一场空军战斗会由两方的两支或以上的可指挥战斗单位在某一个特定区域展开。战斗将会持续至某一方被全部摧毁或是脱离战场。

战斗中视野概念依然存在。

基本概念：

1. 战场
2. 操作
3. 武器运作
4. 战场

在一支（或以上）的单位被另一方的传感器点亮，且处于最大武器射程之内时 [如果 敌方处在我方任意一单位视野内&&在我方任意一单位任意武器射程之内]，一场空军战斗将会开始。在战斗开始时，一个以防御方（被点亮且首先进入射程的单位）为圆心，半径为两方中射程最远武器之射程的120%（避免任何一方离开武器射程后战斗直接结束，允许短时间的追击）的圆将会出现在地图上。这个圆所覆盖的面积就是空战的战场。战场的圆心将持续被放置在离攻击方最近且被点亮的一支防御方可指挥单位上，而圆心的半径将会随着战场中射程最远之武器的存在而变化[圆心半径 = 最远武器射程 \* 1.2]。

1. 操作

与在地图模式中rts式的“选中，点击，移动”的操作不同，在空中战斗中操作更接近于街机或是飞行游戏，并有具体的其他按键绑定来操作一些别的功能例如雷达和攻击。

1. 移动

类似*战争雷霆*中的飞行操控，整个中队会自动地试图指向鼠标所在位置并移动。注意转向不是瞬间完成的而是有转向率与滚转率这个概念的。如果对飞行操纵的概念并不了解，可以类比于速度与加速度：飞机拥有一个最大转向率，但它需要逐渐滚转来达到这个最大转向率。在滚转角度逐渐靠近±90度的过程中，转向率也随之逐渐变大。注意这个操纵的逻辑实际上是飞机进行滚转同时向相对驾驶舱上方仰头从而完成转向，因此这个流程应当是：向最大角度进行滚转并同时操纵仰头达成每一细分时刻允许的最大转向率，直到达到最大转向率时停止滚转并继续仰头。在达到指向目标角度的时候立刻停止仰头并滚转回正。

而移动速度则由节流阀输入间接控制，流程是：节流阀直接控制一个施加在单位尾部方向朝机头的力的大小，通过这个力和我们设定的飞机质量计算出向前的加速度，并间接的影响飞机的移动速度。

在单位移动到鼠标位置且鼠标停留在原地时，应当让其环绕此位置移动（不断转圈）直到鼠标移动至其他位置。

有一个类似自由环绕视角的按键能暂时中止鼠标提供移动目标的功能，在按住此按键时，中队的飞行方向为此按键被按下前一刻鼠标与中队的相对位置（会一直向这个方向飞行），直到自由视角按键被松开，目标位置才会重新更新。

有一个与飞机速度方向相反的阻力会持续施加在飞机上。这个阻力与飞机速度成正比，直到飞机达到最小速度时这个阻力才会消失（因为我们没法在2d世界里让飞机的水平速度停下来，那意味着飞机正在下坠，可我们并没有z轴啊！）。此阻力作为节流阀减小时将飞机慢下来的因素。玩家可以打开减速板让这个阻力增大。

1. 传感器操作

所有的战斗机都至少有一个主传感器，一个人眼，和一个告警传感器。主传感器即为飞机的火控雷达，其朝向与飞机朝向相同，并拥有一个扇形的扫描区域。扇形的角度大小和探测距离取决于飞机类型。这个扇形代表的是雷达扫描的最大角度，实际的扫描则由一个不断在扇形两端间来回移动的窄波束完成。当这个窄波束撞到敌方实际位置时，则将敌方单位的最后已知位置更新到碰撞发生的位置（在没有其他传感器同时探测到目标时）。也就是说，不是只要在这个扇形区域里就会持续不断地点亮目标，而是每扫描到一次目标更新一次目标位置。这个位置不是目标的实际位置，只代表雷达最后一次探测到的目标位置。目标的实际位置可以与这个位置不同。在探测到目标之后，可以操作雷达试图对这个目标进行锁定，即放弃全角度的扫描而在一个更窄但始终试图跟随目标的扇形内扫描，这样能大幅提升目标位置的更新速率，也允许了使用这个位置为雷达制导武器提供目标。雷达可以被一个按键主动关闭。

