TECHNICAL DOCUMENTATION

for Crash Site 3D

Benjamin MARRINER
mar0072@mwsc.vic.edu.au

Data Tables

AccidentDatabase

- Description
    - The Accident Database holds all of the road accident records that the program starts with. There are over 4000 records that come shipped with Crash Site 3D. This data table is read-only and cannot be written to due to the lack of functionality in Unreal Engine 4 around manipulating data tables.
    - The database is loaded into the database array (which can be found in the Program Instance blueprint) at runtime and this is what Crash Site 3D will use when working with the car accident database.

Enumerations

AccidentRecordValues

- Description
    - Lists the names of all the properties in an accident record (i.e.: accident number, date, number of people involved, etc).
    - This list of options is mainly used in the GetColumnValue function from the CrashFunctions library.
- Property Names
    - Latitude
    - Longitude
    - Accident Date
    - Accident Time
    - Accident Day
    - Accident Number
    - Total People
    - Males
    - Females
    - Drivers
    - Young Drivers
    - Old Drivers
    - Motorists
    - Passengers
    - Pillions
    - Bicyclists
    - Pedestrians
    - Pedestrians aged 5-12
    - Pedestrians aged 13-18
    - Old Pedestrians
    - Unknowns
    - Injured or Fatal
    - Fatalities
    - Serious Injury
    - Other Injury
    - Non-Injured
    - Alcohol-Related
    - No. of Vehicles
    - Heavy Vehicles
    - Passenger Vehicles
    - Motorcycles
    - Public Vehicles
    - Unlicensed

CameraStates

- Description
    - This list contains the names of all the modes the camera can be in. This is used to keep track of the state the camera is in at any given moment.

- Camera Modes
    - Street Level
        - Camera is at street level
    - Transitioning
        - Camera is shifting to full map view or back to street level
    - Full Map
        - Camera is in a top view of the city

ToolSelection

- Description
    - A list of all the tools the user can make use of in order to operate the program. It is used to keep track of which tool is currently selected.
- Tools
    - Go To Location
        - Allows the user to click any spot on the map in order to move the camera to that location
    - Plot New Record
        - Allows the user to click any spot on the map to create a new accident record
    - Set New Accident Location
        - Similar to the Plot New Record tool, only that instead of creating a new record, it allows the user to change the geocoordinates of a record to a new geolocation. This tool is initiated from the NewAccidentRecordForm widget.

Structures

AccidentRecordInformation

- Description
    - Stores all the values of the properties that make up the accident record.
    - This structure only stores the record information from the database itself for each record.
- Properties
    - LATITUDE (String)
    - LONGITUDE (String)
    - ACCIDENT_DATE (String)
    - ACCIDENT_TIME (String)
    - DAY_OF_WEEK (String)
    - ACCIDENT_NO (String)
    - TOTAL_PERSONS (Integer)
    - MALES (Integer)
    - FEMALES (Integer)
    - DRIVER (Integer)
    - YOUNG_DRIVER (Integer)
    - OLD_DRIVER (Integer)
    - MOTORIST (Integer)
    - PASSENGER (Integer)
    - PILLION (Integer)
    - BICYCLIST (Integer)
    - PEDESTRIAN (Integer)
    - PED_CYCLIST_5_12 (Integer)
    - PED_CYCLIST_13_18 (Integer)
    - OLD_PEDESTRIAN (Integer)
    - UNKNOWN (Integer)
    - INJ_OR_FATAL (Integer)
    - FATALITY (Integer)
    - SERIOUSINJURY (Integer)
    - OTHERINJURY (Integer)
    - NONINJURED (Integer)
    - ALCOHOL_RELATED (Boolean)
    - NO_OF_VEHICLES (Integer)
    - HEAVYVEHICLE (Integer)
    - PASSENGERVEHICLE (Integer)
    - MOTORVEHICLE (Integer)
    - PUBLICVEHICLE (Integer)
    - UNLICENCSED (Integer)

SearchResultInformation

- Description
    - This structure stores the accident record information structure as a data value, enabling the use of a single record to be stored in it. In addition to this, it also stores a unique ID which corresponds to the row number of the record in the AccidentDatabase. Finally, it stores the converted vector coordinates of the accident. This is the structure that is used in all data processing.
- Properties
    - accidentID (Integer)
        - The row of the record from the AccidentDatabase
    - worldLocation (3D Vector)
        - The converted geocoordinates (latitude and longitude values) as a vector. The geocoordinates are converted using the GeocoordinatesToVector function in the CrashFunctions library
    - accidentRecord (AccidentRecordInformation)
        - The record from the AccidentDatabase

Blueprint Interfaces

BPI_ClickInteraction

- Description
    - This interface is meant to aid the communications between blueprints that require user input from the mouse.
    - It is commonly used when two blueprints of completely different classes needs to communicate with one another, or more precisely, the CrashUser blueprint and the BP_AccidentMarker blueprint.
- Events
    - BeginMouseOver
        - Signals an event to the receiving blueprint when the mouse goes over it.
    - EndMouseOver
        - Signals an event to the receiving blueprint when the mouse is no longer going over it.

Libraries

CrashFunctions

- SearchByAccidentNumber
    - *Inputs:*
        - accidentNumber (String)
            - The accident number to search for.
    - *Local Variables:*
        - searchSuccessful (Boolean)
            - Flag that is determined by whether a record was found or not.
        - searchResult (Search Result Information)
            - The search result that was found (blank if none was found)
    - *Description:*
        - Before the search begins, it removes all current search results using the RemoveAllSearchResults function.
        - The program will iterate through all the records in the accidentDatabase array until it finds a match with the accidentNumber input, in which case, the searchSuccessful flag will be set to true.
        - **searchSuccessful flag is true:** a record was found with the matching accident number. The program will add the record into the cachedAccidentWidgets map and then move the camera to the accident's location.
        - **searchSuccessful flag is false:** no records were found with the accident number inputted by the accidentNumber input. A dialog box will appear, telling the user that no records were found.
- GetColumnValue
    - *Inputs:*
        - column (Accident Record Values)
            - The property being retrieved from the record.
        - row (Search Result Information)
            - The record whose property is being retrieved from.
    - *Outputs:*
        - outColumn (Accident Record Values)
            - Returns the name of the property being retrieved from the record
        - result (String)
            - The value of the property being retrieved.
    - *Description:*
        - Allows for easy retrieval of a property in an accident record.
- SearchByAdvancedFilters
    - *Inputs:*
        - searchCriteria (Search Result Information)
            - A dummy record whose properties are used as the search criteria to search against (for instance, the total people being set to 4 would mean search for records where 4 people were involved in the accident)
        - criteriaStates (Array of Booleans)

