Student: Joey Wild – w039826m 2024/25

University of Staffordshire

Tool Development

Technical Design Document

Lecturer: TRANTER Jamie

w039826m@student.staffs.ac.uk

Contents

Project Introduction

This project is a collection of three tools, developed within a timespan of about 2-3 weeks each. The tools were created with other developers in mind as the intended users.

Tool 1 is a level editor, offering location, rotation, mesh/material, and name editing for various objects in the game world.

Tool 2 is a map generator, offering a way to create dynamic hex-tile-based maps using a set of adjustable parameters.

Tool 3 is a character creator with simple genetic functionality, specifically being a way to take 2 created characters and mix their features together to create a new character.

Project Goals

The goal of the project overall is to create 3 tools that are understandable to use, with navigable UI layout and appropriate features offered to achieve desired effects.

Tool 1 will aim to offer simpler/streamlined methods for laying out actors in the game world, as well as categorising objects or editing simple aspects of them (mesh and material) easily, and in groups.

Tool 2 will aim to offer a quick method to acquire aesthetically-pleasing, small maps with a hex tile style, also allowing access to each tile as an actor for further development into more complex uses (eg strategy games on a hex grid). Ample variety in maps generated is also a goal.

Tool 3 will aim to offer a simple method to simulate genetic mixing of characters, and a way to acquire a range of simplistically shaped characters (eg as stand-ins or placeholders). The genetic mixing functionality may also function as a basis to develop more complex 'family' systems.

Challenges and Risks

The main challenges arise with tools 2 and 3. Complex systems will be attempted in blueprint, and both will require experimentation and research to realise.

Tool 2 (map generator) requires the development of a robust system to lay out hex grid tiles, then apply 'biome'/feature spread across the tiles in believable and aesthetically consistent ways (eg spreading areas of forest/desert, dynamically generating mountains/hills/cliffs/water bodies). A challenge arising from this specifically is the heavy impact on performance in needing to frequently loop through tile actors to apply the correct effects on them, causing stuttering/pausing on generation, saving, and loading. This issue specifically worsened when transitioning from generating on BeginPlay to during editor, as the sphere collision used to gather neighbouring tiles of a tile (essential for generation methods) no longer functioned, and thus required looping through all tiles to find close neighbours.

Tool 3 (character creator / genetic simulator) requires an effective system for creating, passing, and adjusting data using structs. This is due to the created characters holding many features, many of which are involved in the genetic mix functionality. Created characters also must maintain persistence when being saved, loaded, reselected, and edited.

Hardware Requirements

Recommended:

- CPU: Intel® Core™ i7-11700

- GPU: NVIDIA GeForce RTX 3080
- Memory: 32GB
- Storage requirements: 734MB

Note that these are the specs of the PC these tools were developed on. They are thus recommended specs, and testing to find minimum specs has not been done.

Platforms

Target Platform

Windows PC

Engine Specific Specifications and Limitations

These tools are only available for use on Unreal Engine 5.

Note that the storage requirement listed above in the specs only accounts for the tools themselves, and not the overall Unreal project holding it.

Engine Summary

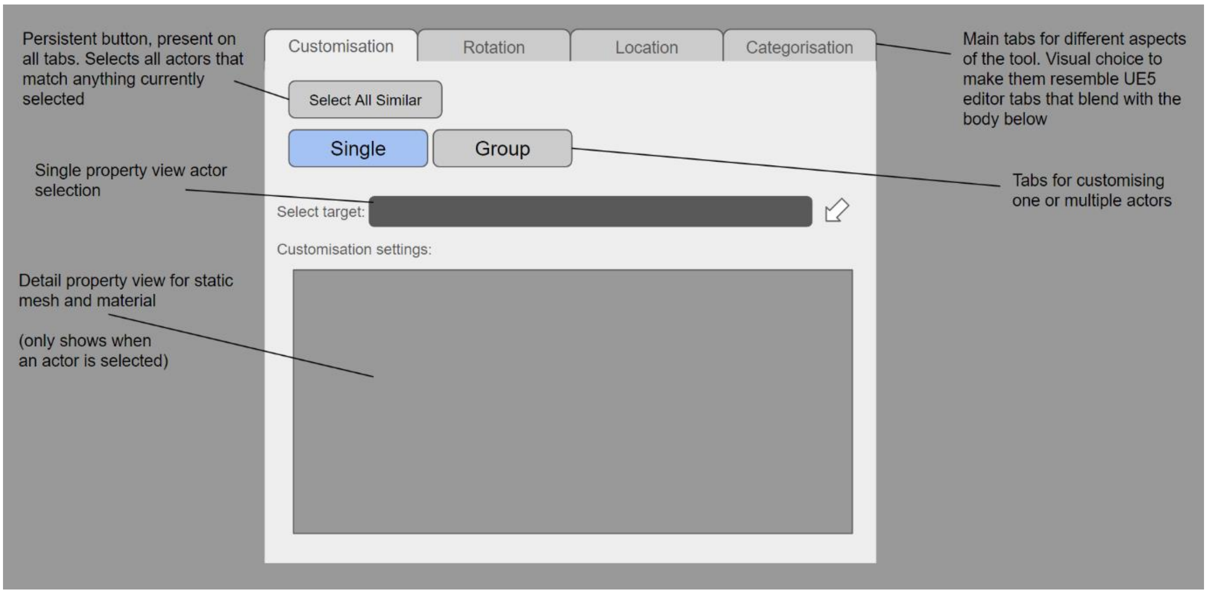
Version 5.4.4. No plugins are used.

Systems and Diagrams

System 1

Tool 1 – Level Editor

Wireframes



Persistent button, present on all tabs. Selects all actors that match anything currently selected

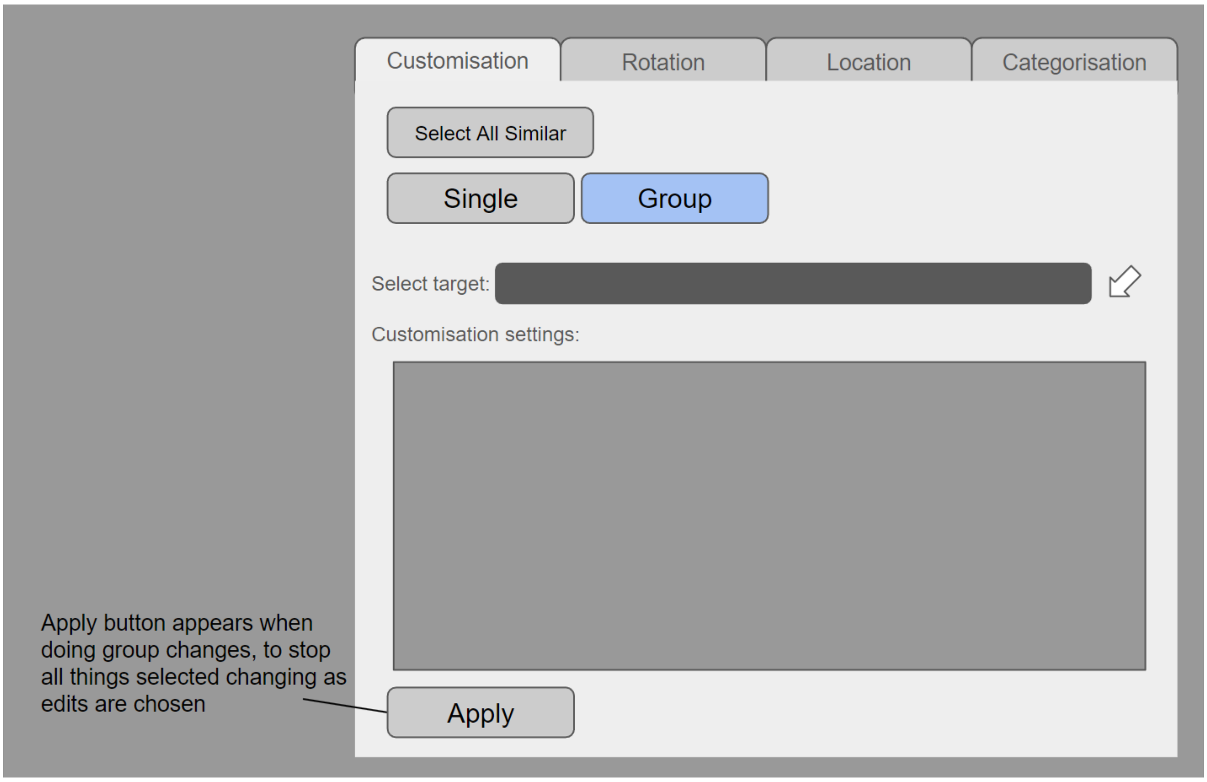
Single property view actor selection

Detail property view for static mesh and material

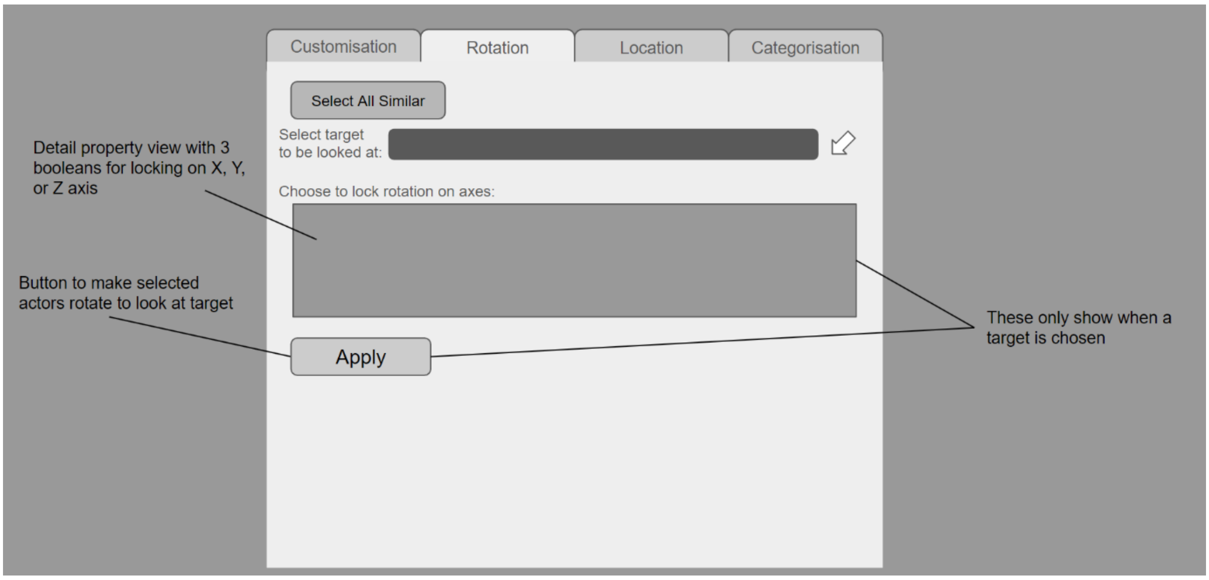
(only shows when an actor is selected)

Main tabs for different aspects of the tool. Visual choice to make them resemble UE5 editor tabs that blend with the body below

Tabs for customising one or multiple actors



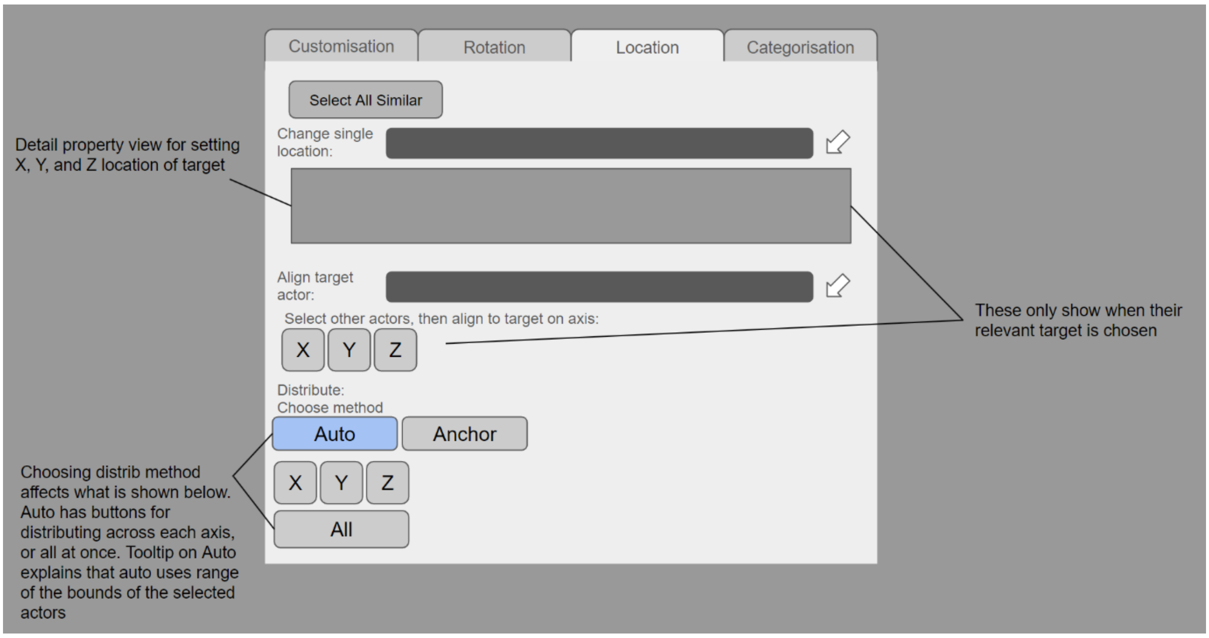
Apply button appears when doing group changes, to stop all things selected changing as edits are chosen



