Due to lack of proper handling variant missions in kam remake I invented the way to store map as text and restore it during mission start.

Example is taken from mission 5 of my campaign On Foreign Lands - Empire Reborn.

Map is generated dynamically based on player decision in mission 4.

To store such a decision we need to use CampaignData built-in record, so in the first place we should create such a script in campaign main folder and named it campaigndata.script. This script should contains structure we need. In my case it will be:

```
record
    //In mission 4 we set number of AI player which escaped. We can call this structure from other missions
    Mission4: record
        VillageEscaped: integer;
    end;
end;
```

Next step will be create a mission script and divide it into variants using if statements and retrieve this variant from campaigndata script:

```
//variant should be declared as global variable because we can call it from different functions inside the script
var variant: integer;

procedure OnMissionStart;
begin
    //retrieving variant from campaigndata file
    variant := CampaignData.Mission4.VillageEscaped;

    if (variant = 1) then begin
        //GENERATE MAP FOR VARIANT = 1
    end
    else if (variant = 2) then begin
        //GENERATE MAP FOR VARIANT = 2
    end
    else if (variant = 3) then begin
        //GENERATE MAP FOR VARIANT = 3
    end;
end;
```

Now, we need to parse maps we would like to include in our variant mission. To parse that map we can use script "mapper.script" you can find in current folder.

NOTE: Maps which will be merged should satisfy following conditions:

1) Human player should be numbered as 1 in game (0 in script)
2) AI players should be numbered from 2 - X (1 - (X-1)) in script
3) AI defence positions (**for attack**) are parsed based on one-soldier-group locations because there is no script to get AI defence coordinates from script, so each defence position should be marked as single soldier. Direction, group type etc. will be created based on this single soldier configuration.
4) AI defence positions (**for defence**) are parsed based on many-soldier-group locations because there is no script to get AI defence coordinates from script, so each defence position should be marked as group with minimum 2 soldiers in it. Direction, group type etc. will be created based on this group configuration.
5) Each parsed map must have the same bounds
6) Dynamic map must have the same bounds as maps which will be merged

To parse map we need to copy "mapper.script" into specified mission main folder and adjust name of file to mission name. Then we need to launch this mission and wait for script log everything :)

Now we need to clean data a little bit:

1) open log file in notepad++ and remove all lines logged as errors/warnings f.e. "Invalid parameter(s) passed to States.PlayerIsAI: 4"
2) copy contents of log file into excel and remove datetime from each line (we can use splitting text into columns function)
3) remove datetime columns
4) split script into map generation part and others part
5) copy map generation part into notepad++ and use advanced replacing to swap ";\r\n" phrase into ','. Then add square bracket with quotation mark and the start of text, and replace "'" phrase at the end with square bracket. Our code should seems like this:

   ['1,1,13,0,33,255','1,2,85,2,30,255', (.....), '27,96,188,3,28,255','27,97,0,0,30,255']

   When it looks like this we can add it as parameter into MapTilesArraySetS function in our merged mission. Other part of script which are responsible for adding houses/units etc. we should copy **after** MapTilesArraySetS function is called.

```
//variant should be declared as global variable because we can call it from different functions inside the script
var variant: integer;

procedure OnMissionStart;
begin
    //retrieving variant from campaigndata file
    variant := CampaignData.Mission4.VillageEscaped;

    if (variant = 1) then begin
        Actions.MapTilesArraySetS(['1,1,13,0,33,255','1,2,85,2,30,255', (.....), '27,96,188,3,28,255','27,97,0,0,30,255'], false, false);

        Actions.HouseAddWaresTo(States.HouseAt(48,20),8,9000);
        Actions.AIStartPosition(4,48,20);
        Actions.GiveHouse(4,0,39,18);
        //....FURTHER CODE.....
    end
    else if (variant = 2) then begin
        //GENERATE MAP FOR VARIANT = 2
    end
    else if (variant = 3) then begin
        //GENERATE MAP FOR VARIANT = 3
    end;
end;
```

We need to do the same for all missions we want to merge.

When all maps are parsed and merged together there is some additional steps we need to do:

1) Each player from parsed maps should have appropriate color,
2) We should declare attacks and group configuration only for each AI player because rest of it configuration is handled by script
3) We should declare where mission starts. Each of your parsed missions can start at different location so there is no easy way to set this in MapEditor. Se we need to use CinematicStart, CinematicPanTo and CinematicEnd procedures to achieve that.

In OnMissionStart we need to start cinematic mode for human player:

```
//variant should be declared as global variable because we can call it from different functions inside the script
var variant: integer;

procedure OnMissionStart;
begin
    //retrieving variant from campaigndata file
    variant := CampaignData.Mission4.VillageEscaped;

    if (variant = 1) then begin
        Actions.MapTilesArraySetS(['1,1,13,0,33,255','1,2,85,2,30,255', (.....), '27,96,188,3,28,255','27,97,0,0,30,255'], false, false);

        Actions.HouseAddWaresTo(States.HouseAt(48,20),8,9000);
        Actions.AIStartPosition(4,48,20);
        Actions.GiveHouse(4,0,39,18);
        //....FURTHER CODE.....
    end
    else if (variant = 2) then begin
        //GENERATE MAP FOR VARIANT = 2
    end
    else if (variant = 3) then begin
        //GENERATE MAP FOR VARIANT = 3
    end;

    //starting cinematic mode for human player
    Actions.CinematicStart(0);
end;
```

Then we need to handle coordinates in OnTick function:

```
procedure OnTick;
begin
    //boolean global variable which tells us that screen was centered already (we need to call this function once)
    if (screenCentered = false) then begin
        if (variant = 3) then begin
            //setting screen center into specified coordinates for variant = 3
            Actions.CinematicPanTo(0, 48, 109, 0);
        end
        else if (variant = 2) then begin
            //setting screen center into specified coordinates for variant = 2
            Actions.CinematicPanTo(0, 11, 92, 0);
        end
        else begin
            //setting screen center into specified coordinates for remaining variants
            Actions.CinematicPanTo(0, 106, 78, 0);
        end;

        //we need to mark that screen was centered to avoid calling this function in future
        screenCentered := true;

        //closing cinematic for user
        Actions.CinematicEnd(0);
    end;
end;
```

If everything is done, we can start our mission and enjoy! :)

NOTE: CampaignData record is not accessible from single missions so you need replace them with some values during testing map in single map mode.

Of course our script works for now, bot looks very ugly. Since 8000+ we can use Pascal include statement **{$I maputils.script}** so we can move all our map generating code into maputils.script and then include it in main script.

In current folder you can find all maps that was parsed and merged into final version of mission 5 campaign On Foreign Lands – Empire Reborn.

I hope you will create something original based on my tutorial!

-Grayter