        - Used to determine which criterion to search against.
        - Each checkbox in the SearchPanel corresponds to a Boolean in this array. The first 6 (indexes 0 to 5) are intentionally false as they represent properties that cannot be found in the advanced filters section (i.e.: accident number, date and time)
- *Local Variables:*
    - filtersToCheck (Array of Integers)
        - An array of integers that corresponds to which filters to run a search on (see SetFilterToggles in SearchPanel for more information on this).
    - totalPeopleResults (Set of Search Result Information)
    - malesResults (Set of Search Result Information)
    - femalesResults (Set of Search Result Information)
    - driversResults (Set of Search Result Information)
    - youngDriversResults (Set of Search Result Information)
    - oldDriversResults (Set of Search Result Information)
    - motoristsResults (Set of Search Result Information)
    - passengersResults (Set of Search Result Information)
    - pillionsResults (Set of Search Result Information)
    - Results (Set of Search Result Information)
    - bicyclistsResults (Set of Search Result Information)
    - pedestriansResults (Set of Search Result Information)
    - pedestrians512Results (Set of Search Result Information)
    - pedestrians1318Results (Set of Search Result Information)
    - oldPedestriansResults (Set of Search Result Information)
    - unknownsResults (Set of Search Result Information)
    - injuredFatalResults (Set of Search Result Information)
    - fatalitiesResults (Set of Search Result Information)
    - seriousInjuryResults (Set of Search Result Information)
    - otherInjuryResults (Set of Search Result Information)
    - nonInjuryResults (Set of Search Result Information)
    - alcoholRelatedResults (Set of Search Result Information)
    - noVehiclesResults (Set of Search Result Information)
    - passengerVehiclesResults (Set of Search Result Information)
    - motorcyclesResults (Set of Search Result Information)
    - publicVehiclesResults (Set of Search Result Information)
    - finalSearchResults (Set of Search Result Information)
        - The search results that satisfy all the criterion specified by the user.
    - heavyVehiclesResults (Set of Search Result Information)
    - unlicensedResults (Set of Search Result Information)
    - accidentVectors (Array of 3D Vectors)
        - Consists of all the vector coordinates of the search results' locations on the map. This is used to find the average (the overall centre) of all the locations.
- *Description:*
    - Before the search begins, it removes all current search results using the RemoveAllSearchResults function.

        - Individual searches are performed using each filter on its own and then populating the corresponding set of Search Result Information.
        - The finalSearchResults set is populated with all the records from the accidentDatabase array and it is compared against all of the sets of Search Result Information for common records, eliminating any records from the finalSearchResults set that are not shared in common with the set of Search Result Information it is being compared against.
        - By this point, any search results that are still in the finalSearchResults set are determined to be ones that meet all of the criterion provided by the user.
        - The vector coordinates of the locations of these accidents are then stored in the accidentVectors array so that the average (the overall centre) of the accidents' locations can be calculated.
        - The camera is moved accordingly to the average location of the search results obtained from the search.
        - If no search results are retained in the finalSearchResults set, the search will be considered unsuccessful and a dialog box will display telling the user that there were no records that matched the criterion.
    - *Procedures:*
        - Determine Indexes
            - For each Boolean in the criteriaStates array, those Booleans that are equal to true will have their index numbers added to the filtersToCheck array. This is enable the search function to know which filters to perform searches with.
        - Load Accident Records into Final Search Results Set
            - Copies the entire accidentDatabase array into the finalSearchResults set so that it can be compared against the individual search results sets.
        - Perform Searches against each criteria
            - Performs searches using each filter one at a time. For each record in the accidentDatabase array per search, the property that the search is looking for from the record is retrieved using the GetColumnValue function and compared with the value of the filter inputted by the user via the searchCriteria input.
            - If the property of the record matches what the user has inputted for the filter, it is added to the corresponding search result set (e.g.: a search for records with a value of 1 for total people would result in the record being added to the totalPeopleResults set).
            - FilteredSearchResults
                - Uses the outColumn output of the GetColumnValue function to choose which search results set to add the record to. The select node has been configured to reflect the list order of properties in the AccidentRecordValues enumeration.
        - Find Common Results from Search

        - Compares the finalSearchResults set with each individual search result set by finding the intersection of the sets. The result of the intersection of two sets will become the new finalSearchResults set. The process repeats until all filters have been accounted for. Records that are still left after the process has finished will be the search results shown on the map.
      - Cach Search Results
        - Moves the records in the finalSearchResults set to the cachedAccidentWidgets map. The vector coordinates of the location of the accident are also added to the accidentVectors array.
        - The number of search results are displayed on screen if the program is in debugging mode.

- TextboxToInteger
  - *Inputs:*
    - textbox (Text Box)
      - The textbox whose value we want to convert
  - *Outputs:*
    - integer (Integer)
      - The value of the textbox we have converted.
  - *Description:*
    - Gets the value of a textbox widget and converts it to an integer. Since we can't convert values in the Text datatype directly to an integer, it converts the textbox's value to a string first, and then to an integer.
- GeocoordinatesToVector
  - *Inputs:*
    - Latitude (String)
      - The latitude geocoordinate to convert. Converts to the y-coordinate.
    - Longitude (String)
      - The longitude geocoordinate to convert. Converts to the x-coordinate.
  - *Outputs:*
    - worldLocation (3D Vector)
      - The geolocation converted to a set of vector coordinates
  - *Description:*
    - This function converts geocoordinates to vector coordinates so that accidents can be plotted on the map correctly.
    - The geocoordinate values are normalised to a value between 0 and 1 using a the coordinate range -37.8506584 to -37.7754478 for latitude and 144.8969574 to 144.9913025 for longitude. These coordinate ranges make up the reference frame for the Melbourne CBD.
    - The number between 0 and 1 is then passed through a lerp which converts it back a vector coordinate form of the reference frame.
    - Essentially, this function works out the equivalent vector coordinates of a set of latitude and longitude coordinates using the reference frame as a guide.

- RemoveAllSearchResults
    - *Description*
        - Clears all current search results from the map. This includes running the RemoveAllAccidentMarkers function in the BP_CityModel and clearing out the cached and renderedAccidentWidgets maps.
- mousePositionToWorldVector
    - *Inputs:*
        - TraceType (Trace Types)
            - Which collision test to perform with line tracing.
        - DrawDebugType (EDraw Debug Trace)
            - How long to show the line on screen for (debug only)
    - *Outputs:*
        - worldLocation (3D Vector)
            - The location of the mouse in 3D space
        - out Hit Hit Actor (Actor)
            - The actor of which the mouse is currently pointing at
        - Out Hit Hit Component (Primitive Component)
            - The primitive component of which the mouse is currently pointing at
    - *Description:*
        - A line is casted from the camera in the direction the mouse is pointing.
- TextToInteger
    - *Inputs:*
        - text (Text)
            - The text to convert to an integer
    - *Outputs:*
        - integer (Integer)
            - The converted text value in integer form
    - *Description:*
        - Since text cannot be directly converted to an integer, this function converts it to a string and then to an integer.
- VectorToGeocoordinates
    - *Inputs:*
        - worldLocation (3D Vector)
            - The vector coordinates to be converted to geocoordinates. the x-coordinate converts to longitude while the y-coordinate converts to a latitude.
    - *Outputs:*
        - Latitude (Float)
            - The latitude coordinate of the converted y-coordinate
        - Longitude (Float)
            - The longitude coordinate of the converted x-coordinate
    - *Description:*
        - The reverse function of the GeocoordinatesToVector function. It uses the vector coordinate form of the reference frame to find a value between 0 and 1 using the worldLocation coordinates and then finding

    - its equivalent geocoordinates using the lerp and the geocoordinate reference frame.
    - See GeocoordinatesToVector for more information on this algorithm.

