[19]中华人民共和国专利局

[51]Int.Cl[6]
G06F 17/00

# [12] 发明专利申请公开说明书

[21] 申请号 98107877.X

[43]公开日 1998 年 11 月 25 日

[11] 公开号 CN 1199896A

[22]申请日 98.4.27

[30]优先权

[32]97.4.25 [33]JP[31]123273/97

[71]申请人 任天堂株式会社

地址 日本京都府

[72]发明人 宫本茂 清水隆雄 今村孝矢
森田和明 木原强

[74]专利代理机构 上海专利商标事务所

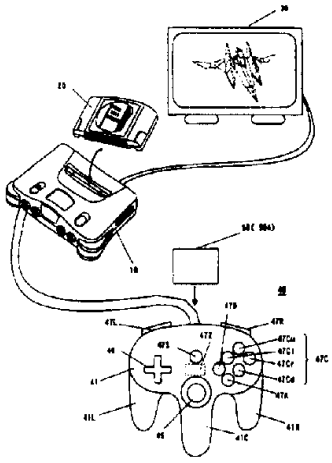
代理人 陈 亮

权利要求书 5 页 说明书 37 页 附图页数 48 页

[54]发明名称 视频游戏系统和视频游戏存储媒体

[57]摘要

一种视频游戏系统，包括视频游戏机和连接到其上的控制器，其中游戏机还与显示装置相连。可分离地连接到游戏机上的视频游戏存储媒体产生显示游戏者目标的图像数据、多个过程以及不是游戏者目标的目标。游戏者通过操作控制器选择一个玩的过程。根据游戏者目标与另一个目标的关系检测给出得分条件。根据该检测，对该过程计数第一次得分。当游戏者目标进入到满足过程清除的条件时，在 RAM 得分寄存区存储第一次得分。如果用户选择玩同一个过程，则把初始值写到该得分寄存区内。



# 权　利　要　求　书

1．一种视频游戏系统，其操作装置由游戏者操作，图像处理设备用于把图像数据提供给显示单元进行显示，并按照程序和根据操作装置的操作状况改变显示图像，其特征在于，

所述操作装置包括：定向装置，用于指定游戏者目标的运动方向，而游戏者可在屏上操纵游戏者目标的运动；和多个运动指定开关，用于指定游戏者目标的运动，所述视频游戏系统包括：

游戏者目标图像数据发生装置，用于产生显示游戏者目标图像的数据；

过程图像数据发生装置，存储有显示不同过程的多个过程数据，游戏者目标可通过这些过程进行，并显示一个根据游戏者对所述操作装置的操作状态而选择的过程；

目标图像数据发生装置，用于对游戏者目标运动的每个过程显示图像，以显示不是游戏者目标的某个目标；

选择屏幕数据发生装置，用于在开始游戏或清除过程时，显示过程选择屏幕，用它可通过操作所述操作装置从诸过程中选择下一次游戏的过程；

给分条件设置装置，用于设置条件对各个过程给出过程中的得分；

给分条件检测装置，用于在游戏者目标正在选择的过程中前进时，根据游戏者目标与其它目标的相互关系检测由所述给分条件设置装置设置的给分条件已经满足；

第一计分装置，用于在游戏者目标正在选择的过程中前进时，根据所述给分条件检测装置的输出对过程计分；

过程清除检测装置，用于检测游戏者目标已满足每个过程的清除条件；

过程得分存储装置，用于以逐次过程为基础存储所述第一计分装置在多次过程中对该次过程计得的分数；

控制装置，用于在所述过程清除检测装置检测出过程清除时，把所述第一计分装置计得的清除过程得分写入清除过程所述过程得分存储装置的相应区域，并且根据进到同一过程的选择，针对立即要清除的过程重新设置所述第一计分装置，以把初始值写入所述过程得分存储装置的相应区域；以及

第二计分装置，用于确定存储在所述过程得分存储装置的逐次过程里的过程得分之和。

2．在一种视频游戏系统中，其操作装置具有方向指定装置，用于指定游戏者目标的运动方向，而游戏者可在屏上操作游戏者目标的运动，多个运动指定开关用于指定游戏者目标的运动，以及游戏机，用于向显示单元提供图像数据，以根据程序和操作装置的操作状况来显示和改变显示图像，其中视频游戏存储媒体存储有所述游戏机处理的游戏程序，所述视频游戏存储媒体包括:

游戏者目标图像数据生成，用于生成显示游戏者目标图像的数据;

过程图像数据生成程序，用于存储显示不同过程使游戏者目标能通过其前进的多个过程数据，并显示一个由游戏者根据所述操作装置的操作状态而选择的过程;

