MarTac Combat Camera Behaviour Spec

Summary

As a 3D game we have the opportunity to use the camera to enhance gameplay and storytelling. Also, because the player has more control over the game’s camera focus, direction and zoom we will need some rules for when we take control back for specific instances.

Flyovers

When a mission begins we’ll have a short introductory sequence that shows the player the combat environment, the enemy placement and the starting location of his Heroes. This will be a simple three-point transition with basic camera settings per-point (focus point, zoom, etc.) starting with an overview, moving to the enemy, then finally player side.

At the enemy portion of the flyover, **one** enemy character facing the player’s team will be randomly selected to play a “intro” animation that will highlight that character and add flavour to the start of the encounter. After the animation is complete the camera will pan back to the player’s side of the board.



Focus Shift

Certain actions will require that the camera react or behave autonomously. Mostly these have to do with targeting and character selection. Common instances include:

**Character Movement**

When a character is moving, either player or AI controlled, the camera should keep them centred and move smoothly with them. In the player’s movement phase the camera should be able to disconnect from the main character, allowing the player to scroll around the map easily. When movement is confirmed, the camera will snap back to focus on the moving unit.

**Target Selection**

When the player is switching between targets or locations for an attack, the camera should centre on whatever selection they make as they click on them (or use the UI to switch between available targets)

**In-play Unit Switching**

When a player is in the movement phase, the camera should keep the current in-play character centred on screen. This includes while they are moving, and after the move is completed. However, when the character has finished their move and action turn and a new character becomes active through the timeline, the camera should quickly track to the next in-play character.

**New Character Arrivals**

When new characters are spawned on the map, we need to move the camera to focus on them along with any story/narrative explanation before returning the view to focus on the current in-play unit.

**Events**

If there is a scripted event as part of the mission that does not have a character associated with it, we should - either through discreet scripting or by the camera being called by an event flag - allow for the camera to focus on it. For instance, if an objective is added part way through a mission or a reactor is about to go critical, the camera should be able to move and frame the reactor object in question to highlight the event.

Action Resolution: Single Target

When one or more characters are engaged in conflict, the camera should move and compose the scene based on the sequences of the action phase, as well as the type, duration and secondary effects of the action itself. Primarily we have two kinds of resolution; **close** and **ranged**. the definition of what’s considered “close” for the camera AI should be a hard limit of grid squares in view, regardless of canvas size or resolution.

**Close actions** should result in the camera focusing on the “actioner” during action select and then attempting to centre both Actioner and Actionee in the same shot when the action is executed. This avoids the need for unnecessary cuts, showing the full progression of the action and its results on the target(s) in one scene.

**Ranged actions**, where both characters cannot fit in the same camera view without aggressive zooming out should result in the camera focusing on the “actioner” during movement action select and then jumping to centre the Actionee when selecting targets as normal. However, instead of trying to frame both units at once, the camera should cut back to the Actioner to show initial action animations/effects then jump cut to the target to show the effects of the action before settling back on the next in-play character.



Action Resolution: Multiple Targets

**Close group** multi-target actions should be handled like a ranged action, in that the camera should seek to contain the targets in a jump-cut after the action has been fired off, ignoring the Actioner in favour of showing the targets during the action-effect phase.

**Scattered groups:** If the targets are scattered over a wide area, the camera should jump between each character in sequence, pausing long enough to see the reaction animation, damage pop-ups and resulting Status effects



KO Cam

Occasionally, a knock out or attack will be so spectacularly awesome that we’ll need to use a special camera for it. the **KO cam** will bring a superheroic feel and allow us to focus on the larger-than-life characters of the game.

KO Cam is triggered automatically based on combat rules (percentage damage taken, remaining enemies, use of certain abilities, etc.) after which the following will happen:

* Camera dollies in for a super-tight shot that frames both characters (attacker and defender)
* An overlay or camera filter is applied to enhance the look and feel of the shot. We could use a number of different effects like slow exposure, vignetting, etc.
* Time is dilated to around 50% speed while we play the “connect” animation on the attacker and the reaction/KO animation on the attackee
* The camera should remain tight on the KO’d character as they play through the sequence, even if the attacker leaves frame
* Camera pulls back to normal once the KO’d character is down and normal turn sequence resumes



Designer Controls (stretch)

In addition to automated camera behaviour, we should allow some Designer-controlled camera movement through scripting or pre-sets to kick off desired movements or camera sequences in combat (for example, when a boss character appears in the map). Useful parameters include:

* **Camera target** (focus the camera on a particular character or grid square)
* **Camera zoom** (far, medium, close)
* **Camera move** (change focus to another character or square)
* **Camera speed** (determine the travel time of the camera when switching positions or settings
* **Wait timer** (set a duration on a particular action)
* **Overlay** (choose an overlay or effect)
* **Dialogue** (call a character dialogue sequence or toast)