Detail property view with 3 booleans for locking on X, Y, or Z axis

Button to make selected actors rotate to look at target

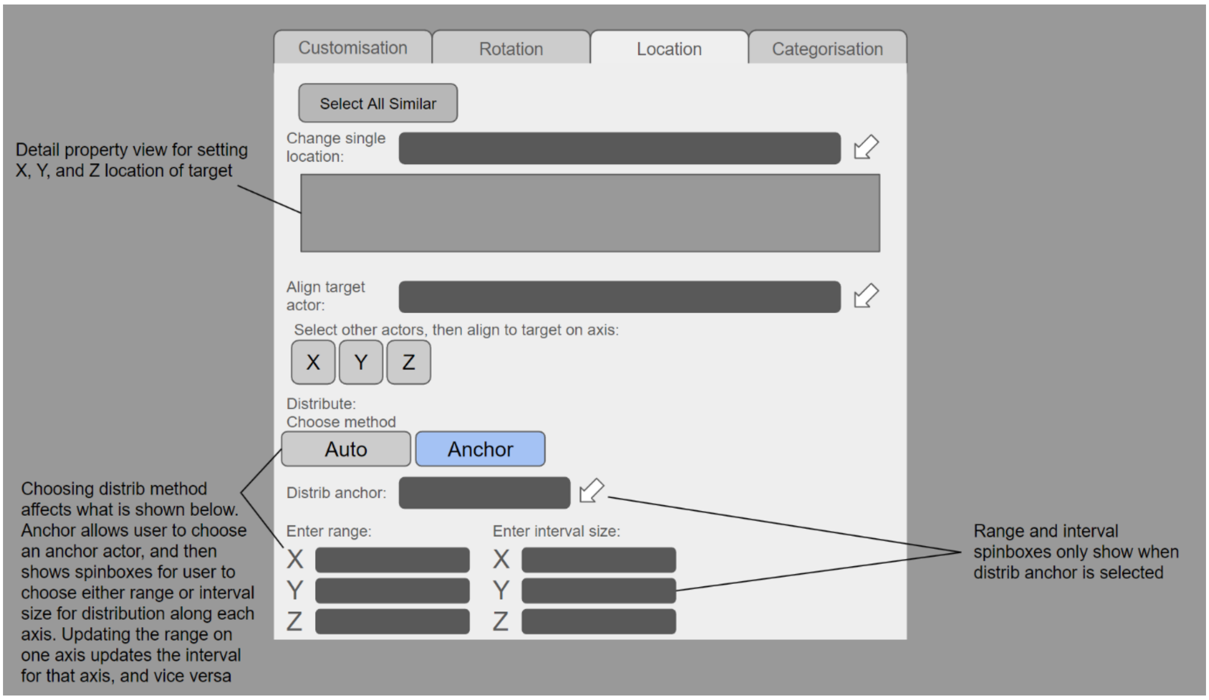
These only show when a target is chosen



Detail property view for setting X, Y, and Z location of target

These only show when their relevant target is chosen

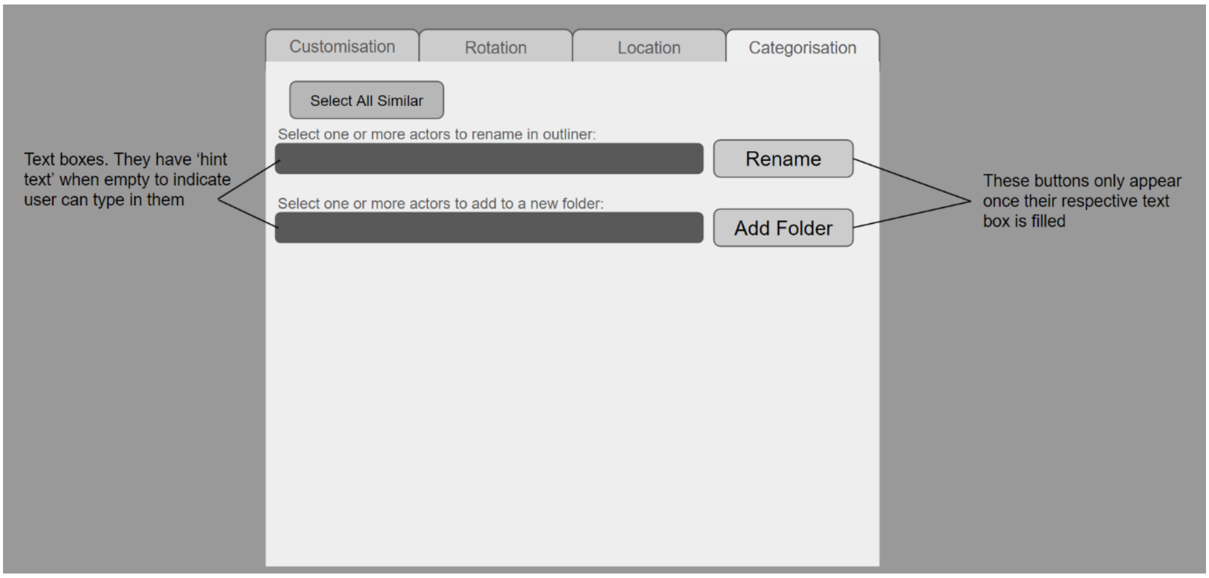
Choosing distrib method affects what is shown below. Auto has buttons for distributing across each axis, or all at once. Tooltip on Auto explains that auto uses range of the bounds of the selected actors



Detail property view for setting X, Y, and Z location of target

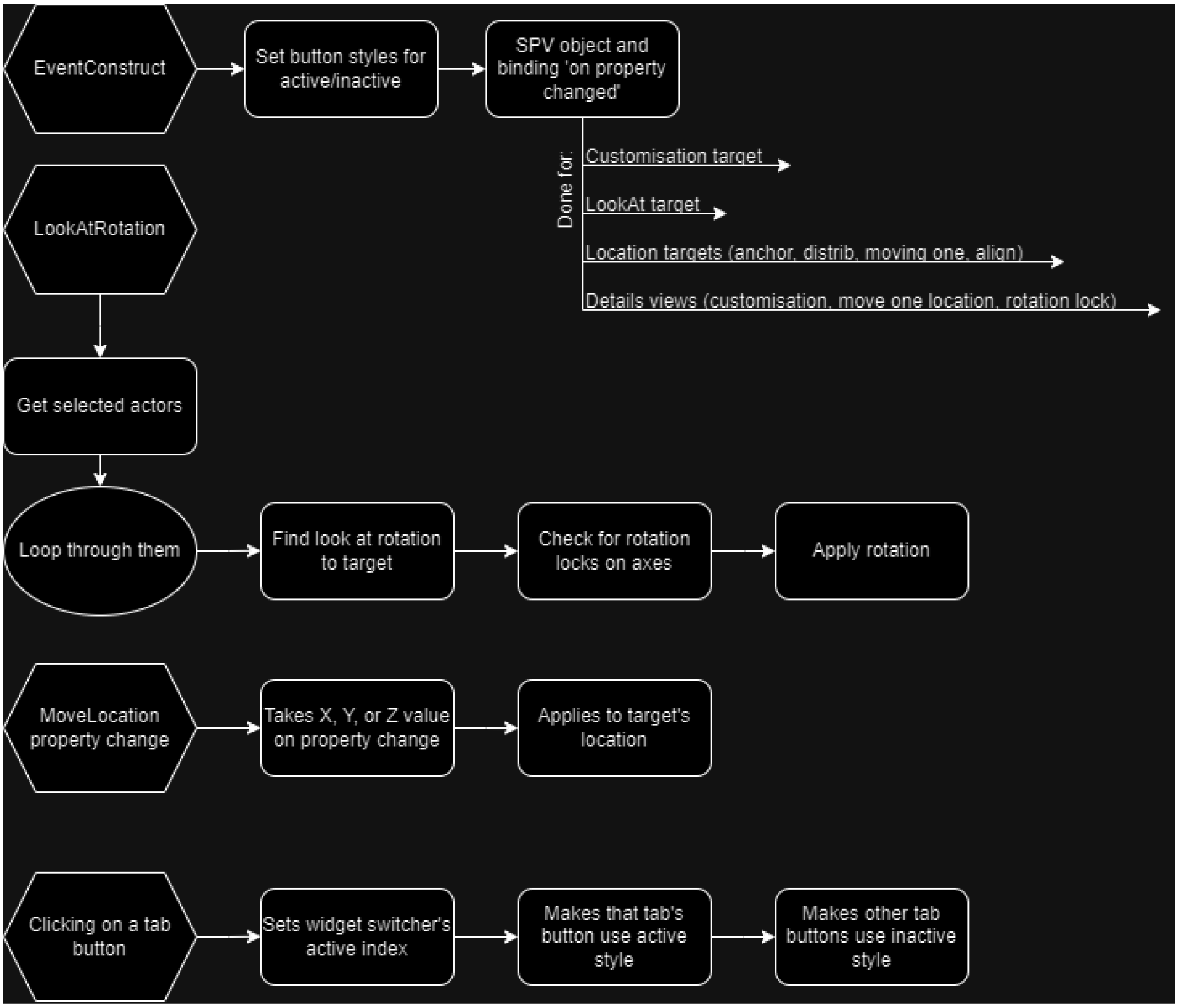
Choosing distrib method affects what is shown below. Anchor allows user to choose an anchor actor, and then shows spinboxes for user to choose either range or interval size for distribution along each axis. Updating the range on one axis updates the interval for that axis, and vice versa

Range and interval spinboxes only show when distrib anchor is selected

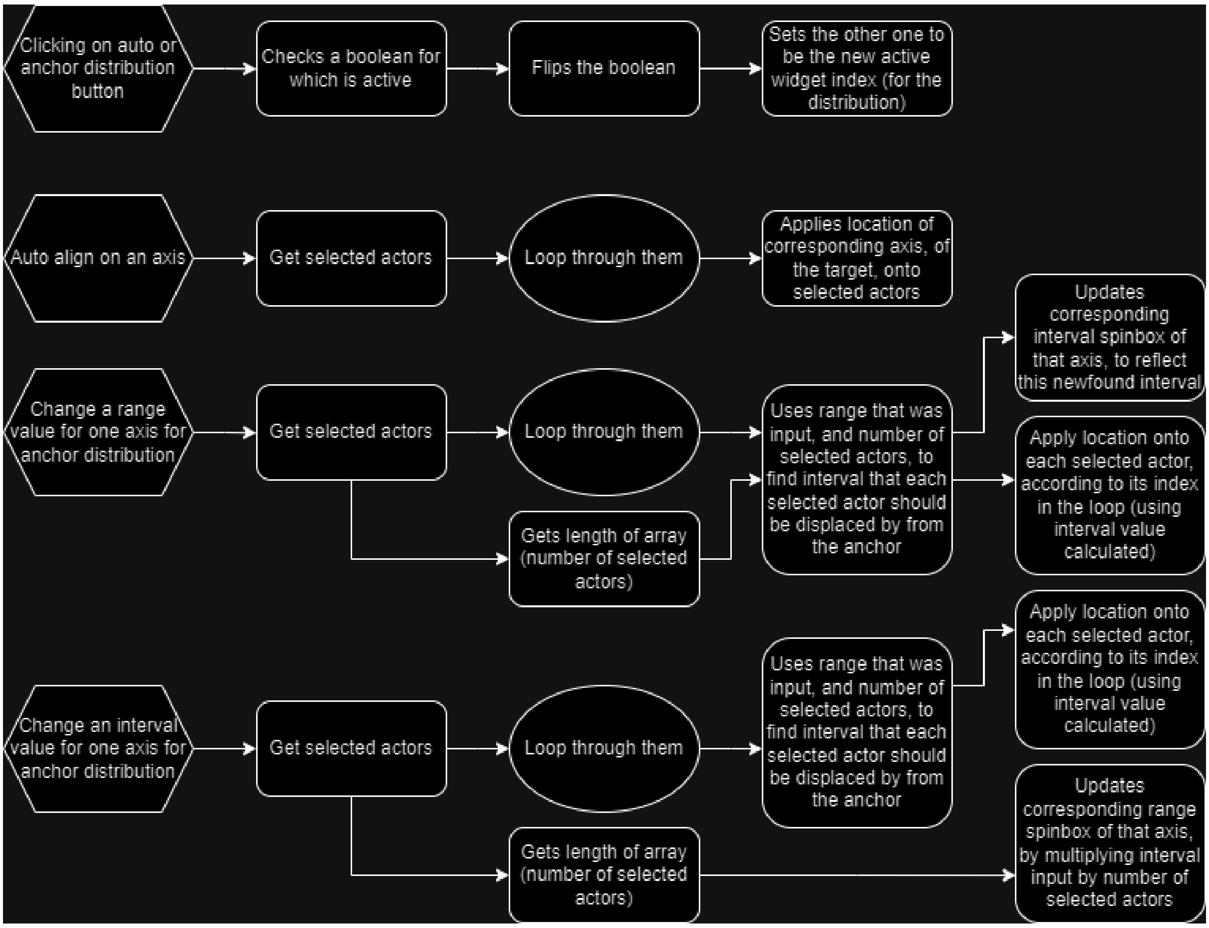

Customisation
Rotation
Location
Categorisation
Select All Similar
Select one or more actors to rename in outliner:
Rename
Select one or more actors to add to a new folder:
Add Folder
Text boxes. They have 'hint text' when empty to indicate user can type in them
These buttons only appear once their respective text box is filled

Blueprint flowcharts

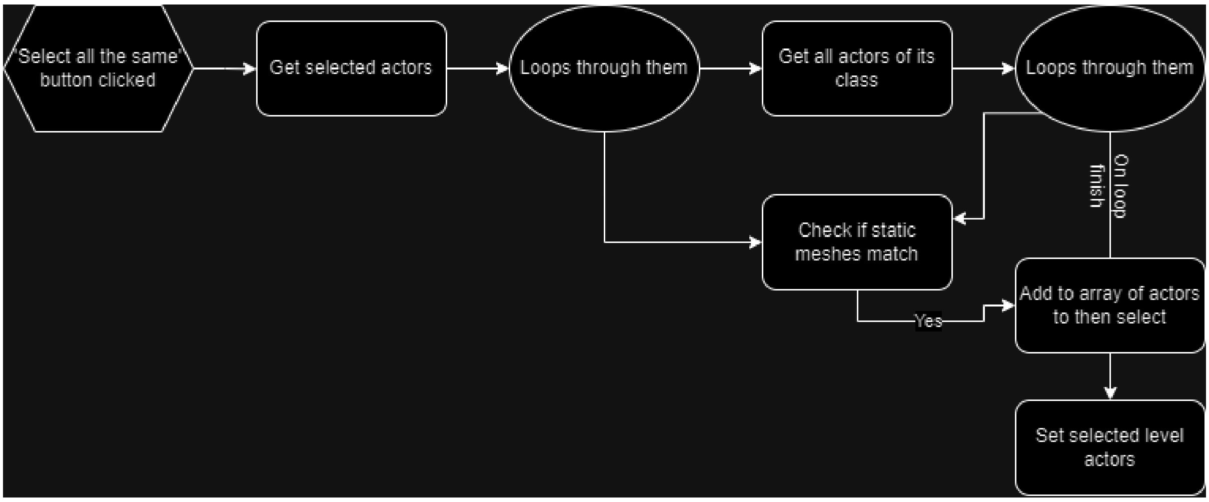
Base events/functions


EventConstruct
Set button styles for active/inactive
SPV object and binding 'on property changed'
Done for:
Customisation target
LookAt target
Location targets (anchor, distrib, moving one, align)
Details views (customisation, move one location, rotation lock)
LookAtRotation
Get selected actors
Loop through them
Find look at rotation to target
Check for rotation locks on axes
Apply rotation
MoveLocation property change
Takes X, Y, or Z value on property change
Applies to target's location
Clicking on a tab button
Sets widget switcher's active index
Makes that tab's button use active style
Makes other tab buttons use inactive style