- DetermineNewAccidentNumber
  - *Inputs:*
    - year (String)
      - The year of which we are trying to determine the new accident number for
  - *Outputs:*
    - accidentNumber (String)
      - The new accident number determined from the year inputted.
  - *Local Variables:*
    - accidentNumbersFromSpecifiedYear (Array of Integers)
      - The last 7 digits of the accident numbers in all the records in the accidentDatabase array whose accident is set in the same year as the one in question.
    - zeroDigits (String)
      - The missing zero digits that are removed when the last 7 digits of the accident number are converted to an integer upon being put in the accidentNumbersFromSpecifiedYear array.
  - *Description:*
    - The DetermineNewAccidentNumber function constructs a new accident number by using the existing records from the accidentDatabase array to determine what the new accident number should be called.
    - It does this by iterating through the accidentDatabase array, getting the records that are from the year set in the the year input and loading the last 7 digits of those records into the accidentNumbersFromSpecifiedYear array.
    - After this, the maximum number of the accidentNumbersFromSpecifiedYear array is found and incremented by one. This will be the integer value of the new accident number in integer form.
    - During the process of adding the last 7 digits of each record of the year in question, the digits would have been converted to an integer and it is due to this that the zero digits would be removed. These are recovered by working out the length of the characters of the integer value of the new accident number and subtracting it from 7 (because there are 7 digits). This works out the number of missing zero digits.
    - These zeroes are stored in the zeroDigits variable and are used as part of the Append function when the final accident number gets constructed.
    - The new accident number is constructed, starting with a T, then the year from the year input, the zero digits and finally, the integer form of the remaining numbers.

CrashMacros/CrashComponentMacros

- Description
  - This is a library of macros that can be used throughout all the blueprints that inherit from Actor class.

    - The CrashComponentMacros library is exactly the same as the CrashMacros library only that it makes it possible to use the macros in blueprints that are inherited from the ActorComponent class instead.

- Macros
    - *GetProgramUtilities*
        - Outputs:
            - programInstance (ProgramInstance)
                - Returns the game instance class aka the ProgramInstance blueprint.
            - database (Search Result Information)
                - Returns the accidentDatabase array from the ProgramInstance blueprint.
            - accidentManager (BP_AccidentManager)
                - Returns the accidentManager component from the ProgramInstance blueprint.
            - city (BP_CityModel)
                - Returns the city variable from the ProgramInstance blueprint.
        - Description:
            - GetProgramUtilities is a macro that allows all-in-one access to the most common global variables and components used throughout Crash Site 3D.
- MapTransfer
    - *MapTransfer*
        - Inputs:
            - map1 (Map, Key: Integer, Value: SearchResultInformation)
                - The map from which the item is being transferred
            - itemToMove (Integer)
                - The item that is being transferred
            - map2 (Map. Key: Integer, Value: SearchResultInformation)
                - The map to which the item is being transferred
        - Description:
            - This macro takes in two maps and an item's key so that the item as well as its key can be transferred from one map to another.
            - It finds the item in map1 using the key in the itemToMove input, adds the item and the key to map2 and then removes the item and key from map1.

Program Blueprints

BP_AccidentManager

- Description
    - The Accident Manager is in charge of keeping track of which accident search results are closest to the camera and which ones are the farthest. Search results that are beyond a certain distance from the camera are not rendered onscreen for optimisation purposes
    - Since this blueprint inherits the actor component class, it needs to be attached to another actor before it can function. In the case of Crash Site 3D, it is attached the the BP_CityModel in the AccidentMap level.
- Variables
    - cachedAccidentWidgets (Map, Key: Integer, Value: SearchResultInformation)
        - Search results that are too far away from the camera will have their records stored in this map until the camera comes into range of them again.
    - renderedAccidentWidgets (Map, Key: Integer, Value: SearchResultInformation)
        - Search results that are currently in range of the camera.
- Functions
    - *RemoveAccidentWidget*
        - Inputs:
            - Key (Integer)
                - Used for looking up the record in the cached or renderAccidentWidgets map
        - Description:
            - Whenever an accident needs to be removed from the world, this function is called so that it can be done safely
    - *UpdateAccidentWidget*
        - Inputs
            - Key (Integer)
                - Used for looking up the record in the cached or renderAccidentWidgets map
            - recordInformation (Search Result Information)
                - The record information to update the accident widget with
        - Description:
            - When a car accident record is saved, this function is called so that the marker's record information can be updated.
            - It updates the record information that's in either the rendered or cachedAccidentWidgets map (most likely the renderedAccidentWidgets map) and then it updates the marker itself.
- Procedures
    - GetInRangeResults
        - Stores all search results in the cachedAccidentWidgets map, then works out which of those records are in range of the camera. Those that are in range are transferred to the renderedAccidentWidgets map.
    - DisplayInRangeResults

        - For each of the records in the renderedAccidentWidgets map, the SpawnAccidentMarker function from the BP_CityModel blueprint is called so that the record can be plotted on the map.
    - TrackCachedAndRenderedMarkers (Debug only)
        - Constantly prints the number of accident records stored in both the cached and renderedAccidentWidgets arrays to the screen.

BP_AccidentMarker

- Description
    - The Accident Marker blueprint represents an accident record that has been plotted on the map after performing a search.
- Variables
    - widgetInformation (SearchResultInformation)
        - The accident record that the marker will configure itself for.
    - widgetID (integer)
        - Used to uniquely identify the accident marker. It is also the same as the record's row number in the AccidentDatabase
- Components
    - AccidentMarker (Billboard Component)
        - Editor only. Symbol for component that displays 2D textures in 3D space
    - AccidentNumberText (Text Render Component)
        - Displays the accident number of the record
    - HitBox (Box Collision)
        - An invisible box that detects when the cursor is on the marker so that a AccidentDetailsWidget can be displayed.
    - accidentWidget (Widget component)
        - Keeps track of (if one exists) the AccidentDetailsWidget that might be on the screen at any given moment.
- Events
    - ConfigureAccidentMarker
        - With the record information provided, the marker will set the title displayed by the AccidentNumberText to the given accident number and in the event when the mouse goes over the marker, it will be able to show the correct information on the AccidentDetailsWidget.
    - BeginMouseOver (see BPI_ClickInteraction)
        - Makes its AccidentDetailsWidget visible
    - EndMouseOver (see BPI_ClickInteraction)
        - Makes its AccidentDetailsWidget no longer visible
- Procedures
    - RenderText
        - **At Street Level:** The position of the camera and the position of the marker are used to calculate the direction in which the marker must face in order to be facing the camera.
        - **During Transition:** The marker is made invisible to the camera.
        - **In Full Map View:** Adjusts the appearance of the marker so that it can be seen correctly by the user (direction, font size, font colour and height off of the ground so that it can be rendered correctly).
    - GetOutOfRangeResults

        - Provided that the camera is in street level mode, the marker will work out whether it is too far away from the camera to be rendered. If it is too far away, the marker will look up the record in the renderedAccidentWidgets map from the accident manager using the marker's assigned widgetID and transfer it back to the cachedAccidentWidgets map. It will then proceed to call the RemoveAccidentMarker function from the BP_CityModel to safely remove the marker from the map.
    - ConfigureMarker
        - Using the record it has been provided (contained in the widgetInformation variable), it will set the Accident Number from it as the AccidentNumberText text. It will also calculate the size of the HitBox's dimensions according to the length of the text.
    - MouseOverMarker
        - Creates an AccidentDetailsWidget at the location of the marker and configures it with the information that the marker was given. The AccidentDetailsWidget is then stored in the accidentWidget variable so that the widget can be removed from the screen later.
    - EndMouseOverMarker
        - Removes the AccidentDetailsWidget from the screen via the accidentWidget variable.