目标图像数据生成程序，用于显示图像，以对游戏者目标前进的每个过程显示不是游戏者目标的目标;

选择屏幕数据生成程序，用于在开始游戏或清除过程时显示一个过程选择屏幕，由此通过操作所述操作装置从诸过程中选择下一次游戏过程;

给分条件设置程序，用于设置对各过程给出过程中得分的条件;

给分条件检测程序，用于在游式者目标正在选择的过程中前进时，根据游戏者目标与其它目标的相互关系检测已满足所述给分条件设置装置所设置的给分条件;

第一计分程序，用于在游戏者目标正在选择的过程中进动时，根据所述给分条件检测装置的输出过程计分;

过程清除检测程序，用于检测游戏者目标已满足每个过程的清除条件;

过程得分存储程序，用于根据逐个过程为原则存储所述第一计分装置在诸过程中对该过程已计得的得分;

控制程序，用于在所述过程清除检测装置检测过程清除时，针对清除的过程把所述第一计分装置计得的清除过程得分写入所述过程得分存储装置的相应区域里，并根据进到同一过程的选择，针对立即要清除的过程重新设置所述第一计分装置，以把初始值写入所述过程得分存储装置的相应区域里；以及

第二计分程序，用于确定存储在所述过程得分存储装置的逐个过程区域里的过程得分之和。

3．一种视频游戏系统，其操作装置由游戏者操作，其图像处理设备用于向显示单元提供图像数据，以根据程序和操作装置的操作状况来显示和改变显示图像，其特征在于，

所述操作装置包括：方向指定装置，用于指定游戏者目标的运动方向，而游戏者可在屏上操作该目标的运动；以及多个运动指定开关，用于指定游戏者目标的运动，所述视频游戏系统包括：

游戏者目标图像根据发生装置，用于产生显示游戏者目标图像的数据，该目标组合了多个由游戏者目标划分的部分；

影响目标图像数据发生装置，用于产生显示影响目标图像的影响目标图像数据，所述影响目标图像显示在游戏者目标的周围并对游戏目标产生影响；

击中确定装置，用于在影响目标对游戏者目标有影响时，对游戏者目标的每个划分部分作击中确定；

图像改变装置，用于改变所述游戏者目标图像数据发生装置产生的图像数据，从而在显示中改变被所述击中确定装置作出中确定的游戏者目标的划分部分；以及

运动状态改变装置，用于根据被所述图像改变装置作出改变的游戏者目标划分部分的位置，改变游戏者目标的运动状态。

4．如权利要求3所述的视频游戏系统，其特征在于游戏者目标由多个划分部分形成，而划分部分进一步由子划分部分形成，

所述图像改变装置改变由所述游戏目标图像数据发生装置产生的图像数据，从而在显示中改变被所述击中确定装置确定为击中的游戏者目标划分部分的子划分部分。

5．如权利要求3所述的视频游戏系统，其特征在于游戏者目标是在平面上观看是左右对称选择的，

所述击中确定装置检测左右部分哪一部分是击中的游戏者目标；

所述图像改变装置对所述击中确定装置确定为击中的一面改变形状；以及

所述运动状态改变装置在所述图像改变装置改变的一面与未改变的一面之间的不同的方式改变运动状态。

6．根据权利要求3所述的游戏系统，其特征在于以飞机形状选择游戏者目标，

所述击中确定装置检测左、右或中部的哪一部分为击中的游戏者目标；

所述图像改变装置改变被所述击中确定装置确定为击中一面的形状；以及

所述运动状态改变装置在所述图像改变装置改变的一面与未改变一面之间以不同的方式改变运动状态。

7．如权利要求 3 所述的视频游戏系统，其特征在于所述方向指定装置是一模拟操纵杆，及

所述运动状态改变装置根据被所述图像改变装置改变的游戏者目标的划分部分位置和所述模拟操纵杆的操作状态改变运动状态。

8．如权利要求 3 所述的视频游戏系统，其特征在于所述运动状态改变装置把运动方向的改变加到已被所述图像改变装置改变的游戏者目标。

9 等等如权利要求 3 所述的视频游戏系统，其特征在于当在影响目标与游戏者目标任一划分部分之间出现影响时，所述击中确定装置确定影响量，且该系统包括一影响量存储装置，用于存储击中部分受影响的程度；以及

所述图像改变装置对所述游戏者目标图像数据发生装置产生的图像数据作出改变，从而当存储在所述影响量存储装置中的影响量变为预定值时，在显示中改变其游戏者目标的相关划分部分。

