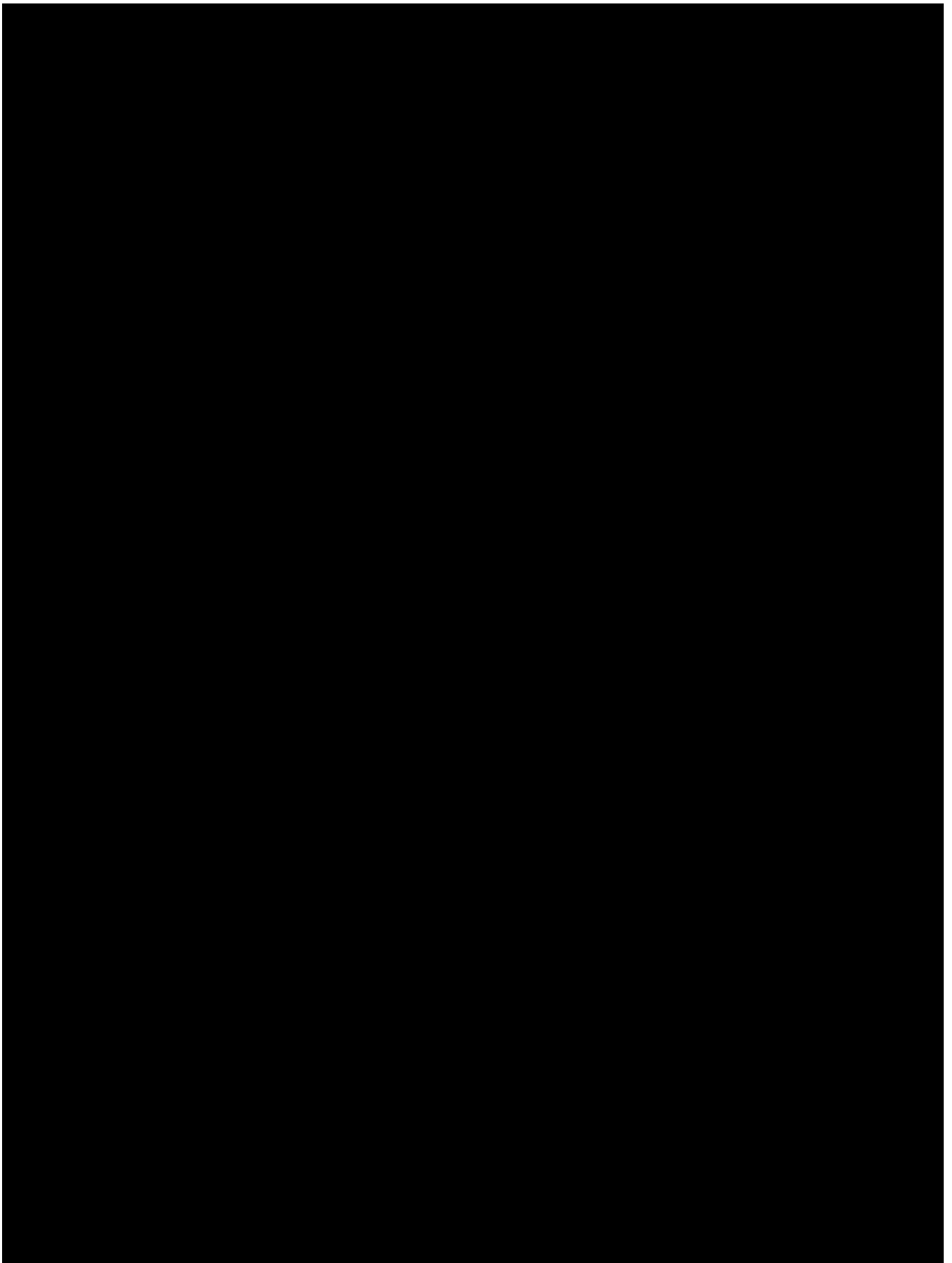


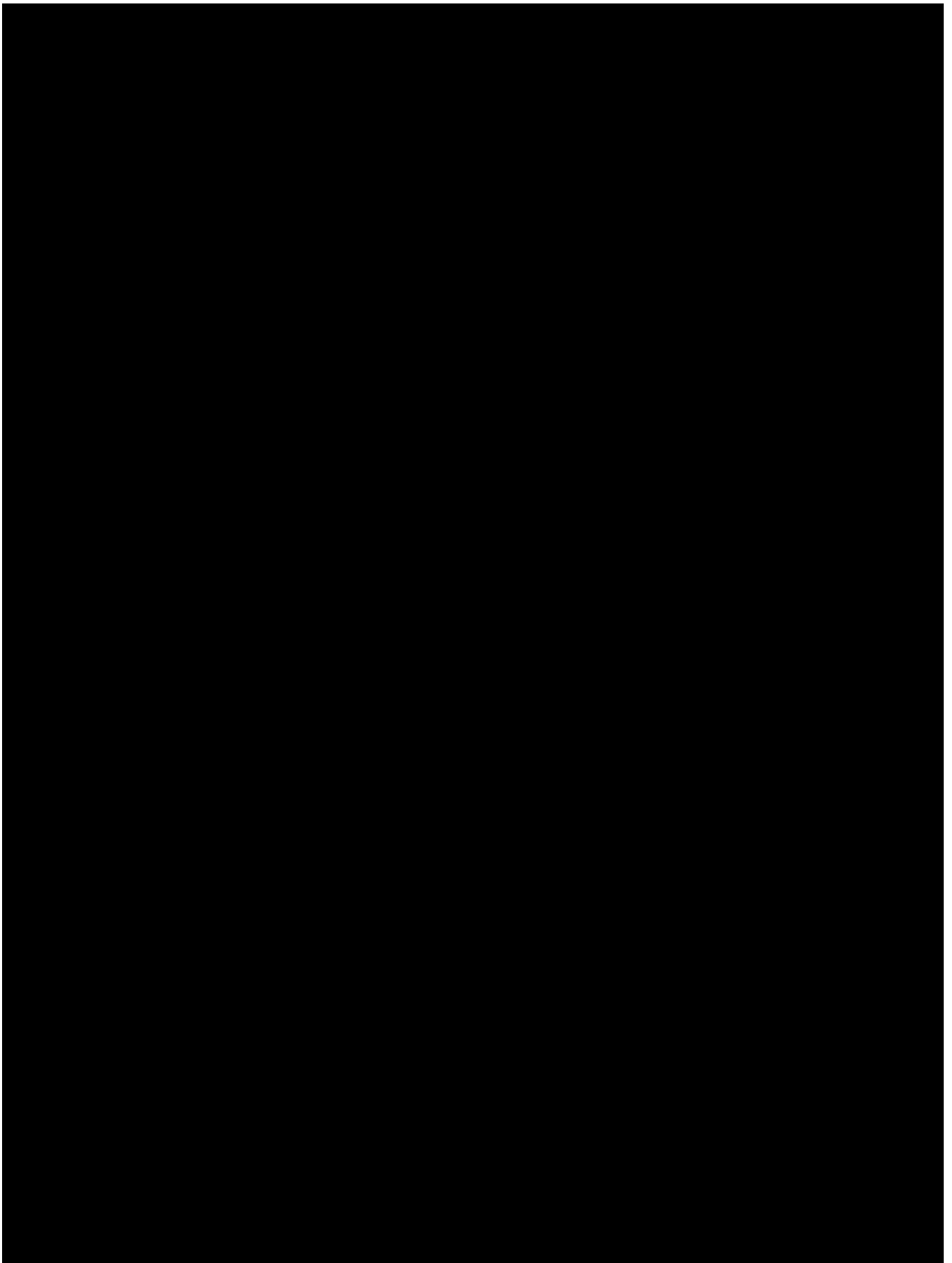