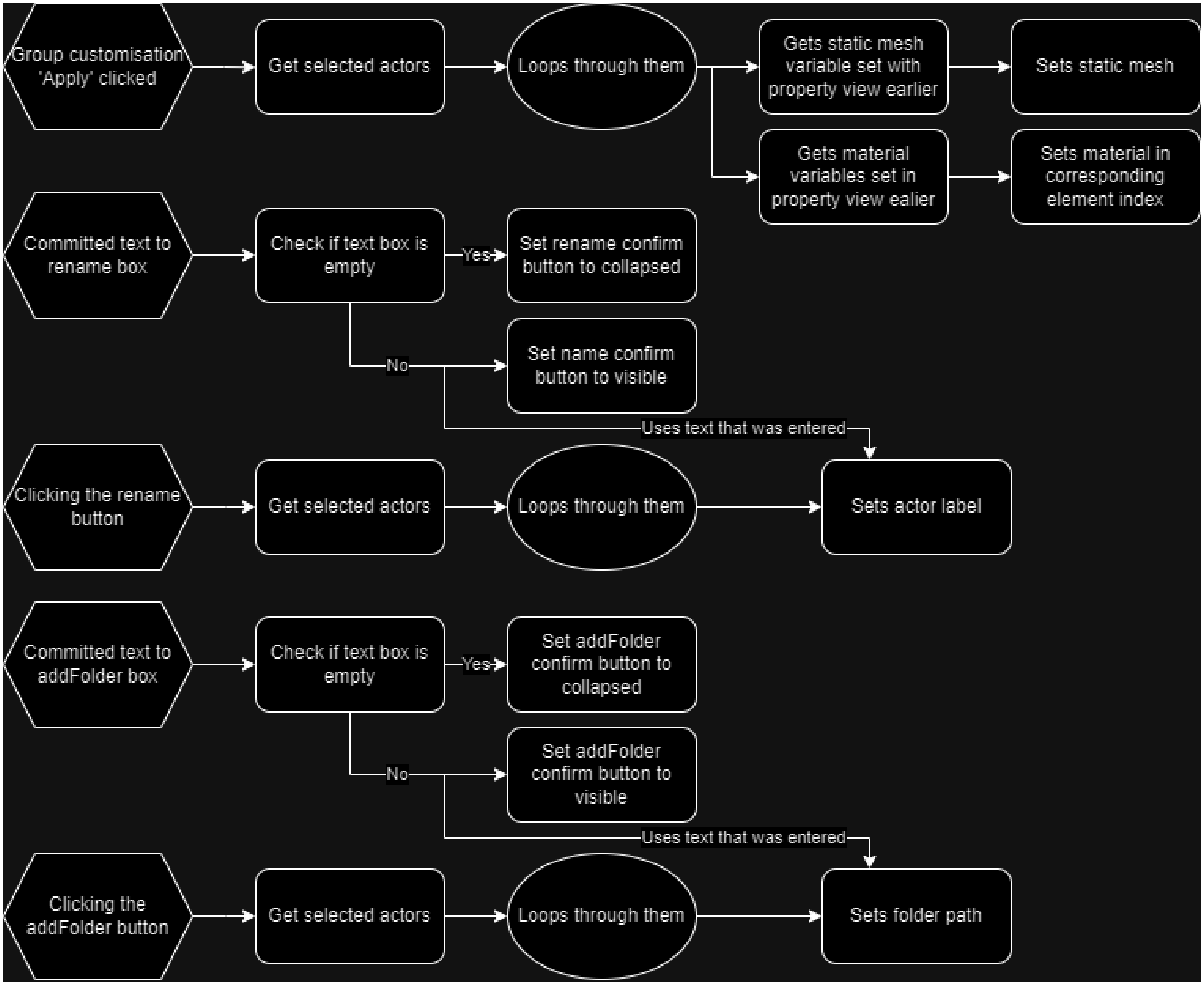
Location functions


Clicking on auto or anchor distribution button
Checks a boolean for which is active
Flips the boolean
Sets the other one to be the new active widget index (for the distribution)
Auto align on an axis
Get selected actors
Loop through them
Applies location of corresponding axis, of the target, onto selected actors
Change a range value for one axis for anchor distribution
Get selected actors
Loop through them
Gets length of array (number of selected actors)
Uses range that was input, and number of selected actors, to find interval that each selected actor should be displaced by from the anchor
Updates corresponding interval spinbox of that axis, to reflect this newfound interval
Apply location onto each selected actor, according to its index in the loop (using interval value calculated)
Change an interval value for one axis for anchor distribution
Get selected actors
Loop through them
Gets length of array (number of selected actors)
Uses range that was input, and number of selected actors, to find interval that each selected actor should be displaced by from the anchor
Apply location onto each selected actor, according to its index in the loop (using interval value calculated)
Updates corresponding range spinbox of that axis, by multiplying interval input by number of selected actors

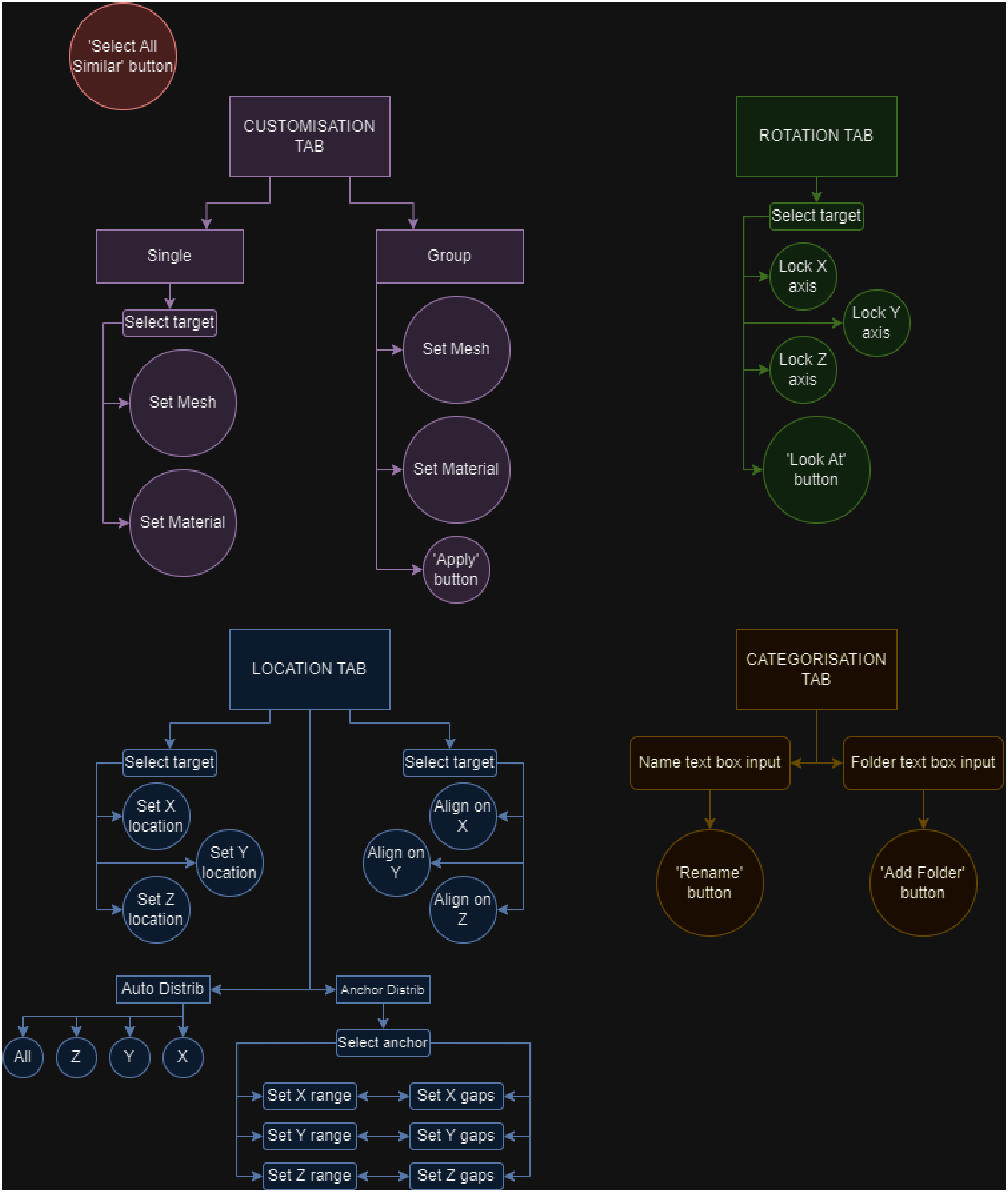
'Select all the same' function


'Select all the same' button clicked
Get selected actors
Loops through them
Get all actors of its class
Loops through them
Check if static meshes match
Yes
On loop finish
Add to array of actors to then select
Set selected level actors

Customisation and categorisation functions


Group customisation 'Apply' clicked
Get selected actors
Loops through them
Gets static mesh variable set with property view earlier
Sets static mesh
Gets material variables set in property view ealier
Sets material in corresponding element index
Committed text to rename box
Check if text box is empty
Yes
Set rename confirm button to collapsed
No
Set name confirm button to visible
Uses text that was entered
Clicking the rename button
Get selected actors
Loops through them
Sets actor label
Committed text to addFolder box
Check if text box is empty
Yes
Set addFolder confirm button to collapsed
No
Set addFolder confirm button to visible
Uses text that was entered
Clicking the addFolder button
Get selected actors
Loops through them
Sets folder path

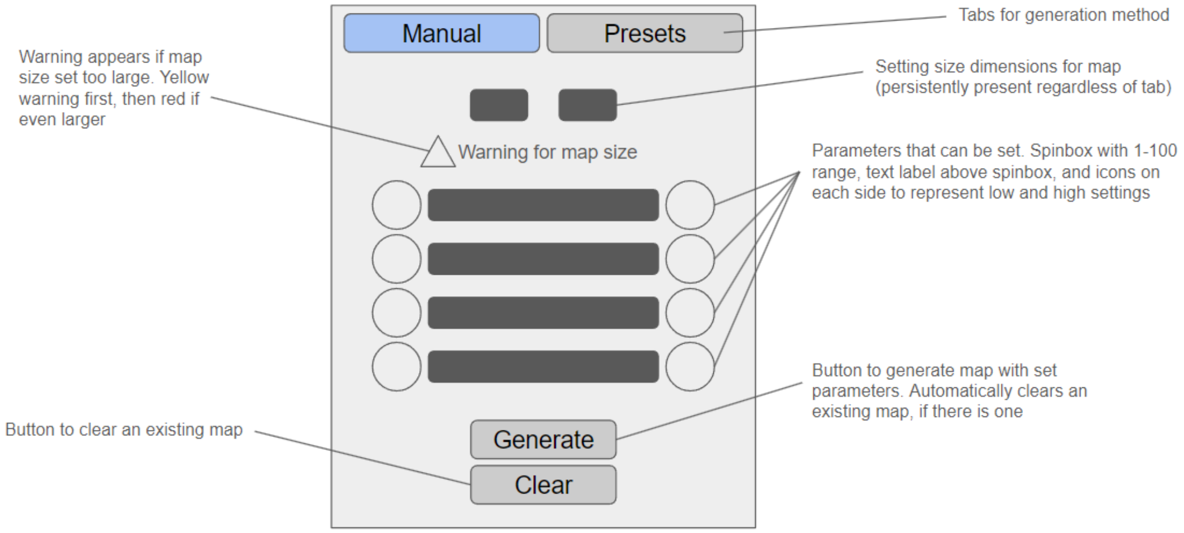
Information Architecture diagram



System 2

Tool 2 – Map Generator

Wireframes



Tabs for generation method

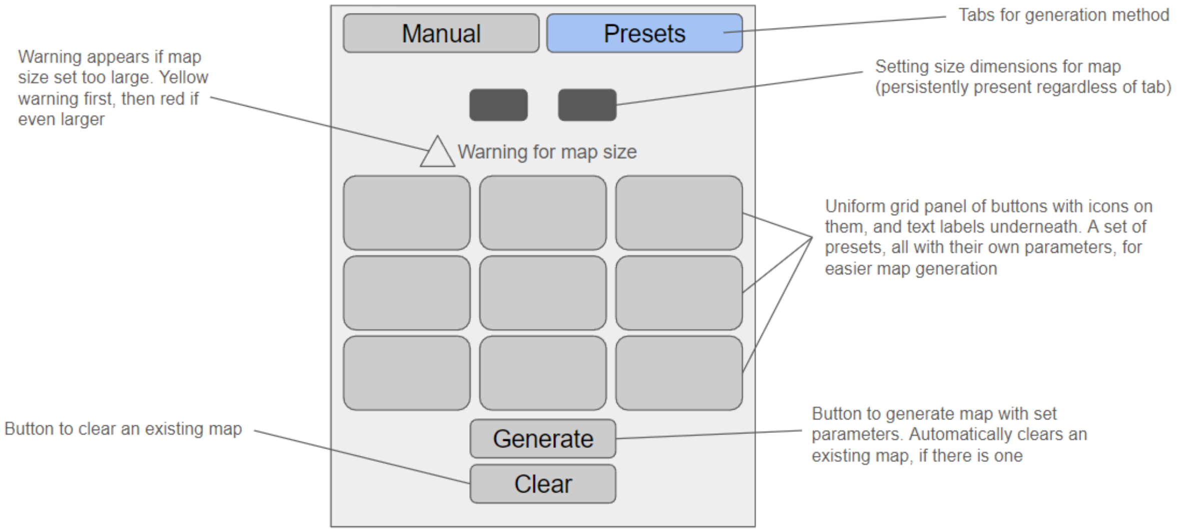
Setting size dimensions for map (persistently present regardless of tab)