10．一种存储有视频游戏系统程序的视频游戏存储媒体，其操作装置的方向指定装置用于指定游戏者目标的运动方向；而游戏者可在屏上操作该目标的运动，多个运动定开关用于指定游戏者目标的运动，图像处理设备向显示单元提供图像数据，并根据程序与操作装置的操作显示和改变显示图像，所述游戏机和所述视频游戏存储媒体包括：

游戏者目标图像数据生成程序，用于产生显示游戏者目标图像的数据，而游戏者目标组合有多个由该目标划分的部分；

影响目标图像数据生成程序，用于产生显示影响目标图像的影响目标图像数据，所述影响目标图像显示在游戏者目标周围并对其产生影响；

击中确定程序，用于当影响目标影响游戏者目标时，确定游戏者目标每个划分部分的击发；

图像改变程序，用于改变由所述游戏者目标图像数据发生装置产生的图像数据，从而在显示中改变被所述击中确定装置确定击中的游戏者目标的划分部分；以及

运动状态改变程序，用于根据所述图像改变装置作出改变的游戏者目标划分部分的位置改变游戏者目标的运动状态。

# 说　明　书

## 视频游戏系统和视频游戏存储媒体

本发明涉及视频游戏系统和视频游戏存储媒体，尤其涉及一种诸如在射击游戏中通过选择一条路线或显示移向敌方的游戏者目标而游戏的视频游戏系统和视频游戏存储媒体。

在传统视频游戏中无法在游戏的半途再来一次一度被清除的过程。一度被清除的过程无法重现，除非使游戏机复位而重新再开始游戏。同时，其它的已有技术几乎全是显示移动游戏者目标的视频游戏，在移动的一点上确定击中。

在多次过程中以清除过程总得分来比赛的游戏中，甚至在以高分清除了一次过程时，假如又以低分清除另一次过程，则总分很低。因此，游戏者难以获取高分。同时，在重来一次过程时，游戏机要求复位以让游戏重新开始。即便是不愿意再玩的过程，仍得被迫再玩。因此，游戏者感得负担不希望付出的精力，对游戏失去兴趣，或感到不胜负担。

而且，在采用被构筑成以一点确定击中的游戏者目标的情况下，不能变换花样显示游戏者目标的每个有效部分，难以显示多样化的游戏者目标，从而使图像显示受到限制，不能进一步提高游戏的吸引力。

所以，本发明的主要目的是提供一种视频游戏系统和游戏处理存储媒体。

本发明的另一个目的在于提供一种视频游戏系统和游戏处理存储媒体，它在提供多个过程的情况下，即使在有难度的游戏中也能获取高分，而且清除一个过程以后或者场景难以操纵的情况下进入同一个过程（即重来），让游戏者可赢得满意的高分而不必把游戏中途数据返回清除状态，以提高游戏的兴趣，满足游戏者的成就感。

本发明的再一目的是提供一种视频游戏系统和视频游戏存储媒体，其游戏者目标分成多个被确定击中的部分，显示中可根据击毁点改变游戏者目标，从而作出全面更改，增大兴趣。

本发明的又一个目的是提供一种视频游戏系统和视频游戏存储媒体，它根据必须操纵操作装置的游戏者以不同的方式改变游戏者目标的运动，对游戏者目标

给出逼真的运动，并作出全面更改，增大兴趣。

本发明还有一个目的是提供一种视频游戏系统和视频游戏存储媒体，它根据对每个分区部分的影响量改变游戏者目标的运动，对游戏者给出逼真的运动，并作出全面更改，增大兴趣。

一种根据本发明的视频游戏系统，其操作装置由游戏者操作，图像处理设备用于把图像数据提供给显示单元进行显示，并按照程序和根据操作装置的操作状况改变显示图像，操作装置包括：定向装置，用于指定游戏者目标的运动方向，而游戏者可在屏上操纵游戏者目标的运动；和多个运动指定开关，用于指定游戏者目标的运动，视频游戏系统包括：游戏者目标图像数据发生装置，用于产生显示游戏者目标图像的数据；过程图像数据发生装置，存储有显示不同过程的多个过程数据，游戏者目标可通过这些过程进行，并显示一个根据游戏者对操作装置的操作状态而选择的过程；目标图像数据发生装置，用于对游戏者目标运动的每个过程显示图像，以显示不是游戏者目标的某个目标；选择屏幕数据发生装置，用于在开始游戏或清除过程时，显示过程选择屏幕，用它可通过操作操作装置从诸过程中选择下一次游戏的过程；给分条件设置装置，用于设置条件对各个过程给出过程中的得分；给分条件检测装置，用于在游戏者目标正在选择的过程中前进时，根据游戏者目标与其它目标的相互关系检测由给分条件设置装置设置的给分条件已经满足；第一计分装置，用于在游戏者目标正在选择的过程中前进时，根据给分条件检测装置的输出对过程计分；过程清除检测装置，用于检测游戏者目标已满足每个过程的清除条件；过程得分存储装置，用于以逐次过程为基础存储第一计分装置在多次过程中对该次过程计得的分数；控制装置，用于在过程清除检测装置检测出过程清除时，把第一计分装置计得的清除过程得分写入清除过程过程得分存储装置的相应区域，并且根据进到同一过程的选择，针对立即要清除的过程重新设置所述第一计分装置，以把初始值写入过程得分存储装置的相应区域；以及第二计分装置，用于确定存储在过程得分存储装置的逐次过程里的过程得分之和。