BP_CityModel

- Description
    - The City Model blueprint contains the entire City of Melbourne model. It is also responsible for plotting the accident records that come up in the search results.
    - At runtime, this blueprint, along with its BP_AccidentManager component is set as the city variable and accidentManager component from the ProgramInstance blueprint respectively
- Components
    - Melbourne Model meshes (Static Mesh Components)
        - The city model is made up of multiple meshes. Each mesh is a chunk of the entire City of Melbourne. The reason for this is that there are over 20,000 structures in the model. Even the most powerful of systems would run Crash Site 3D terribly because 10s of thousands of draw calls had to be made each frame. Dividing the model into chunks makes it much more manageable for not just those kinds of systems but also for ones that meet the minimum hardware requirements.
    - BP_AccidentManager (BP_AccidentManager)
        - BP_AccidentManager blueprints inherit from actor components and can therefore not function without being attached to another actor first.
- Variables
    - renderedMarkers (Map, Key: Integer, Value: Child Actor Components that use the BP_AccidentMarker class)
- Functions
    - *SpawnAccidentMarker*
        - Inputs
            - widgetInfo (Search Result Information)
                - The record information to give the accident marker

        - Description
            - Checks if a marker with this accident record doesn't exist on the map already and if it doesn't, it will plot a marker on the map and add it to the renderedMarkers map to keep track of.
    - *RemoveAccidentMarker*
        - Inputs
            - widgetID (integer)
                - The unique ID of the marker. This will point it to the correct marker in the renderedMarkers map.
        - Description
            - Looks up the ID of the marker to be removed and deletes the marker.
    - *RemoveAllAccidentMarkers*
        - Description
            - Performs the RemoveAccidentMarker function on all of the markers in the renderedMarkers map. It also clears out the left-over IDs in the map so the map variable is empty.
    - *UpdateAccidentMarker*
        - Inputs
            - Key (Integer)
                - Used for looking up the marker in the renderedMarkers map
            - recordInformation (Search Result Information)
                - The record information to update the accident marker with
        - Description
            - Looks up the marker that needs to be updated based on the record that is currently being edited and calls its ConfigureAccidentMarker function to reconfigure itself.

ProgramInstance

- Description
    - The ProgramInstance blueprint is responsible housing all of the procedures that globally affect the program. For instance, it is responsible for knowing which start-up procedure to perform when the user executes the program as well as handling saving of the database to the hard drive and storing it in memory.
- Components
    - accidentManager (BP_AccidentManager)
        - The accident manager component of the BP_CityModel in the AccidentMap level.
- Variables
    - database (Data Table)
        - A reference to the AccidentDatabase Data Table. The accidentDatabase array is populated with all the records from this variable during the First-time Setup Procedure.
    - city (BP_CityModel)
        - The AccidentMap level's BP_CityModel.
    - accidentDatabase (array of Search Result Information)

        - The database array that is used by the program to read and write to new and existing accident records as well as perform searches upon.
    - save (BP_Save)
        - A reference to the blueprint that defines what is saved onto the hard drive. For Crash Site 3D, this is just an instance of the accidentDatabase array.
    - don’tSave (Boolean) (Debug only)
        - Forces the First-time Setup Procedure to occur every time on startup.
    - debuggingMode (Boolean) (Debug only)
        - Enables various debugging features in the program such as printing specific values of records onto the screen and real-time tracking of the number of cached markers and rendered markers.
    - currentCameraLocation (3D Vector)
- Functions
    - *SaveDatabase*
        - Description
            - Sets the userAccidentDatabase array of the BP_Save class in the save variable to the accidentDatabase array in the ProgramInstance blueprint. It then saves to the hard drive.
            - This function is usually called when a change has been made to a record or a new one has been created.
- Procedures
    - Load Database
        - Loads the database into memory by loading all records into the accidentDatabase array via the BP_Save’s userAccidentDatabase array.
    - First-time Setup Procedure
        - Retrieves all the accident records from the database Data Table and populates the accidentDatabase array with them. The SaveDatabase function is then executed to create the save file and save the database to the hard drive.

AccidentMap

- Description
    - AccidentMap is the level that is used by the program to actually display everything. The entire program’s functionality takes place inside this level.
    - It also contains its own blueprint which initialises the program on startup.
- Referenced Level Actors
    - BP_CityModel (BP_CityModel)
        - The model of the city
- Events
    - Event BeginPlay
        - Configures the city and accidentManager variables in the ProgramInstance blueprint to reference the ones from the level

User Blueprints

BP_Save

- Description
    - The BP_Save blueprint contains all of the information that needs to be saved to the user's hard drive. Apart from this, it does not really do anything else
- Variables
    - userAccidentDatabase (array of Search Result Information)
        - The database array that is saved to the user's hard drive

CrashGameMode

- Description
    - This is a data-only blueprint that sets the default HUD and pawn classes when the program loads. In other words, it tells the program to use the CrashHUD blueprint for the user interface and CrashUser blueprint for the user's controls when the program starts.

CrashHUD

- Description
    - This blueprint is responsible for displaying the user interface on the screen. It starts by creating a CrashInterface Widget and storing it in the HUD variable for later reference by other parts of the program and then displaying it on screen.
    - It is also responsible for enabling the use of the cursor.
- Variables
    - HUD (CrashInterface)
        - The CrashInterface widget being displayed on screen

CrashUser

- Description
- Components
    - CapsuleComponent (Capsule Collision)
    - ArrowComponent (Arrow Component)
    - Mesh (Skeletal Mesh Component
    - Sphere (Static Mesh Component)
        - Used to represent the location of the camera on the map
    - SpringArm (Spring Arm Component)
        - The camera is attached to this to keep the camera oriented correctly
    - Camera (Camera Component)
        - The view of the user
    - GeocoordinatesDisplay (Widget Component)
        - When picking a new location for a record, this component stores the GeocoordinatesDisplay widget that is on the map.
    - Transition (Timeline Component)
        - The timeline that gradually increases/decreases the value used to interpolate the camera between street level and full map view.
    - CharacterMovement (Character Movement Component)
- Variables
    - mouseLocationInWorldSpace (3D Vector)
        - The vector coordinates of the location the mouse is currently pointing to