人眼则是一个同样是扇形，探测距离更短，角度大小固定，能持续性点亮目标的次要传感器。只要在人眼扇形的范围内，敌方的已知位置就会显示为敌方的实际位置，直到其不再处于此范围中。人眼永远指向鼠标所在方向，在自由视角被激活时也是如此。

告警传感器则是全角度的被动传感器。对不同单位来说，它的数值变化只有探测距离不同。告警传感器能向单位汇报任何由主动传感器发出的探测波束的方向（如：被火控雷达扫描到，被开机的导弹雷达扫描到）并告知来源类型（是什么飞机探测到了自己，或在被导弹和火控雷达锁定时直接开始警报提醒）。但对被动探测器如人眼和红外制导则无法告警。

1. 武器  
   如果需要使用飞机的传感器引导导弹，那么就需要首先用雷达锁定目标。在锁定目标后发射武器，则武器（本身也是一个飞行器，所以移动逻辑和前面的飞机逻辑完全相同，只不过它是自动的而且没有节流阀大小）会朝雷达提供的目标位置移动，并在一定时间后开启自己的传感器锁定目标，并将追踪目标切换至自己的传感器提供的位置。为了节约运算资源，雷达制导导弹的传感器并不会像飞机一样复杂，它的运作逻辑和人眼类似，只不过会触发告警传感器。

武器也可以在无目标的情况下发射。这时，武器会直线飞行并在一定时间后打开自己的传感器。如果传感器探测到目标，则按照上述正常运作逻辑追踪。如果没有探测到目标，则什么都不会发生，继续向前飞行直到动力耗尽。

红外制导导弹可以按照雷达制导导弹的逻辑被飞机的火控雷达引导，但也可以直接在飞机上启动其传感器锁定目标后发射。红外制导导弹的传感器与雷达制导导弹的传感器运作完全相同，但不会触发告警传感器。

机炮会向飞机机鼻方向开火，在按下开火键的时候持续发射射弹。在雷达锁定目标的情况下，会为机炮提供一个计算过的提前量标识。没有锁定时，则需要玩家自己判断。

1. 武器运作

此部分作为上述部分的补充，如果存在没有描述到的方面，请参考上述部分。

1. 导弹类武器

导弹在玩家按下发射后会启动自己的发动机。这个发动机会像飞机的发动机一样给导弹施加一个向前的力，直到其做功时间（一个有我们设定的值）耗尽。除此之外，导弹的移动逻辑与飞机完全相同。导弹并不直接向探测到的目标位置移动，而是会根据当前速度和目标速度和方向计算一个在目标移动路径上的拦截点并朝这个拦截点移动。只要有传感器给导弹提供目标（来自发射飞机或来自导弹自己的传感器），那么这个拦截点就会持续更新。

一部分红外制导导弹没有这个计算拦截点的功能。

导弹不再拥有目标时只会向前移动。

导弹同样会有一个与速度大小成正比方向相反的阻力为其减速。当导弹减速到最小速度时，视为其不再有威胁并将其从战场中移除。

导弹的命中计算碰撞箱比导弹的实际大小更大，应把这个概念看作导弹爆炸的杀伤范围。只需要这个范围碰撞到了目标，就视为命中。

1. 机炮类武器

运作时，机炮在弹药耗尽前都会以一个固定的射速持续发射射弹。射弹会有一个最大的初速度并随固定速率减小至最小速度后被从战场中移除（实际上，这间接决定了机炮的射程）。射弹必须要直接碰撞到目标才被认为是命中。它没有一个比自己更大的爆炸范围碰撞箱。

注意事项：

1. 玩家同时只能操控一个中队。如果有多支我方中队在场，其他的将会自动交给与敌人相同的电脑逻辑操作。