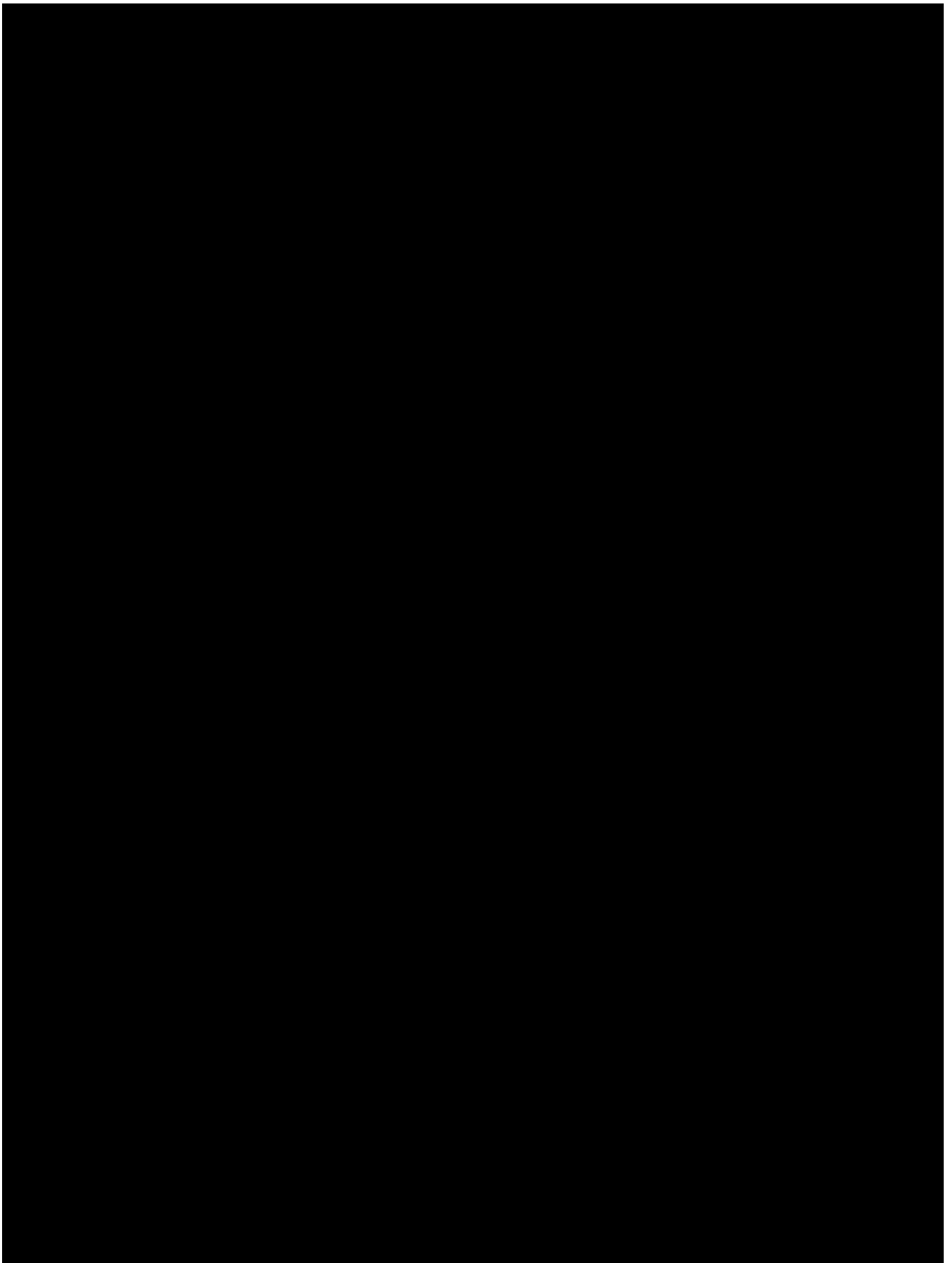


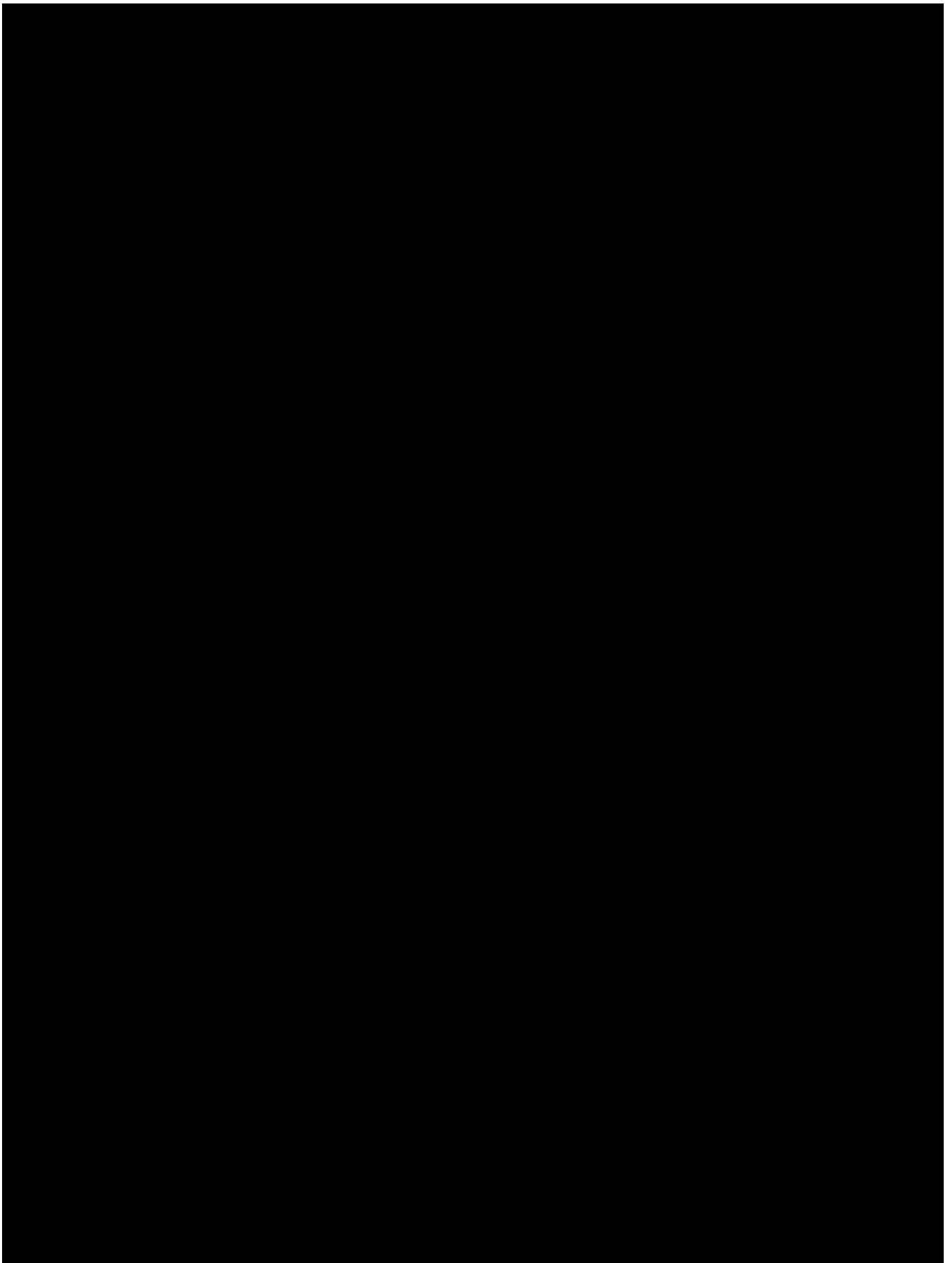


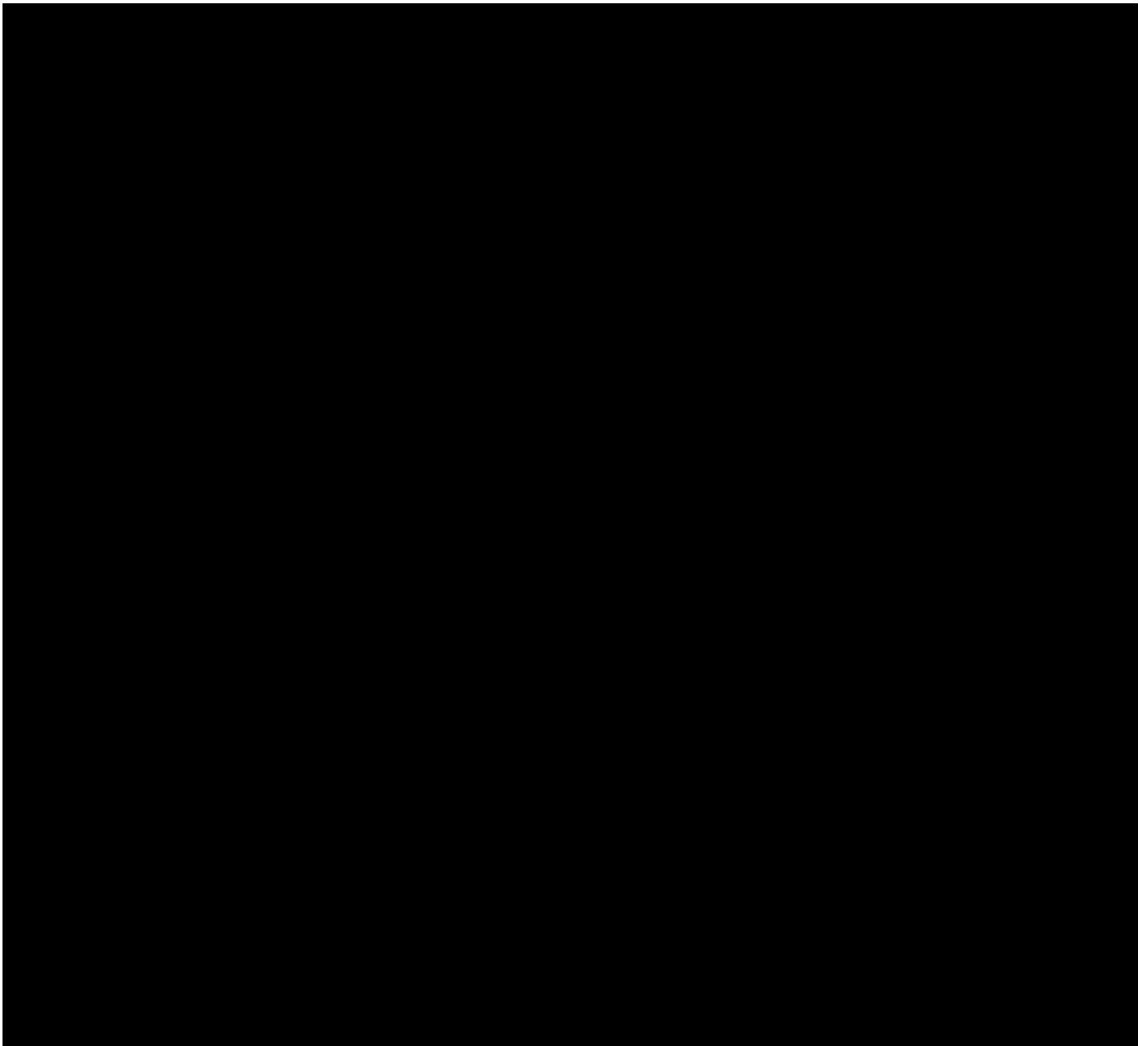












*This sheet contains references to all relevant workshop-related enabling tools as well as how to use them. Please read the table of contents and the TMAF section below if you are beginning modeling.*

[illegible][illegible]





[illegible]







[illegible]