- currentMouseOverActor (Actor)
    - The actor that the mouse is currently pointing to
- previousMouseOverActor (Actor)
    - The actor that the mouse had previously been pointing to
- currentToolSelection (ToolSelection)
    - The current tool that the user has selected
- isMenuOpen (Boolean)
    - A flag that keeps track of whether a window of some sort is currently open. This is used to temporarily disable the user's navigation controls
- cameraState (CameraStates)
    - The current state of the camera. Street level, transitioning or Full map view
- previousCameraTransform (3D Transform)
    - Contains the location and rotation of the camera before it was put into full map view by the user
- previousArmLength (Float)
    - The spring arm length of the spring arm component before the camera was put into full map view by the user
- isInFullMap (Boolean)
    - A flag that keeps track of whether the camera is in full map view or not

- Functions
    - *CreateGeocoordinatesDisplay*
        - Inputs:
            - searchResultInformation
                - The record that is currently being edited
        - Description:
            - Creates a Geocoordinates Display widget to be used for picking the new location of an accident. It creates the widget component, loads the record information from the searchResultInformation input into it and is then stored in the GeocoordinatesDisplay component.
    - *DestroyGeocoordinatesDisplay*
        - Description:
            - Gets the GeocoordinatesDisplay component and destroys it, effectively removing it from the user's screen
- Macros
    - *isMouseClickAllowed*
        - Outputs:
            - ReturnValue
                - Returns whether the mouse click will have an effect or not.
        - Description:
            - Determines whether a mouse click will do anything depending on the current circumstances of whether a menu is currently open and whether the mouse is currently pointing to an actor.
- Events
    - InputAction LeftMouseAction
        - Event fired by the left mouse button
        - The appropriate procedures are fired based on the currently selected tool

        - Left mouse clicks are blocked if the mouse is currently pointing at a BP_AccidentMarker
    - InputAxis TurnRight
        - Event fired from moving the mouse left or right
        - If the camera is in full map view, it triggers the Panning Control procedure, otherwise, it rotates the camera provided the right mouse button is being held down as well as a menu not being open.
    - InputAxis LookUp
        - Event fired from moving the mouse up or down
        - If the camera is in full map view, it triggers the Panning Control procedure, otherwise, it rotates the camera provided the right mouse button is being held down as well as a menu not being open.
    - InputAxis Zoom
        - Event fired from scrolling up or down with the mouse wheel
        - On the conditions that there is no menu open and the mouse is currently scrolling, the appropriate procedure is triggered based on the current camera state.
    - SwitchView
        - Event fired from hitting the btnSwitchView button. This will set of the procedure that transitions the camera from one view to another.
    - BeginPlay
        - Code to execute on program start
        - Enables mouse over events and sets the limits of which the camera can be rotated up or down.
    - Tick
        - Code to execute every frame
        - Keeps track of which actor the mouse is pointing at, which actor it was previously pointing at as well as the location of the mouse.
        - In charge of deciding when and which actors to call the BeginMouseOver and EndMouseOver events on.
        - If a GeocoordinatesDisplay is present, the event is in charge of keeping the display attached to the mouse.
- Procedures
    - Set Current and Previous Mouse Over Actors
        - Checks to see whether the nextMouseOverActor input is not the same as the actor stored in the currentMouseOverActor variable.
        - If nextMouseOverActor is not the same as the currentMouseOverActor, the currentMouseOverActor becomes the previousMouseOverActor and then the nextMouseOverActor becomes the current.
        - If they are the same however (the mouse is pointing at the same actor as before) then the nextMouseOverActor becomes the current.
    - Panning Control
        - Math Equation: $\frac{13}{3950}x + \frac{112000}{79}$
            - Used to calculate how many units the camera should move. It takes into account the orthographic width value of the camera as its input and outputs the base number of units.

        - Takes into account the Lookup and TurnRight axis values to calculate how far to move the camera in full map view.
        - It also keeps the camera within the bounds of the map
    - Zoom In/Out
        - Calculates the number of units the camera's ortho width value should increase/decrease by when scrolling with the mouse wheel in full map view
    - Increase/Decrease Spring Arm Length
        - Calculates the number of units the spring arm's target arm length value should increase/decrease by when scrolling with the mouse wheel at street level
    - Camera Animation
        - Interpolates the spring arm's target arm length value between the previousArmLength value and the arbitrary value of 650000.
        - Interpolates the camera's rotation value between the previousCameraTransform's rotation value and the arbitrary value of X: -180, Y: -90, Z: -90.
    - Set Camera Mode Settings
        - Post-transition configurations applied depending on which camera state the camera has transitioned to.
        - Upon being in full map view, the camera state is set accordingly and various camera property values are set.
        - Upon being at street level, only the camera state needs to be set accordingly.
    - Pre-Transition Code
        - In charge of setting various different values before performing the camera transition animation.
    - Click on Location
        - Sets the camera to the location being pointed at by the mouse
    - Click on Location From Full Map View
        - Hijacks the previousCameraTransform value and changes it to the location being pointed at by the mouse. It then fires the SwitchView event so that the camera can transition to that new location.
    - Plot New Record
        - Constructs a new record with initialised values, using the mouse's current location for the geocoordinates and the DetermineNewAccidentNumber function for the year of the new record.
        - The record is then injected into the cachedAccidentWidgets map so that it can be displayed on the map.
        - A NewAccidentRecordForm is created and displayed on screen using this new record.
        - The current tool selection is set back to Go To Location so that the user cannot mistakenly plot another new record should they do not wish to.
    - Set New Accident Location
        - Calls the SetAccidentCoordinates function from the GeocoordinatesDisplay component in order to set the new coordinates of the accident record.
        - The current tool selection is set to Go To Location.

Widget Blueprints

AccidentDetailsWidget

- Description
  - Displays basic information about the accident record, namely its date, time, day, accident number and geolocation.
- Variables
  - recordInformation (Search Result Information)
    - The record information to display in the widget
  - lblAccidentDateValue (Text)
    - Value of the Accident Date
  - lblAccidentDayValue (Text)
    - Value of the Accident Day
  - lblAccidentNumberValue (Text)
    - Value of the Accident Number
  - lblAccidentTimeValue (Text)
    - Value of the Accident Time
  - lblLatitudeValue (Text)
    - Value of the latitude geocoordinate
  - lblLongitudeValue (Text)
    - Value of the longitude geocoordinate
- Functions
  - *WidgetDoubleClicked*
    - Inputs
      - MyGeometry (Geometry)
      - MouseEvent (Pointer Event)
    - Outputs
      - ReturnValue (Event Reply)
    - Description
      - When the widget is double clicked, it will create a new accident details window to show the full details and statistics of the record.
- Events
  - ConfigureWidget
    - Takes the record information from its recordInformation input and sets the values on the widget