Warning appears if map size set too large. Yellow warning first, then red if even larger

Parameters that can be set. Spinbox with 1-100 range, text label above spinbox, and icons on each side to represent low and high settings

Button to generate map with set parameters. Automatically clears an existing map, if there is one

Button to clear an existing map



Tabs for generation method

Setting size dimensions for map (persistently present regardless of tab)

Warning appears if map size set too large. Yellow warning first, then red if even larger

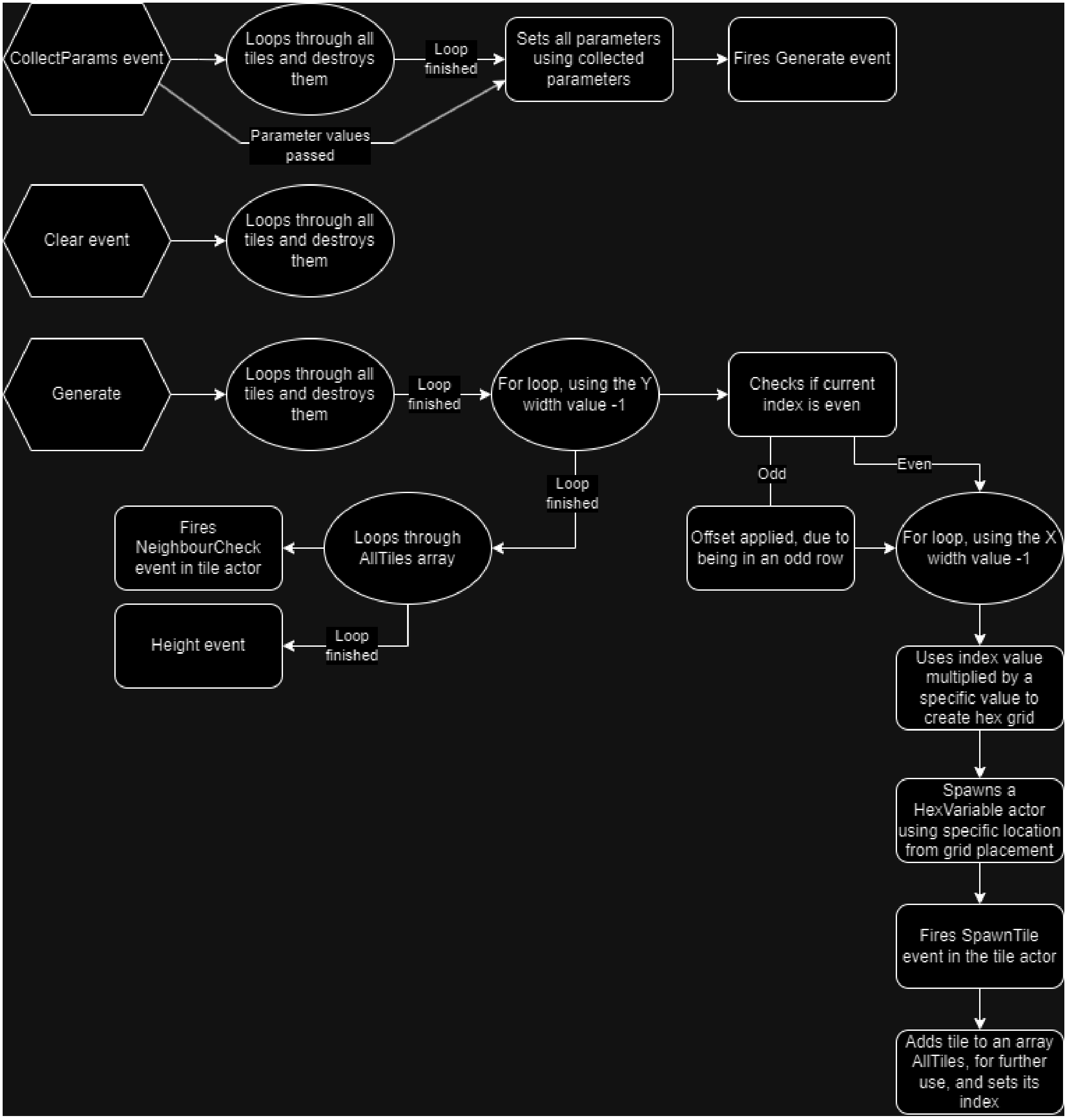
Uniform grid panel of buttons with icons on them, and text labels underneath. A set of presets, all with their own parameters, for easier map generation

Button to generate map with set parameters. Automatically clears an existing map, if there is one

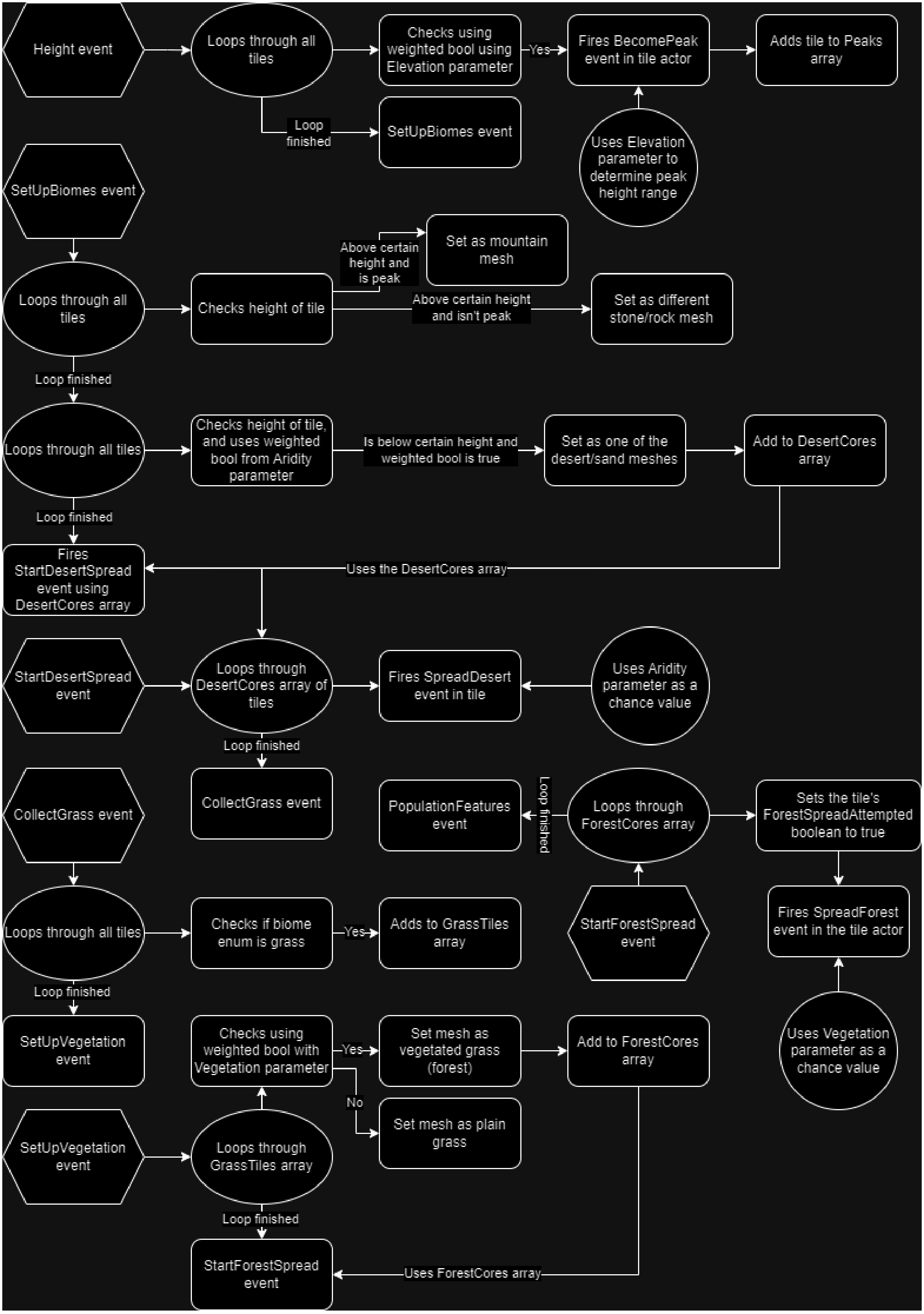
Button to clear an existing map

Blueprint flowcharts

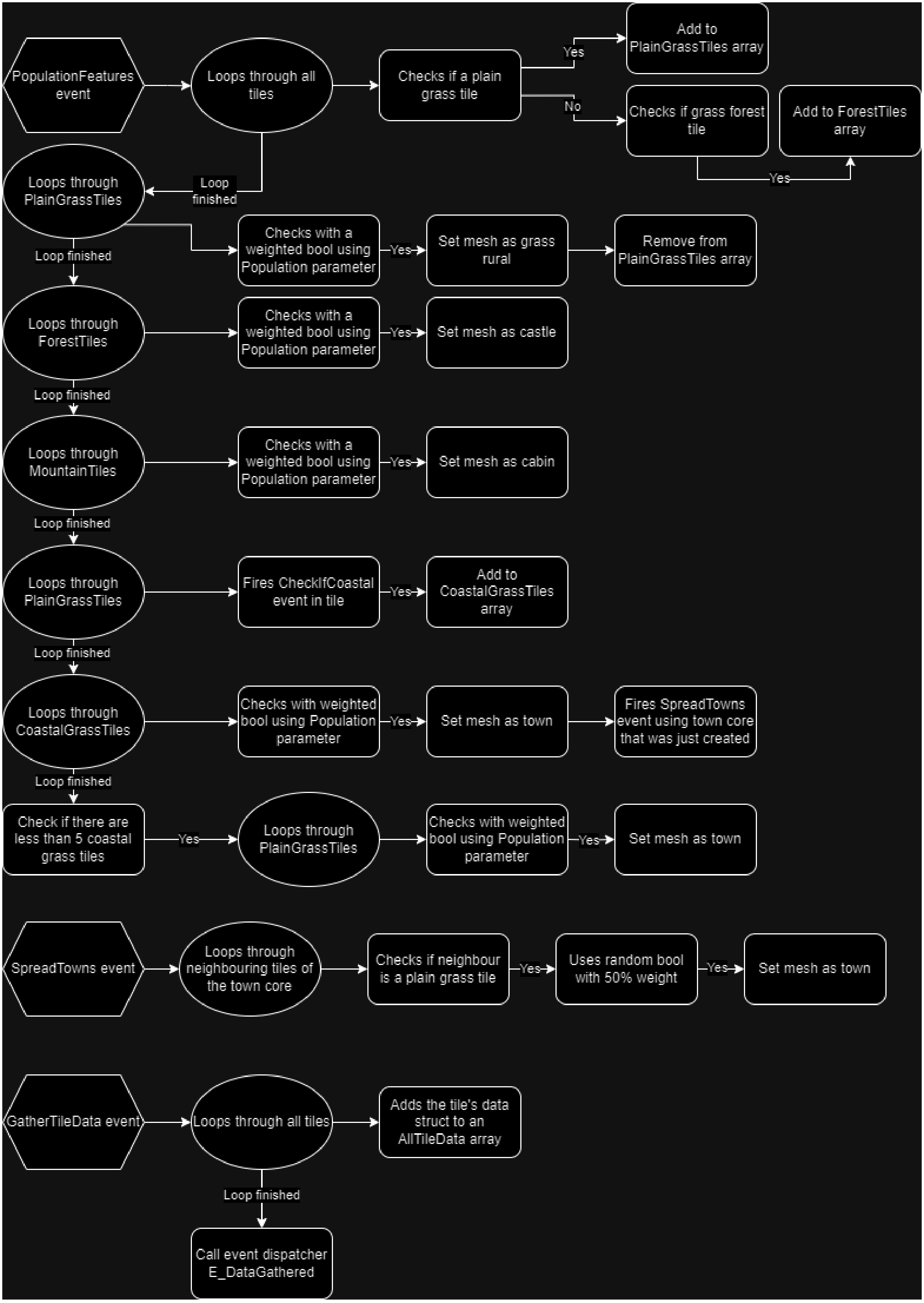
MapGen actor base functions


CollectParams event
Loops through all tiles and destroys them
Loop finished
Sets all parameters using collected parameters
Fires Generate event
Parameter values passed
Clear event
Loops through all tiles and destroys them
Generate
Loops through all tiles and destroys them
Loop finished
For loop, using the Y width value -1
Checks if current index is even
Odd
Even
Loop finished
Fires NeighbourCheck event in tile actor
Loops through AllTiles array
Offset applied, due to being in an odd row
For loop, using the X width value -1
Height event
Loop finished
Uses index value multiplied by a specific value to create hex grid
Spawns a HexVariable actor using specific location from grid placement
Fires SpawnTile event in the tile actor
Adds tile to an array AllTiles, for further use, and sets its index

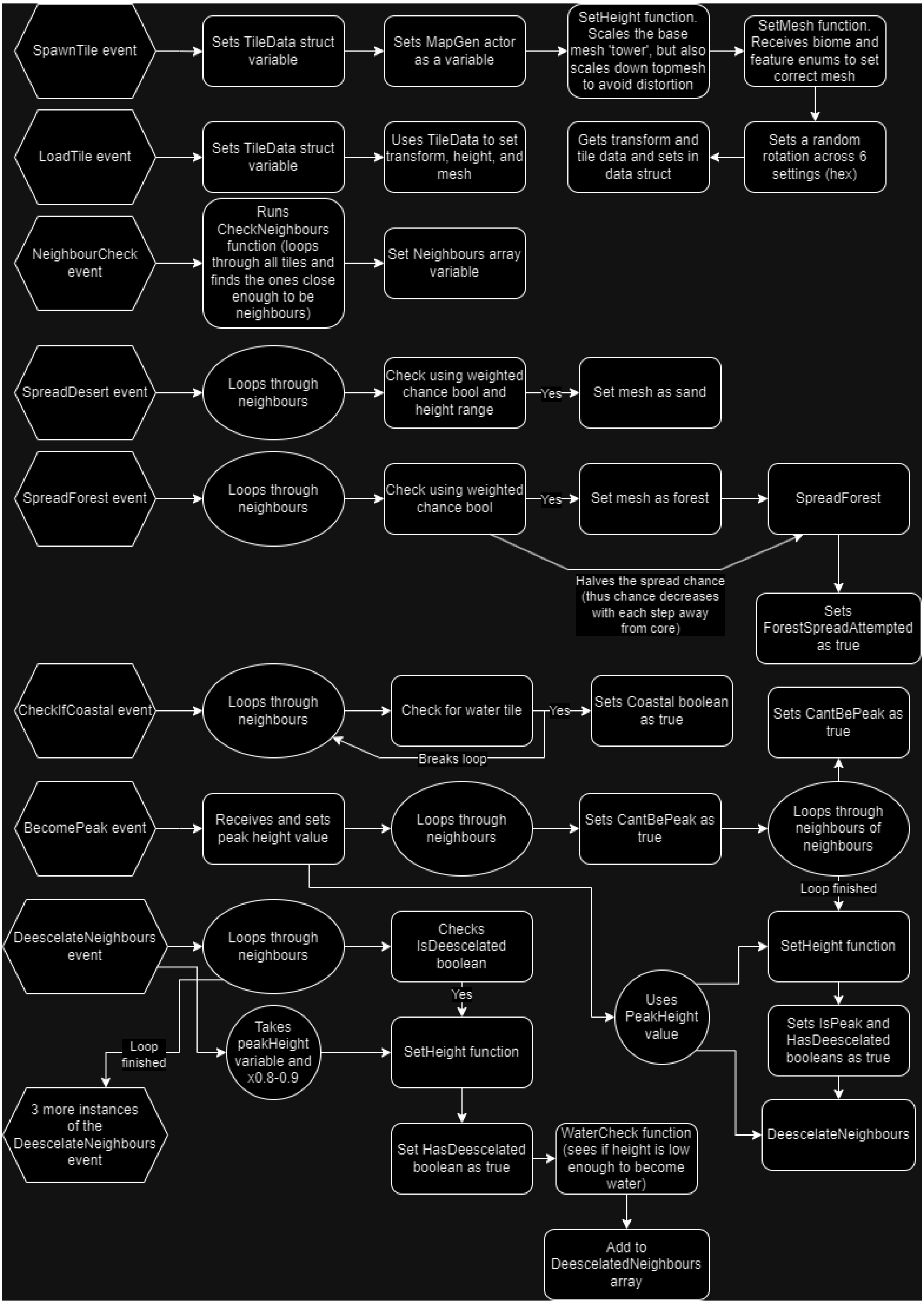
MapGen parameter functions


Height event
Loops through all tiles
Checks using weighted bool using Elevation parameter
Yes
Fires BecomePeak event in tile actor
Adds tile to Peaks array
Loop finished
SetUpBiomes event
Uses Elevation parameter to determine peak height range
SetUpBiomes event
Above certain height and is peak
Set as mountain mesh
Loops through all tiles
Checks height of tile
Above certain height and isn't peak
Set as different stone/rock mesh
Loop finished
Loops through all tiles
Checks height of tile, and uses weighted bool from Aridity parameter
Is below certain height and weighted bool is true
Set as one of the desert/sand meshes
Add to DesertCores array
Loop finished
Fires StartDesertSpread event using DesertCores array
Uses the DesertCores array
StartDesertSpread event
Loops through DesertCores array of tiles
Fires SpreadDesert event in tile
Uses Aridity parameter as a chance value
Loop finished
CollectGrass event
CollectGrass event
PopulationFeatures event
Loop finished
Loops through ForestCores array
Sets the tile's ForestSpreadAttempted boolean to true
Loops through all tiles
Checks if biome enum is grass
Yes
Adds to GrassTiles array
StartForestSpread event
Fires SpreadForest event in the tile actor
Loop finished
SetUpVegetation event
Checks using weighted bool with Vegetation parameter
Yes
Set mesh as vegetated grass (forest)
Add to ForestCores array
Uses Vegetation parameter as a chance value
No
Set mesh as plain grass
SetUpVegetation event
Loops through GrassTiles array
Loop finished
StartForestSpread event
Uses ForestCores array

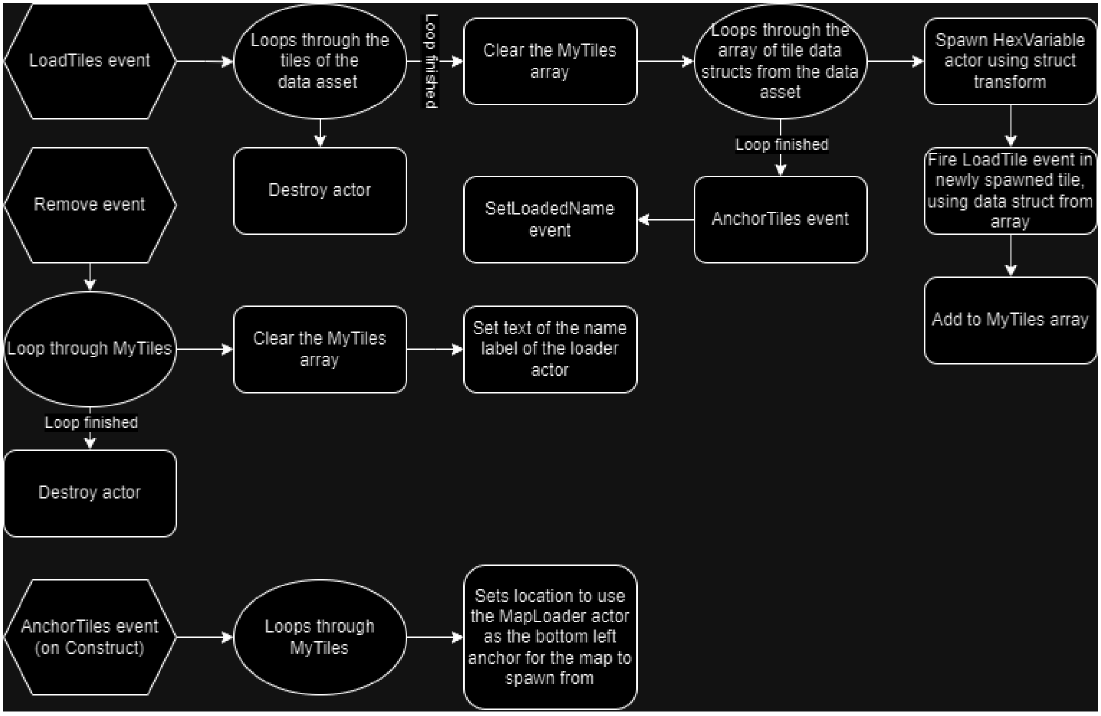
MapGen parameter functions (continued – population features)


PopulationFeatures event
Loops through all tiles
Checks if a plain grass tile
Yes
Add to PlainGrassTiles array
No
Checks if grass forest tile
Yes
Add to ForestTiles array
Loop finished
Loops through PlainGrassTiles
Checks with a weighted bool using Population parameter
Yes
Set mesh as grass rural
Remove from PlainGrassTiles array
Loop finished
Loops through ForestTiles
Checks with a weighted bool using Population parameter
Yes
Set mesh as castle
Loop finished
Loops through MountainTiles
Checks with a weighted bool using Population parameter
Yes
Set mesh as cabin
Loop finished
Loops through PlainGrassTiles
Fires CheckIfCoastal event in tile
Yes
Add to CoastalGrassTiles array
Loop finished
Loops through CoastalGrassTiles
Checks with weighted bool using Population parameter
Yes
Set mesh as town
Fires SpreadTowns event using town core that was just created
Loop finished
Check if there are less than 5 coastal grass tiles
Yes
Loops through PlainGrassTiles
Checks with weighted bool using Population parameter
Yes
Set mesh as town
SpreadTowns event
Loops through neighbouring tiles of the town core
Checks if neighbour is a plain grass tile
Yes
Uses random bool with 50% weight
Yes
Set mesh as town
GatherTileData event
Loops through all tiles
Adds the tile's data struct to an AllTileData array
Loop finished
Call event dispatcher E_DataGathered

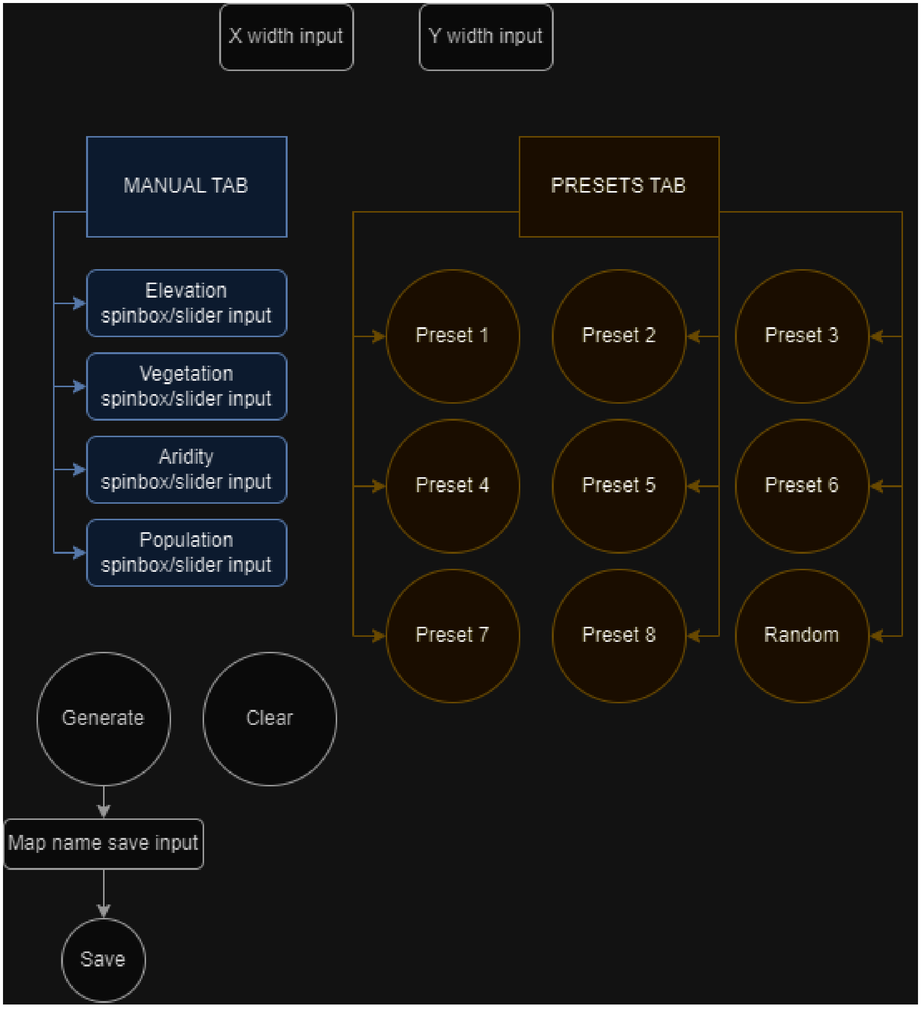
HexVariable actor


SpawnTile event
Sets TileData struct variable
Sets MapGen actor as a variable
SetHeight function. Scales the base mesh 'tower', but also scales down topmesh to avoid distortion
SetMesh function. Receives biome and feature enums to set correct mesh
LoadTile event
Sets TileData struct variable
Uses TileData to set transform, height, and mesh
Gets transform and tile data and sets in data struct
Sets a random rotation across 6 settings (hex)
NeighbourCheck event
Runs CheckNeighbours function (loops through all tiles and finds the ones close enough to be neighbours)
Set Neighbours array variable
SpreadDesert event
Loops through neighbours
Check using weighted chance bool and height range
Yes
Set mesh as sand
SpreadForest event
Loops through neighbours
Check using weighted chance bool
Yes
Set mesh as forest
SpreadForest
Halves the spread chance (thus chance decreases with each step away from core)
Sets ForestSpreadAttempted as true
CheckIfCoastal event
Loops through neighbours
Check for water tile
Yes
Sets Coastal boolean as true
Breaks loop
Sets CantBePeak as true
BecomePeak event
Receives and sets peak height value
Loops through neighbours
Sets CantBePeak as true
Loops through neighbours of neighbours
Loop finished
SetHeight function
Sets IsPeak and HasDeescelated booleans as true
DeescelateNeighbours
Uses PeakHeight value
DeescelateNeighbours event
Loops through neighbours
Checks IsDeescelated boolean
Yes
Loop finished
Takes peakHeight variable and x0.8-0.9
SetHeight function
3 more instances of the DeescelateNeighbours event
Set HasDeescelated boolean as true
WaterCheck function (sees if height is low enough to become water)
Add to DeescelatedNeighbours array

MapLoader actor


LoadTiles event
Loops through the tiles of the data asset
Loop finished
Clear the MyTiles array
Loops through the array of tile data structs from the data asset
Spawn HexVariable actor using struct transform
Remove event
Destroy actor
Loop finished
SetLoadedName event
AnchorTiles event
Fire LoadTile event in newly spawned tile, using data struct from array
Loop through MyTiles
Clear the MyTiles array
Set text of the name label of the loader actor
Add to MyTiles array
Loop finished
Destroy actor
AnchorTiles event (on Construct)
Loops through MyTiles
Sets location to use the MapLoader actor as the bottom left anchor for the map to spawn from