AccidentDetailsWindow

- Description
  - A window display the accident record's full details and statistics
- Variables
  - btnClose (Button)
  - btnEditRecord (Button)
  - lblAccidentDateValue (Text)
    - Value of Accident Date
  - lblAccidentDayValue (Text)
    - Value of Accident Day

- lblAccidentNumberValue (Text)
  - Value of Accident Number
- lblAccidentTimeValue (Text)
  - Value of Accident Time
- lblAlcoholRelatedValue (Boolean)
  - Value of Alcohol-Related
- lblBicyclistsValue (Text)
  - Number of Bicyclists
- lblDriversValue (Text)
  - Number of Drivers
- lblElderlyDriversValue (Text)
  - Number of Elderly Drivers
- lblElderlyPedestriansValue (Text)
  - Number of Elderly Pedestrians
- lblFatalitiesValue (Text)
  - Number of Fatalities
- lblFemalesValue (Text)
  - Number of Females
- lblHeavyVehiclesValue (Text)
  - Number of Heavy Vehicles
- lblInjuredFatalValue (Text)
  - Number of injured and killed victims
- lblLatitudeValue (Text)
  - Value of the Latitude geocoordinate
- lblLongitudeValue (Text)
  - Value of the Longitude geocoordinate
- lblMalesValue (Text)
  - Number of males
- lblMotorcyclesValue (Text)
  - Number of motorcycles
- lblMotoristsValue (Text)
  - Number of Motorists
- lblNonInjuriesValue (Text)
  - Number of Non-injured victims
- lblNoVehiclesValue (Text)
  - Number of vehicles
- lblOtherInjuriesValue (Text)
  - Number of victims with other injuries
- lblPassengersValue (Text)
  - Number of passengers
- lblPassengerVehiclesValue (Text)
  - Number of Passenger vehicles
- lblPedestriansValue (Text)
  - Number of pedestrians
- lblPillionsValue (Text)
  - Number of Pillions
- lblPublicVehiclesValue (Text)
  - Number of Public Vehicles

- lblSeriousInjuriesValue (Text)
    - Number of victims with serious injuries
- lblTotalPeopleValue (Text)
    - Number of people
- lblUnknownsValue (Text)
    - Number of unknowns
- lblUnlicensedValue (Text)
    - Number of unlicensed drivers
- lblYoungDriversValue (Text)
    - Number of young drivers
- recordInformation (Search Result Information)
    - The record to display in the window

- Events
    - ConfigureWindow
        - Using its recordInformation input, it sets the recordInformation variable to store the information of the record to be displayed in the winodow and then sets all of the label values in the window to their corresponding details and statistics
    - OnPressed (btnClose)
        - Unlocks mouse interactions by setting the isMenuOpen flag to false and then removes the window from the screen
    - OnPressed (btnEditRecord)
        - Creates a NewAccidentRecordForm and configures it with the information displayed in the AccidentDetailsWindow. The AccidentDetailsWindow is then removed from the screen
    - Event Construct
        - Disables mouse interactions by setting the isMenuOpen flag to true
- Procedures
    - SetLabelValues
        - Gets all of the label values and sets them to their corresponding record properties from the recordInfo input

CrashInterface

- Description
    - The central user interface widget that gives the user control over the program
- Variables
    - brdFileMenu (Border)
        - Container that contains the btnFile button
    - btnFile (Button)
        - Opens the brdFileMenu container
    - btnQuit (Button)
        - Quits Crash Site 3D
    - btnSwitchView (Button)
        - Switches Camera's View mode
    - SearchPanel (SearchPanel)
        - Contains the search features used for retrieving record from the database
    - ToolsSelectionMenu (ToolsSelectionMenu)
        - Lets the user select between navigating the map and plotting a new record

    - NewAccidentRecordForm (NewAccidentRecordForm)
        - Stores the current NewAccidentRecordForm if one is currently on screen
- Functions
    - *CreateAccidentDetailsWindow*
        - Inputs
            - recordInformation (Search Result Information)
                - The record to display in the AccidentDetailsWindow
        - Description
            - Displays and configures a new AccidentDetailsWindow
    - *CreateNewAccidentRecordForm*
        - Inputs
            - recordInformation (Search Result Information)
                - The record to display in the NewAccidentRecordForm
            - isNewRecord (Boolean)
                - Is the record we are displaying a newly plotted record or an existing record?
        - Description
            - Displays and configures a new NewAccidentRecordForm as well as stores itself in the NewAccidentRecordForm variable
    - *SetNewAccidentRecordFormVisibility*
        - Inputs
            - recordInformation (Search Result Information)
                - The new record information to display in the NewAccidentRecordForm
        - Description
            - Makes visible and reconfigures a new NewAccidentRecordForm with the new information
    - *SetSwitchViewButtonText*
        - Outputs
            - returnValue (Text)
                - The new text to display on the btnSwitchView button
        - Description
            - Sets the btnSwitchView button text to either 'View Full Map' or 'Go To Street Level' based on whether or not the user is in full map mode or not
    - *CreateDialogBox*
        - Inputs
            - message (Text)
                - The message to display in the dialog box
        - Description
            - Displays a dialog box on the screen to prompt the user with the message from the message input

        - *TrackNumberOfSearchResults*
            - Outputs
                - returnValue (Text)
                    - The number of records being displayed out of the number of search results
            - Description
                - Controls the text displayed in the top-right corner of the screen that tells the user how many accident markers are being displayed around the camera out of the number of search results
                - To do this, it uses the length of the renderedAccidentWidgets map as the number of markers being displayed
                - It then adds it to the length of the cachedAccidentWidgets map. From this, we get the total number of search results

- Events
    - OnPressed (btnFile)
        - Toggles the visibility of the brdFileMenu container
    - OnClicked (btnQuit)
        - Quits Crash Site 3D
    - OnPressed (btnSwitchView)
        - Toggles between the street level and full map view mode on the camera

DialogBox

- Description
    - This is used when a message needs to be conveyed to the user
- Variables
    - btnOK (Button)
        - Closes the dialog box
    - lblMessage (Text)
        - The message to display
- Events
    - ShowDialogBox
        - Sets the lblMessage to display the message from its message input and sets the isMenuOpen flag to true to disable mouse interaction.
    - OnPressed (btnOK)
    - Sets the isMenuOpen flag to false to enable mouse interaction and then closes the dialog box

GeocoordinatesDisplay

- Description
    - Used for displaying the geocoordinates of where the mouse is currently pointed at
- Variables
    - lblLatitudeValue (Text)
        - Latitude geocoordinate value
    - lblLongitudeValue (Text)
        - Longitude geocoordinate value
    - searchResultInformation (Search Result Information)

        - The record that is currently being edited is stored here so that the new coordinates can be applied to it and passed back to the NewAccidentRecordForm
- Events
    - Event Construct
        - Sets the current tool selection to SetNewAccidentLocation
    - Event Tick
        - Keeps track of where the mouse is pointed to by converting the mouseLocationInWorldSpace vector to geocoordinates and displaying it
    - SetAccidentCoordinates
        - Sets the isMenuOpen flag to true to disable mouse interaction
        - If the cancelledAction input is true, the NewAccidentRecordForm will be displayed with no changes made to the record.
        - If the cancelledAction input is false, the NewAccidentRecordForm is displayed again but this time, with the new geocoordinates. The record and marker is updated takes the camera to the location of the new coordinates. Finally the geocoordinates display is removed from the screen.
        - If debugging mode is enabled, the converted vector coordinates of the newly chosen geocoordinates will be displayed on screen.
- Procedures
    - Cancel Set New Coordinates
        - Displays the NewAccidentRecordForm again with the record unchanged. The geocoordinates display is then removed from the screen
    - Reconstruct Record
        - Returns the record but with the new geocoordinates converted from the mouseLocationInWorldSpace vector

NewAccidentRecordForm

- Description
    - A variant of the AccidentDetailsWindow. This variant allows for the editing and creation of accident records.
- Variables
    - btnCloseWithoutSaving (Button)
        - Closes the form without saving the record
    - btnSaveRecord (Button)
        - Saves the record, then closes the form
    - btnSetNewCoordinates (Button)
        - Creates a GeocoordinatesDisplay, enables mouse interactions and then hides the form
    - cmbxAccidentDateValue (ComboBox)
        - A list of dates that the Accident Date value can be set to.
        - Includes numbers from 1 to 31
    - cmbxAccidentDayValue (ComboBox)
        - A list of days that the Accident Day value can be set to.
        - Includes days from Monday to Sunday as well as Unknown if the day is not present in the record
    - cmbxAccidentHourValue (ComboBox)
        - A list of hour values that the Accident Date value can be set to.

    - Includes numbers from 00 to 23
- cmbxAccidentMinuteVaue (ComboBox)
    - A list of minute values that the Accident Minute value can be set to.
    - Includes numbers from 00 to 59
- cmbxAccidentMonthValue (ComboBox)
    - A list of month values that the Accident Month value can be set to.
    - Includes numbers from 01 to 12
- cmbxAccidentSecondValue (ComboBox)
    - A list of second values that the Accident Second value can be set to.
    - Includes numbers from 0 to 59
- cmbxAccidentYearValue (ComboBox)
    - A list of year values that the Accident Year value can be set to.
    - Includes numbers from 2013 to 2050
    - Controls the lblAccidentNumberSevenDigits and lblAccidentNumberYear values
- cmbxAlcoholRelatedValue (ComboBox)
    - Can be set to yes or no depending on if the accident was alcohol related
- lblAccidentNumberSevenDigits (Text)
    - The last seven digits of the accident number
- lblAccidentNumberYear (Text)
    - The year digits of the accident number
- lblLatitudeValue (Text)
    - Value of the latitude geocoordinate
- lblLongitudeValue (Text)
    - Vale of the longitude geocoordinate
- statBicyclists (StatSetter)
    - Number of bicyclists
- statDrivers (StatSetter)
    - Number of drivers
- statElderlyDrivers (StatSetter)
    - Number of elderly drivers
- statElderlyPedestrians (StatSetter)
    - Number of elderly pedestrians
- statFatalities (StatSetter)
    - Number of fatalities
- statFemales (StatSetter)
    - Number of females
- statHeavyVehicles (StatSetter)
    - Number of heavy vehicles
- statInjuredFatal (StatSetter)
    - Number of injured victims and victims who have died
- statMales (StatSetter)
    - Number of males
- statMotorcycles (StatSetter)
    - Number of motorcycles
- statMotorists (StatSetter)
    - Number of motorists
- statNonInjuries (StatSetter)

        - Number of victims without injuries
    - statNoVehicles (StatSetter)
        - Number of vehicles
    - statOtherVehicles (StatSetter)
        - Number of other vehicles
    - statPassengers (StatSetter)
        - Number of passengers
    - statPassengerVehicles (StatSetter)
        - Number of passenger vehicles
    - statPedestrians (StatSetter)
        - Number of pedestrians
    - statPedestrians1318 (StatSetter)
        - Number of pedestrians aged between 13 and 18
    - statPedestrians512 (StatSetter)
        - Number of pedestrians aged between 5 and 12
    - statPillions (StatSetter)
        - Number of pillions
    - statPublicVehicles (StatSetter)
        - Number of public vehicles
    - statSeriousInjuries (StatSetter)
        - Number of victims with serious injuries
    - statTotalPeople (StatSetter)
        - Number of people
    - statUnknowns (StatSetter)
        - Number of unknowns
    - statUnlicensed (StatSetter)
        - Number of unlicensed drivers
    - statYoungDrivers (StatSetter)
        - Number of young drivers
    - recordInformation (Search Result Information)
        - The record being edited
    - isNewRecord (Boolean)
        - Whether or not the record being edited is a new one or an existing one
- Functions
    - *AccidentYearValue*
        - Outputs
            - returnValue (Text)
                - The currently selected year value of the cmbxAccidentYearValue combobox
        - Description
            - Gets the current selection of the accident year combobox and sets the lblAccidentNumberYear value to it
    - *SaveRecord*
        - Outputs
            - record (Search Result Information)
                - The record with its updated information

        - Description
            - Gets all the record property values in the form and constructs a record out of it. This then updates the recordInformation variable with the new information.
    - *UpdateDatabaseAndMarker*
        - Inputs
            - record (Search Result Information)
                - The record to update the database, marker and widget with
        - Description
            - Updates the database as well as the marker and widget with the new information of this record.
- Macros
    - *SingleDigitRemoveZero*
        - Outputs
            - string (String)
                - Returns the number with or without the missing zero
        - Description
            - This macro either does nothing to or adds the missing zero digit onto the beginning of a number that gets removed when converting from an string to an integer (only works for single-digit numbers)
    - *CheckNullDay*
        - Inputs
            - string (String)
                - Value of day to check
        - Outputs
            - option (String)
                - Returns either the day value or unknown if the string was blank
        - Description
            - This macro checks if the day value in the record is blank or not. If it is, then it means the day of which the accident took place is unknown
- Events
    - ConfigureWindow
        - Stores the record information from the recordInformation input in the recordInformation variable
        - If a new record is being created, the CreateNewRecord procedure is executed
        - If an existing record is being edited, the EditExistingRecord procedure will be executed
    - OnPressed (btnCloseWithoutSaving)
        - Closes the form without saving the record
    - OnPressed (btnSaveRecord)

        - Saves the record to the database, updates the marker and widget*, then closes the form
    - OnPressed (btnSetNewCoordinates)
        - Allows the user to pick a new location for the record using the GeocoordinatesDisplay
    - Event Construct
        - Disables mouse interactions
    - On Selection Changed (cmbxAccidentNumberYearValue)
        - Determines a new accident number based on what year value the user selects
- Procedures
    - Create New Record
        - Configures the window so that it displays the initialised values of all the details and statistics
    - Edit Existing Record
        - Configures the window to reflect all of the details and statistics in the record information given