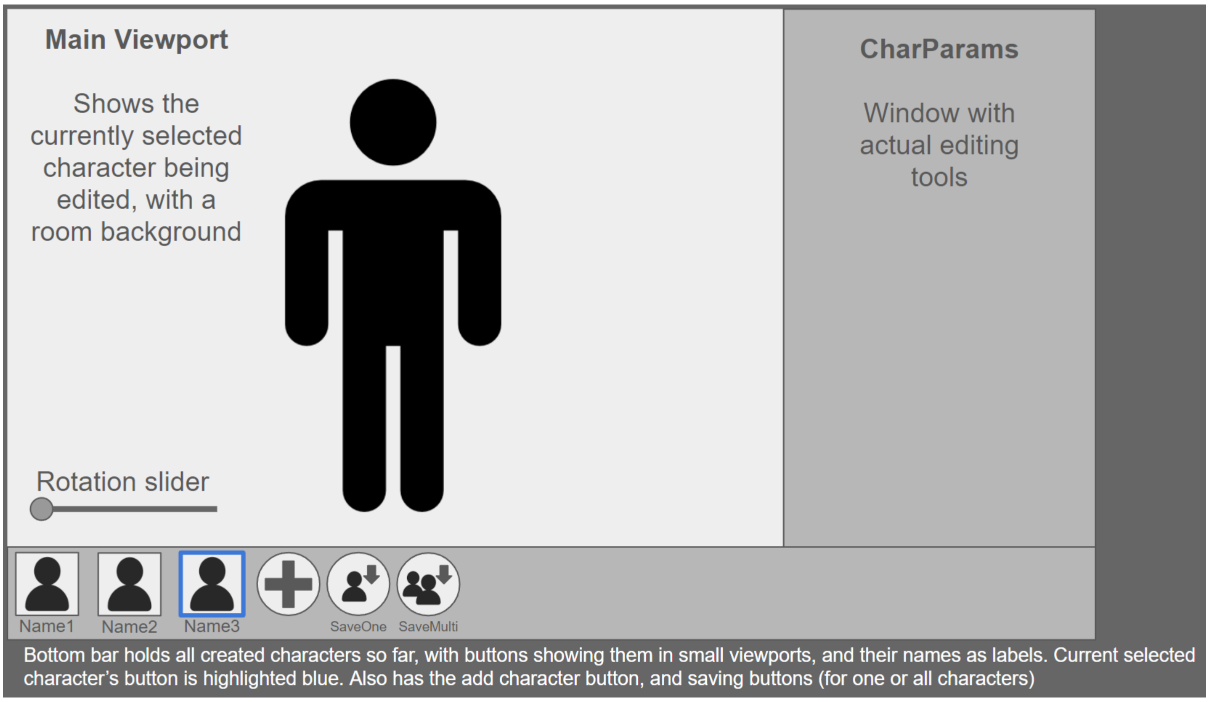
Information Architecture diagram



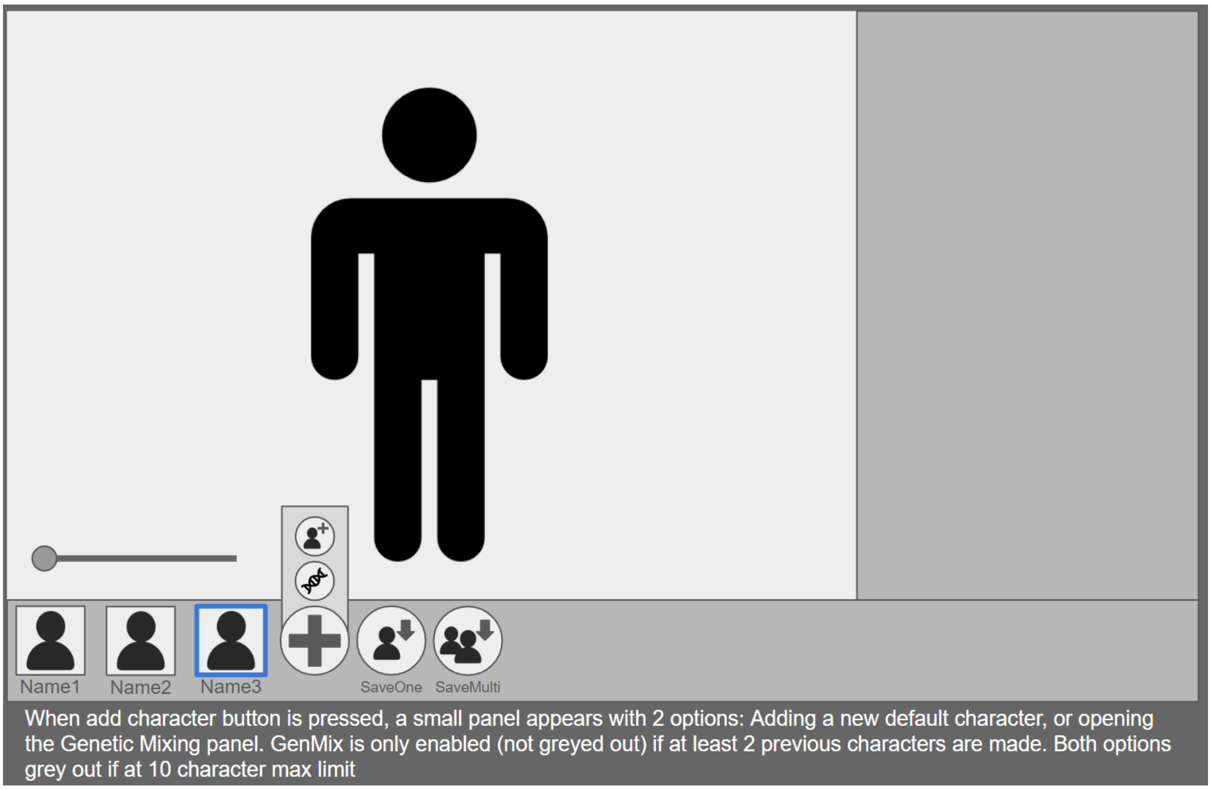
System 3

Tool 3 – Character Creator / Genetics Simulator

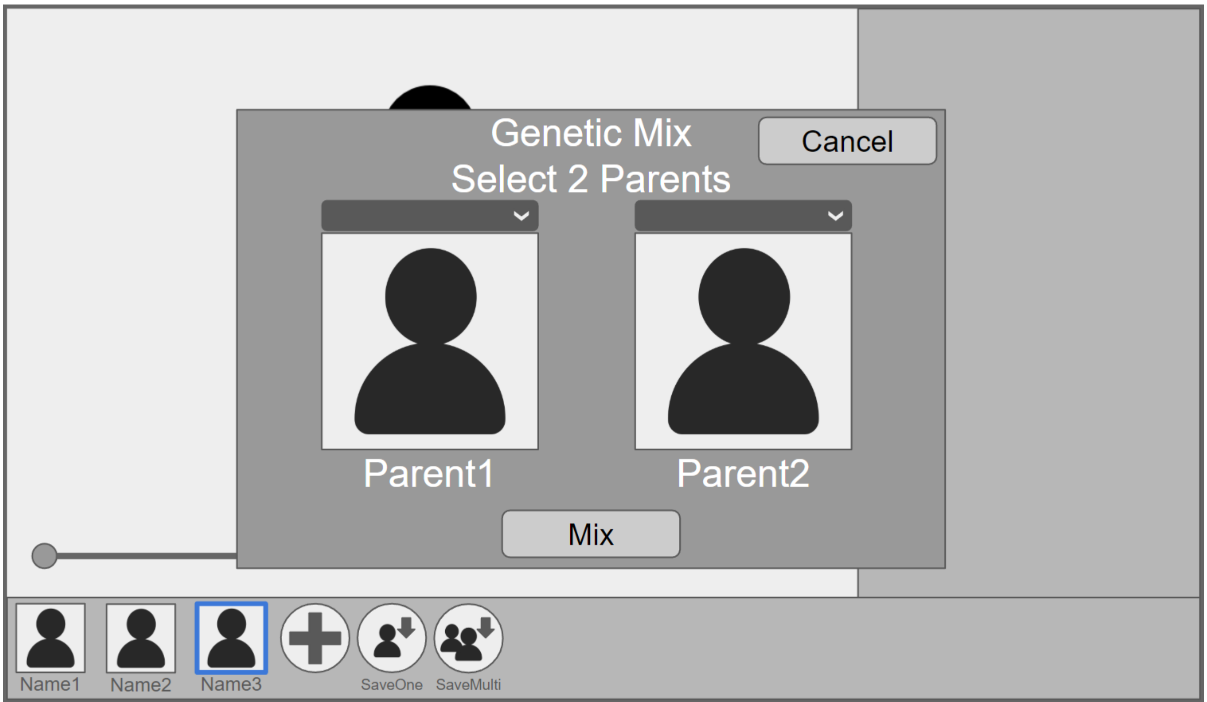
Wireframes


Main Viewport
Shows the currently selected character being edited, with a room background
CharParams
Window with actual editing tools
Rotation slider
Name1
Name2
Name3
SaveOne
SaveMulti

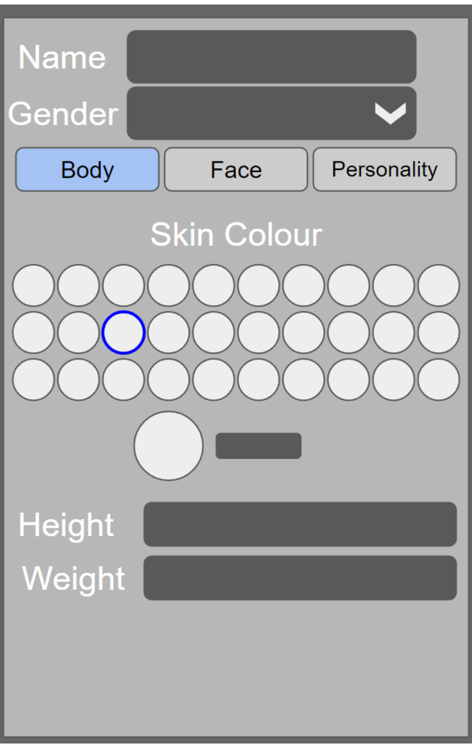
Bottom bar holds all created characters so far, with buttons showing them in small viewports, and their names as labels. Current selected character's button is highlighted blue. Also has the add character button, and saving buttons (for one or all characters)



When add character button is pressed, a small panel appears with 2 options: Adding a new default character, or opening the Genetic Mixing panel. GenMix is only enabled (not greyed out) if at least 2 previous characters are made. Both options grey out if at 10 character max limit



Genetic Mix panel opens. 2 drop boxes which hold available characters. Upon choosing from the drop box, the character fills in the parent image slot (and their name label). When both characters are selected, the Mix button becomes available. Upon selecting one parent, they are removed as an option from the other drop box.

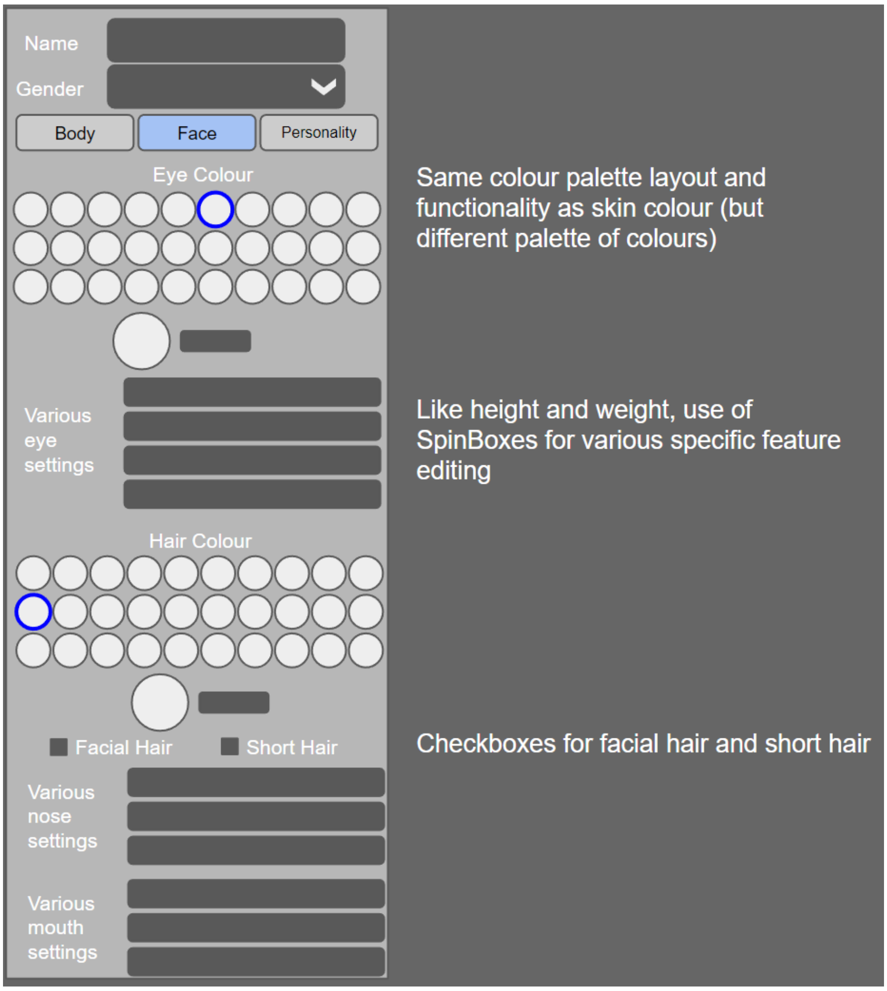


3 tabs which change content below

Name text box and gender drop box remain persistent at top of CharParam window, regardless of tab

Colour palette of 30 buttons. Selected colour button is highlighted blue. Big button at the bottom is a custom colour selector, using a single property view

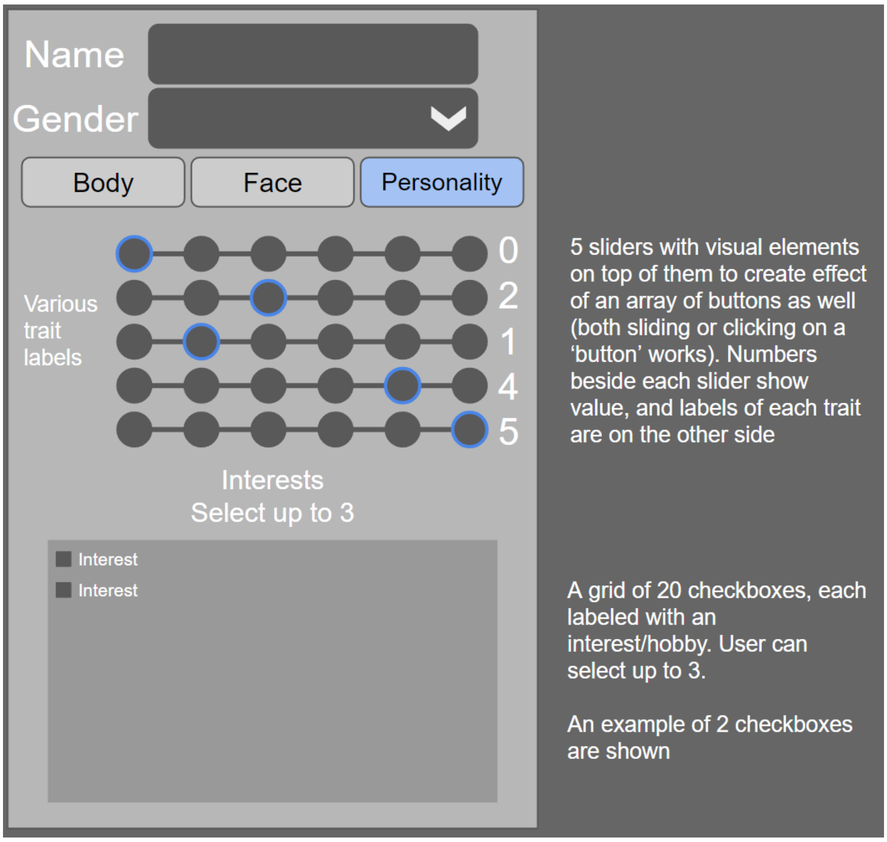
Height and weight sliders (uses SpinBox elements for ease of setting specific numbers as well)