SearchPanel

- Description
    - The widget used to conduct searches on the database. The user can use this widget to search by accident number or by using advanced filters
- Variables
    - btnApplyFiltersAndSearch
        - Run a search using advanced filters
    - btnResetFilters
        - Reset the filters menu to initialised states and values
    - btnSearch
        - Run a search by accident number
    - chkboxFilterName (Checkbox)
        - These checkboxes toggle the state of the filter itself. Determines whether we run a search with that filter or not
    - txtboxFilterNameValue (Text Box)
        - These textboxes hold the values that the user types as the criteria for that record property.
    - isShowingAdvancedFilters (Boolean)
        - Whether or not the advanced filters display is open
    - filterToggles (Array of Booleans)
        - Array of all of the states of all of the advanced filters
    - filterTextboxInputs (Array of Textboxes)
        - Array of all of the values from all of the filters
    - filterCheckboxes (Array of Checkboxes)
        - Array of all the checkboxes that hold the states of the filters
- Events
    - OnClicked (btnShowHideArrow)
        - Toggles the display of the advanced filters
    - OnClicked (btnSearch)
        - Run a search by accident number

    - OnClicked (btnApplyFiltersAndSearch)
        - Determines which filters are enabled, then creates a record with the property values as the specified criterion
        - The program then determines whether any filters have been switched on. If the program is in debugging mode and no filters are enabled, it will display all of the accidents in the database. Otherwise, if debugging mode is not enabled, a dialog box will be shown to the user telling them to tick at least one filter
        - Runs a search using advanced filters with this record and the filter states specified in the filterToggles array
    - OnClicked (btnResetFilters)
        - Reset the filters menu to initialised states and values
    - Event On Initialised
        - Runs the PopulateUserInputsArrays procedure
- Procedures
    - Show/Hide Search Menu
        - Plays the Show Advanced Filters animation forwards or backwards depending on whether or not the advanced filters display is on screen
    - Set Filter Toggles
        - Rolls the states of all of the filter checkboxes into one array and sets it as the filterToggles array.
        - The first 6 are intentionally set to false because they correspond to the accident date, time, day, number and location properties- properties that right now cannot have searches run on them
    - Assign Filter Values
        - Creates a record using the criteria specified in the advanced filters display. Values that are blank or initialised are values not needed for the search
    - Populate User Input Arrays
        - Populates the filterTextboxToggles and filterCheckboxes arrays with the filter checkboxes and their textboxes so that they can be reset later if need be
    - Filters Toggles All False
        - A version of the Set Filter Toggles procedures that sets all of the filter toggle states to false. This is used to mimic all the filters being disabled in order to determine whether the user has enabled any or not.

StatSetterSmall/Medium/Large

- Description
    - This is a widget used extensively throughout the NewAccidentRecordForm. It is used for incrementing and decrementing an accident statistic on the form whilst making sure the value stays above zero.
    - The StatSetter comes in three sizes: small, medium and large. This only reflects the font size of txtStat value.
- Variables
    - btnAdd (Button)
        - Increments the txtStat value by 1
    - btnSubtract (Button)
        - Decrements the txtStat value by 1 if it is greater than 0

    - txtStat (Text)
        - The value of the accident statistic
- Events
    - OnPressed (btnAdd)
        - Increments the txtStat value by 1
    - OnPressed (btnSubtract)
        - Decrements the txtStat value by 1 if it is greater than 0

ToolSelectionMenu

- Description
    - Lets the user select the tool that they want to use. They can click to go to different locations on the map or click to plot a new record. There is a third hidden tool that is only used when selecting a new location for an accident record
- Variables
    - btnCancel (Button)
        - Cancels the new geocoordinates selection action for the record being edited and reopens the NewAccidentRecordForm.
    - btnGoToLocation (Button)
        - Sets the current tool selection to Go To Location
    - btnPlotNewRecord (Button)
        - Sets the current tool selection to Plot New Record
- Events
    - OnPressed (btnCancel)
        - Cancels the new geocoordinates selection action for the record being edited and reopens the NewAccidentRecordForm. The current selection is then reverted back to Go To Location to avoid the mistake of the user plotting a new record by accident if they wanted to navigate the map
    - OnPressed (btnGoToLocation)
        - Sets the current tool selection to Go To Location
    - OnPressed (btnPlotNewRecord)
        - Sets the current tool selection to Plot New Record
    - Event Tick
        - Keeps track of which buttons to have visible in the tools selection menu.
        - Essentially, the cancel button should be the only button visible when new geocoordinates for an accident record are being set. Otherwise, it is the Go To Location and Plot New Record buttons that are visible

Program Testing

Item Tested	Test Data	Expected Result	Actual Results
SearchByAccidentNumber	**accidentNumber:** T20130016071	The record with Accident Number 'T20130016071' appears on the map	The record with Accident Number 'T20130016071' appears on the map
GetColumnValue	**row:** T20150027179 **column:** No. of Vehicles	Gets value of No. of Vehicles in record **outColumn:** 1	Gets value of No. of Vehicles in record **outColumn:** 1
DetermineNewAccidentNumber	**year:** 2015	T20150027180	T20150027180
MapTransfer	**map1:** cachedAccidentWidgets **map2:** renderedAccidentWidgets **itemToMove:** 56	cachedAccidentWidgets does not contain with key '56' renderedAccidentWidgets contains item with key '56'	cachedAccidentWidgets does not contain with key '56' renderedAccidentWidgets contains item with key '56'

Usability Testing

Product Test Description

We are testing to see that the program can run properly when carrying out its intended tasks.

Test Objectives

1. To determine whether the program can correctly recall records, edit them and plot new ones.
2. To gauge how easy the controls are to use.

Business Case

We are testing the program to ensure that when in use by the end-user, data in regards to the car accident records are kept as accurate as possible. Failure to do so may result in inaccurate data when referencing accident records in the future.

Equipment

- Computer
- Keyboard & Mouse
- Crash Site 3D
- Test Table

Test Procedure

1. Look up record T20130016234 using the search bar and record its number of females.
2. Look up record T20140014677 and change its number of passengers and its location. Record the number of passengers and the new location coordinates.
3. Plot a new record anywhere on the map and set the following details:
    - Date: 27/09/2016
    - Time: 22:40:32
    - Day: Tuesday
    - Total People: 7
    - Males: 3
    - Females: 4
    - Drivers: 2
    - No. of Vehicles: 3
4. Record the accident number of the newly created record
5. Search up records where the total number of people involved is 4. Record the number of search results.
6. Using the advanced filters, look up accidents where the criterion are set to the details in the 3rd question. Record its accident number.
7. Look up record T20140014677 again and record its number of passengers and its location.

Test Table

Question	*Expected Value*	*Actual Value*	*Pass/Fail*
Number of Females:	2	2	Pass
Question 2 Number of Passengers:	4	4	Pass
Question 2 New Location Coordinates	-37.806644, 144.918884	-37.806644, 144.918884	Pass
Question 4 Accident Number:	T20160028122	T20160028122	Pass

Question 5 Number of Search Results:	240	240	Pass
Question 6 Accident Number:	T20160028122	T20160028122	Pass
Question 7 Number of Passengers:	4	4	Pass
Question 7 Location Coordinates:	-37.801064, 144.911697	T20160028122	Pass