Same colour palette layout and functionality as skin colour (but different palette of colours)

Like height and weight, use of SpinBoxes for various specific feature editing

Checkboxes for facial hair and short hair



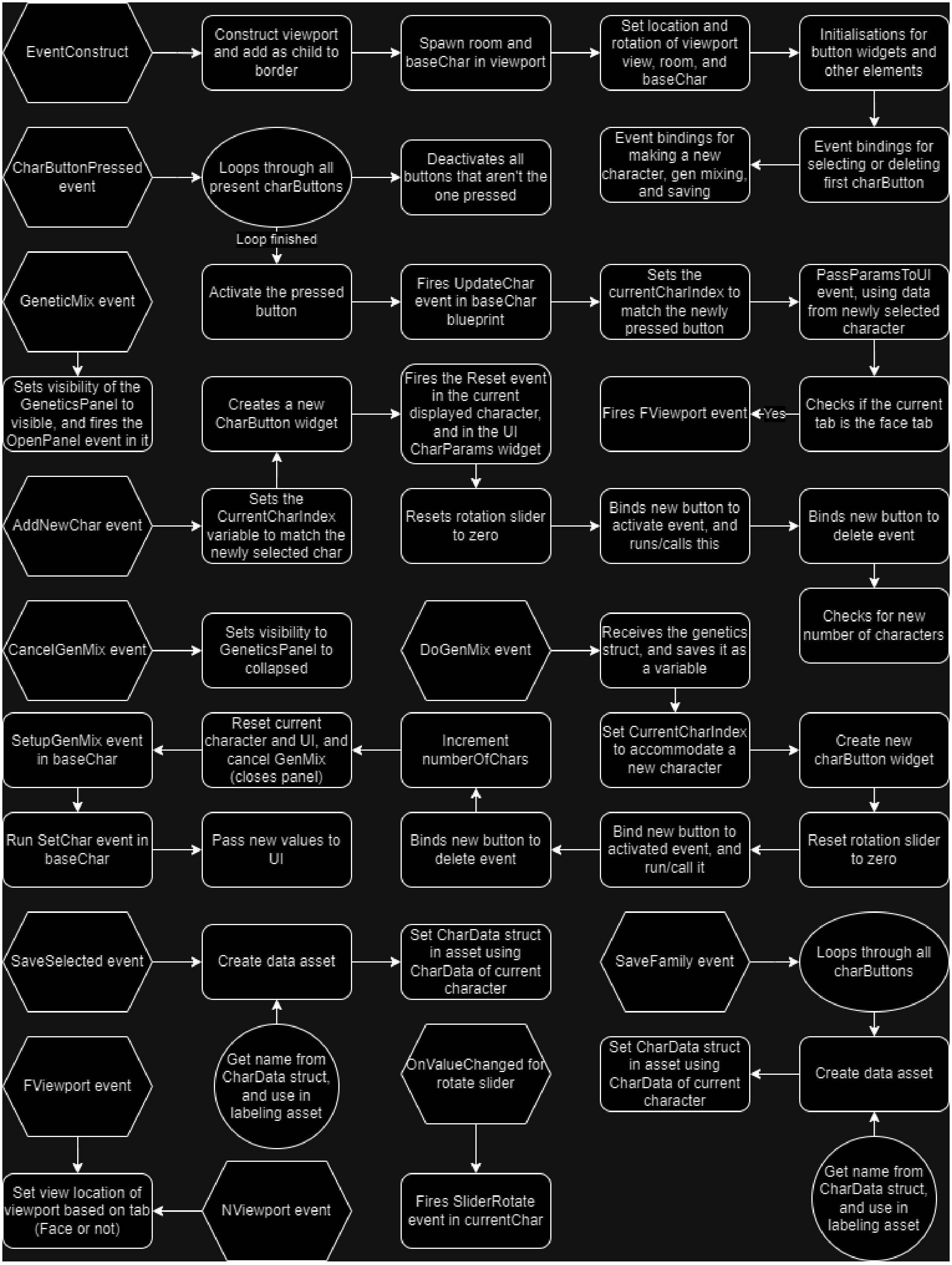
5 sliders with visual elements on top of them to create effect of an array of buttons as well (both sliding or clicking on a 'button' works). Numbers beside each slider show value, and labels of each trait are on the other side

A grid of 20 checkboxes, each labeled with an interest/hobby. User can select up to 3.

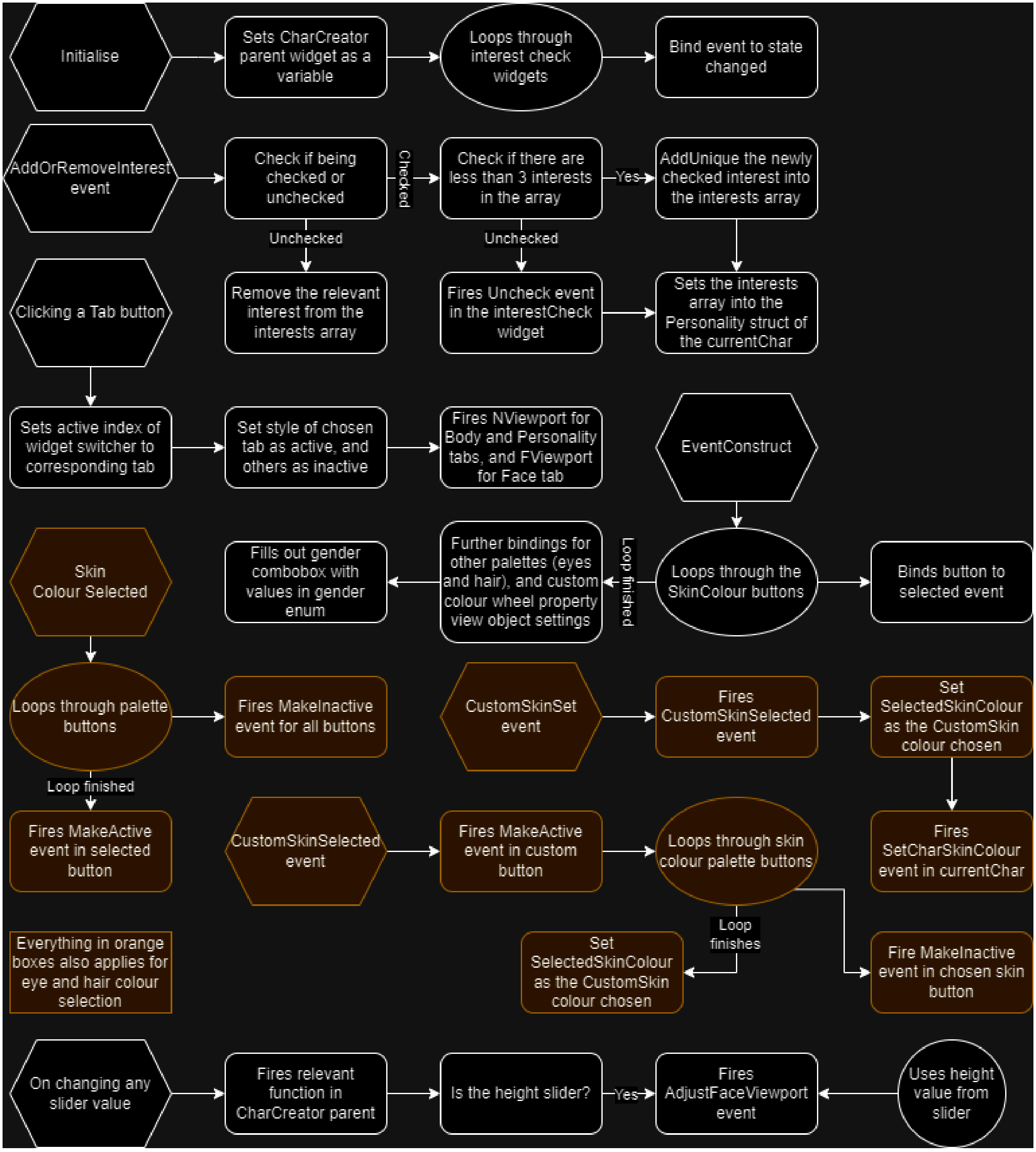
An example of 2 checkboxes are shown

Blueprint flowcharts

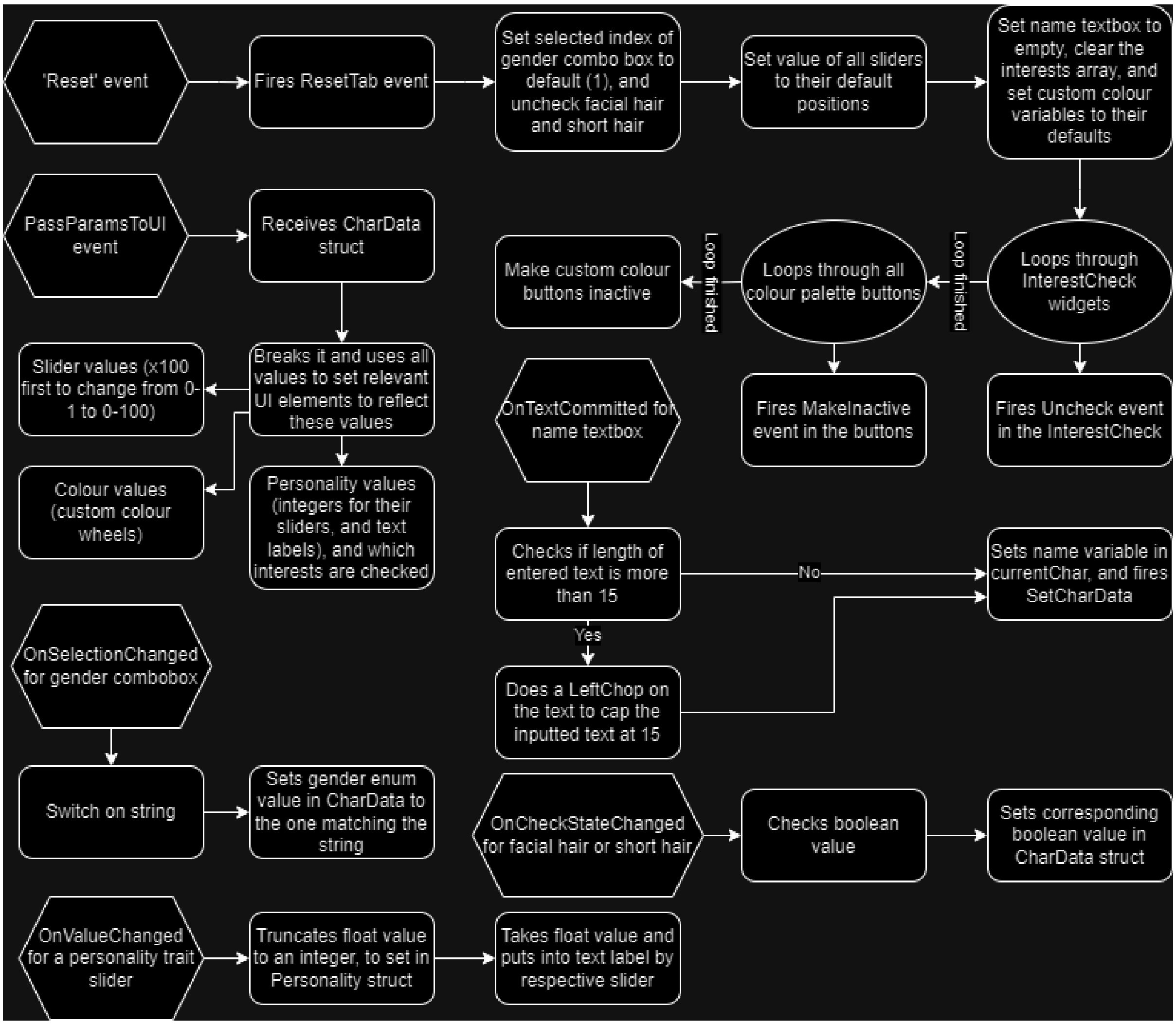
CharCreator main widget


EventConstruct
Construct viewport and add as child to border
Spawn room and baseChar in viewport
Set location and rotation of viewport view, room, and baseChar
Initialisations for button widgets and other elements
CharButtonPressed event
Loops through all present charButtons
Deactivates all buttons that aren't the one pressed
Event bindings for making a new character, gen mixing, and saving
Event bindings for selecting or deleting first charButton
Loop finished
GeneticMix event
Activate the pressed button
Fires UpdateChar event in baseChar blueprint
Sets the currentCharIndex to match the newly pressed button
PassParamsToUI event, using data from newly selected character
Sets visibility of the GeneticsPanel to visible, and fires the OpenPanel event in it
Creates a new CharButton widget
Fires the Reset event in the current displayed character, and in the UI CharParams widget
Fires FViewport event
Yes
Checks if the current tab is the face tab
AddNewChar event
Sets the CurrentCharIndex variable to match the newly selected char
Resets rotation slider to zero
Binds new button to activate event, and runs/calls this
Binds new button to delete event
CancelGenMix event
Sets visibility to GeneticsPanel to collapsed
DoGenMix event
Receives the genetics struct, and saves it as a variable
Checks for new number of characters
SetupGenMix event in baseChar
Reset current character and UI, and cancel GenMix (closes panel)
Increment numberOfChars
Set CurrentCharIndex to accommodate a new character
Create new charButton widget
Run SetChar event in baseChar
Pass new values to UI
Binds new button to delete event
Bind new button to activated event, and run/call it
Reset rotation slider to zero
SaveSelected event
Create data asset
Set CharData struct in asset using CharData of current character
SaveFamily event
Loops through all charButtons
FViewport event
Get name from CharData struct, and use in labeling asset
OnValueChanged for rotate slider
Set CharData struct in asset using CharData of current character
Create data asset
Set view location of viewport based on tab (Face or not)
NViewport event
Fires SliderRotate event in currentChar
Get name from CharData struct, and use in labeling asset

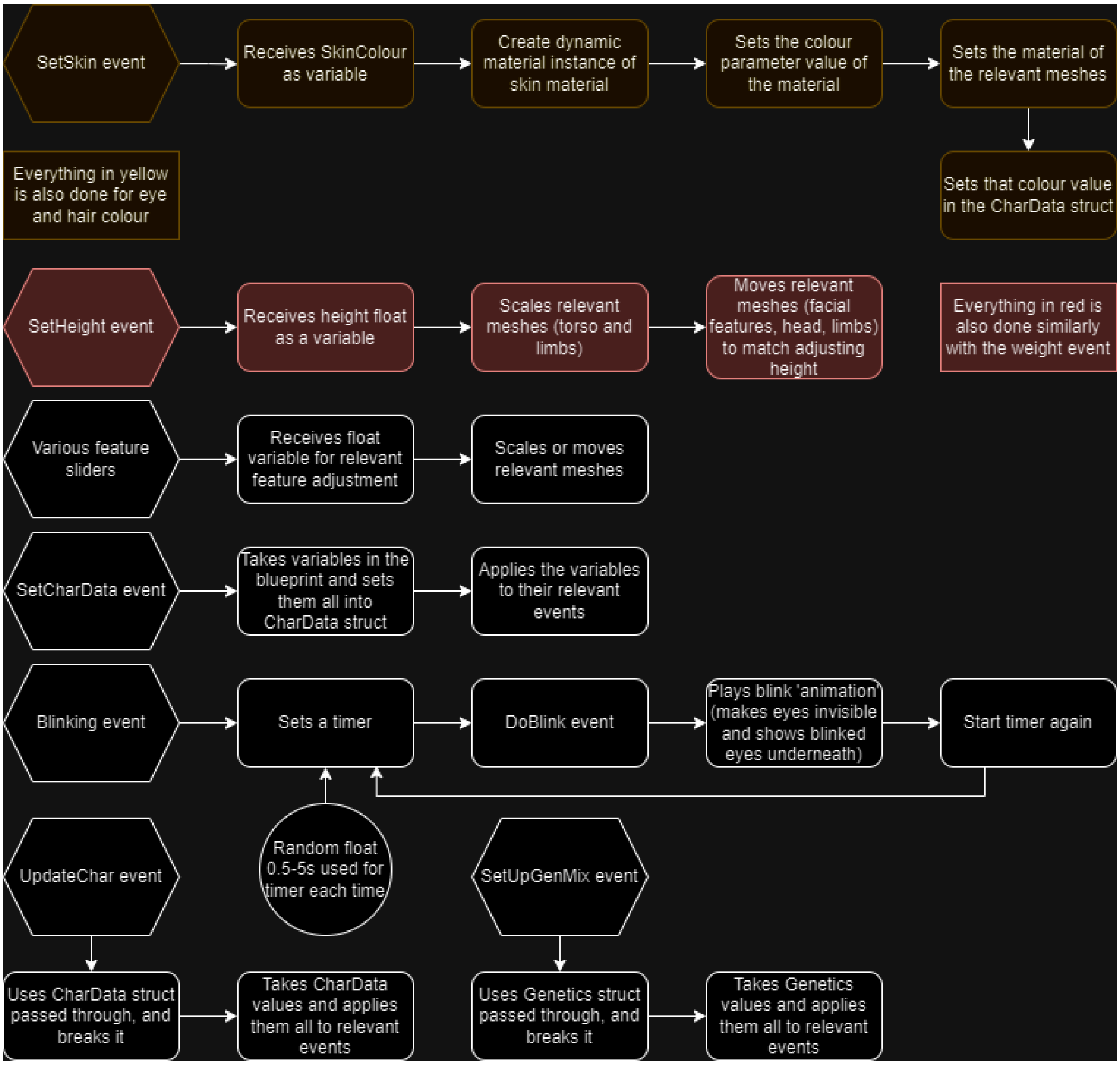
CharParams widget


Initialise
Sets CharCreator parent widget as a variable
Loops through interest check widgets
Bind event to state changed
AddOrRemoveInterest event
Check if being checked or unchecked
Checked
Check if there are less than 3 interests in the array
Yes
AddUnique the newly checked interest into the interests array
Unchecked
Unchecked
Clicking a Tab button
Remove the relevant interest from the interests array
Fires Uncheck event in the interestCheck widget
Sets the interests array into the Personality struct of the currentChar
Sets active index of widget switcher to corresponding tab
Set style of chosen tab as active, and others as inactive
Fires NViewport for Body and Personality tabs, and FViewport for Face tab
EventConstruct
Skin Colour Selected
Fills out gender combobox with values in gender enum
Further bindings for other palettes (eyes and hair), and custom colour wheel property view object settings
Loop finished
Loops through the SkinColour buttons
Binds button to selected event
Loops through palette buttons
Fires MakeInactive event for all buttons
CustomSkinSet event
Fires CustomSkinSelected event
Set SelectedSkinColour as the CustomSkin colour chosen
Loop finished
Fires MakeActive event in selected button
CustomSkinSelected event
Fires MakeActive event in custom button
Loops through skin colour palette buttons
Fires SetCharSkinColour event in currentChar
Everything in orange boxes also applies for eye and hair colour selection
Set SelectedSkinColour as the CustomSkin colour chosen
Loop finishes
Fire MakeInactive event in chosen skin button
On changing any slider value
Fires relevant function in CharCreator parent
Is the height slider?
Yes
Fires AdjustFaceViewport event
Uses height value from slider

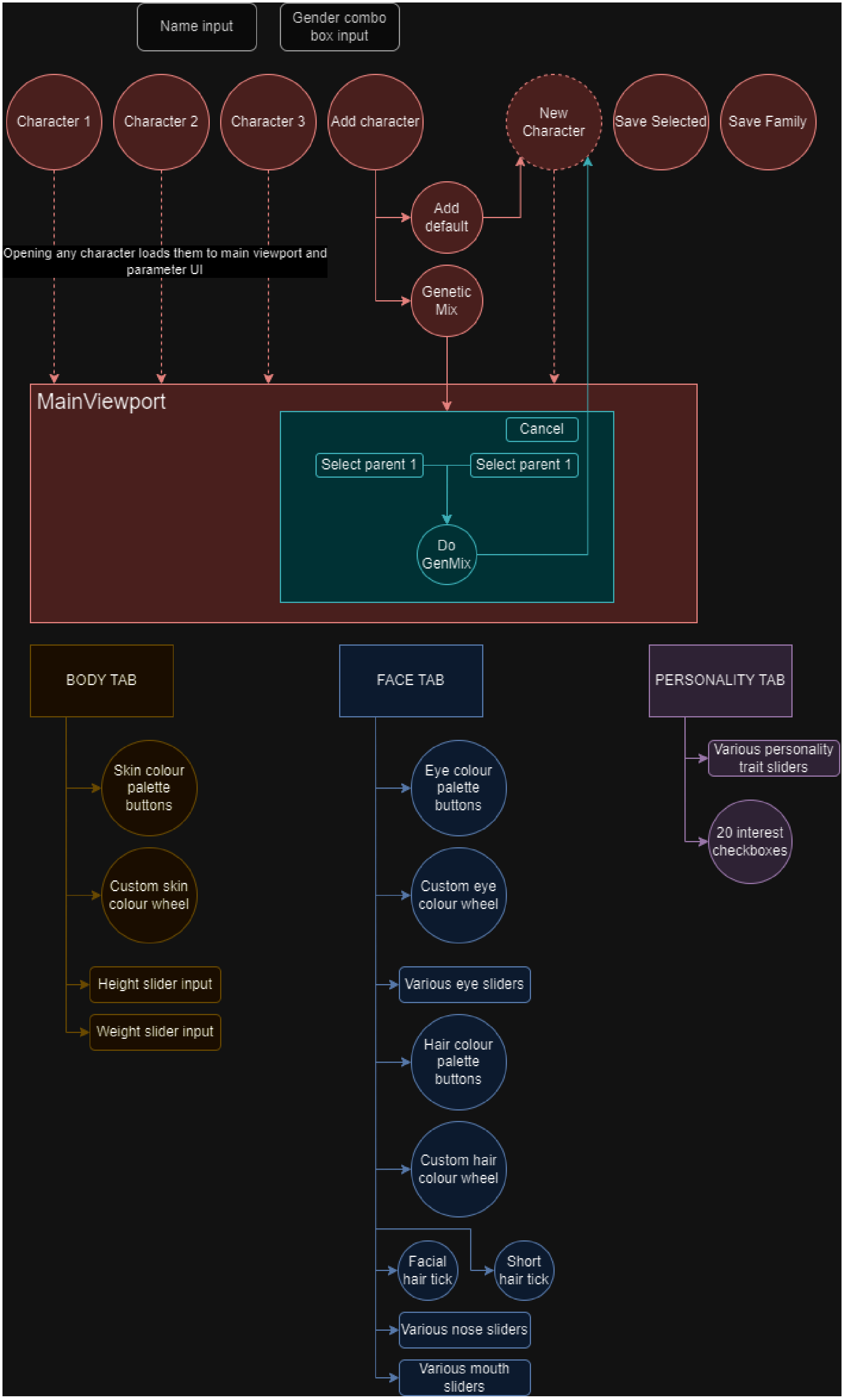
CharParams (continued on next page)


'Reset' event
Fires ResetTab event
Set selected index of gender combo box to default (1), and uncheck facial hair and short hair
Set value of all sliders to their default positions
Set name textbox to empty, clear the interests array, and set custom colour variables to their defaults
Loops through InterestCheck widgets
Loop finished
Loops through all colour palette buttons
Loop finished
Make custom colour buttons inactive
Fires Uncheck event in the InterestCheck
Fires MakeInactive event in the buttons
PassParamsToUI event
Receives CharData struct
Breaks it and uses all values to set relevant UI elements to reflect these values
Slider values (x100 first to change from 0-1 to 0-100)
Colour values (custom colour wheels)
Personality values (integers for their sliders, and text labels), and which interests are checked
OnTextCommitted for name textbox
Checks if length of entered text is more than 15
No
Yes
Sets name variable in currentChar, and fires SetCharData
Does a LeftChop on the text to cap the inputted text at 15
OnSelectionChanged for gender combobox
Switch on string
Sets gender enum value in CharData to the one matching the string
OnCheckStateChanged for facial hair or short hair
Checks boolean value
Sets corresponding boolean value in CharData struct
OnValueChanged for a personality trait slider
Truncates float value to an integer, to set in Personality struct
Takes float value and puts into text label by respective slider

BaseChar actor (next page)


SetSkin event
Receives SkinColour as variable
Create dynamic material instance of skin material
Sets the colour parameter value of the material
Sets the material of the relevant meshes
Sets that colour value in the CharData struct
Everything in yellow is also done for eye and hair colour
SetHeight event
Receives height float as a variable
Scales relevant meshes (torso and limbs)
Moves relevant meshes (facial features, head, limbs) to match adjusting height
Everything in red is also done similarly with the weight event
Various feature sliders
Receives float variable for relevant feature adjustment
Scales or moves relevant meshes
SetCharData event
Takes variables in the blueprint and sets them all into CharData struct
Applies the variables to their relevant events
Blinking event
Sets a timer
DoBlink event
Plays blink 'animation' (makes eyes invisible and shows blinked eyes underneath)
Start timer again
Random float 0.5-5s used for timer each time
UpdateChar event
Uses CharData struct passed through, and breaks it
Takes CharData values and applies them all to relevant events
SetUpGenMix event
Uses Genetics struct passed through, and breaks it
Takes Genetics values and applies them all to relevant events

Information Architecture diagram



Optimisation and Profiling

Profiling Systems

A planned testing session will occur on the 4th week of development (allowing testing of the first two tools, and implementation of feedback). Continued testing will occur throughout development as well, particularly nearer to the end of each tool's development (to allow actual testing of implemented features, and further improvement).

Coding Standards

Programming Standards

- Minimising use of casts
- Consistent use of naming convention prefixes, some examples:
  - S = data struct
  - TXT = texture
  - TEXT = text box
  - SB = size box
  - EUW = editor utility widget
  - BP = blueprint (general actor)
  - ENUM = enumerator

Style Guide

Commenting all blueprint (particularly larger / more complex blueprints) into cohesive sections, with clear and descriptive titles.

Also maintaining neat node connections and layout (especially considering the common use of loops and interlinking variables in these blueprints).

Production Overview

Moscow

Green: done, Red: not done

- MUST
  - Tool 1 – basic location movement, rotation functions, and mesh/material editing
  - Tool 2 – generation of a small simple map with a small range of features
  - Tool 3 – creation of simple characters with a small range of editable features/data
- SHOULD
  - Tool 1 – more complex location functions such as alignment, and also actor naming
  - Tool 2 – functionality of hex grid layout, and more of a range of features/biomes (desert, mountainous), and map saving/loading
  - Tool 3 – increased range of features editable in the character, character saving/loading
- COULD
  - Tool 1 – complex distribution function, folder functionality, lighting editing

  - Tool 2 – more robust function for the elevation deescelation method, and also allowing loaded maps to be moved
  - Tool 3 – genetic mixing functionality
- WOULD
  - Tool 1 – small preview viewports for editing actors, more robust editing for different actor types
  - Tool 2 - wider range of biomes (swamps/rainforests), addressing generation issues to allow larger maps easily, reworking tiles to not be actors but meshes of one larger actor
  - Tool 3 – animated characters, playable characters with stats that reflect in gameplay, family tree display

Timeline

Task	Start Date	Completion Date
Initial research/brainstorming	30/9/24	7/10/24
Developing Tool 1	5/10/24	14/10/24
Planning & Research Tool 2	15/10/24	20/10/24
Developing Tool 2	18/10/24	31/10/24
Planning & Research Tool 3	28/10/24	2/11/24
Developing Tool 3	1/11/24	20/11/24
Bug fixing and polish	21/11/24	26/11/24
Finalising documentation	21/11/24	29/11/24

Budgeting

8-9 weeks development time (overall, including research, planning, and documentation). Approximately 6 hours of work a day, on average.

Primarily one developer, with input and assistance for some bug fixing and mechanic questions from lecturers and fellow students.

No financial resources required, as any assets are either made myself (eg simple meshes for character creator) or used free asset packs (eg Kenney's Hex Tiles).