视频游戏机安装有诸如 ROM 等的外部存储媒体。外部存储媒体包含：游戏者目标图像数据生成，用于生成显示游戏者目标图像的数据；过程图像数据生成程序，用于存储显示不同过程使游戏者目标能通过其前进的多个过程数据，并显示一个由游戏者根据操作装置的操作状态而选择的过程；目标图像数据生成程

序，用于显示图像，以对游戏者目标前进的每个过程显示不是游戏者目标的目标；选择屏幕数据生成程序，用于在开始游戏或清除过程时显示一个过程选择屏幕，由此通过操作操作装置从诸过程中选择下一次游戏过程；给分条件设置程序，用于设置对各过程给出过程中得分的条件；给分条件检测程序，用于在游戏者目标正在选择的过程中前进时，根据游戏者目标与其它目标的相互关系检测已满足给分条件设置装置所设置的给分条件；第一计分程序，用于在游戏者目标正在选择的过程中进动时，根据给分条件检测装置的输出过程计分；过程清除检测程序，用于检测游戏者目标已满足每个过程的清除条件；过程得分存储程序，用于根据逐个过程为原则存储所述第一计分装置在诸过程中对该过程已计得的得分；控制程序，用于在过程清除检测装置检测过程清除时，针对清除的过程把所述第一计分装置计得的清除过程得分写入过程得分存储装置的相应区域里，并根据进到同一过程的选择，针对立即要清除的过程重新设置第一计分装置，以把初始值写入过程得分存储装置的相应区域里；以及第二计分程序，用于确定存储在过程得分存储装置的逐个过程区域里的过程得分之和。

存储媒体，例如外部 ROM 事先存储有游戏者目标图像数据发生程序、过程图像数据发生程序和目标图像数据发生程序，所以视频游戏机可以根据这些程序提供的图像数据在显示装置上显示游戏者目标图像和其它目标图像。

视频游戏机包括 CPU 和 RCP(实际协处理器)通过 CPU 和 RCP 在显示装置上显示选择屏，所以游戏者可以在游戏开始或清除一个过程时选择一个过程。根据其选择，游戏者可以选择一个要进入的过程。从而开始游戏。

在玩游戏期间，得分条件检测装置检测相对于游戏者目标对其它目标的关系给分(击中检测)设置是否已完成。如果满足条件，则第一得分计数装置对该过程计数一个得分。如果过程清除检测装置检测到过程清除，则把第一得分计数装置计数的第一得分写入到过程得分存储装置的一个区域内，并把第一得分计数装置复位。

当已经清除铁一个过程再次被游戏者选择时，就把初始值写入到过程得分存储装置的相关区域内。因此，该过程可以重新开始玩，并计数第一得分。第二得分计数装置累加上该第一得分。

根据本发明，视频游戏系统和视频游戏存储媒体可以提供视频游戏的真实感，满足游戏者的成就感，提供视频游戏的兴趣。

此外，在提供多次过程的情况下，即使在有难度的游戏中也能获取高分，甚至在清除某次过程后或场景难以操纵的情况下进入同一个过程（即重来），故游戏者能赢得满意地高分而不必把游戏中途数据返回清除状态，从而提高了游戏的兴趣并满足游戏者的成就感。

在本发明的一个方面，视频游戏系统包含：游戏者目标图像根据发生装置，用于产生显示游戏者目标图像的数据，该目标组合了多个由游戏者目标划分的部分；影响目标图像数据发生装置，用于产生显示影响目标图像的影响目标图像数据，影响目标图像显示在游戏者目标的周围并对游戏者目标产生影响；击中确定装置，用于在影响目标对游戏者目标有影响时，对游戏者目标的每个划分部分作击中确定；图像改变装置，用于改变游戏者目标图像数据发生装置产生的图像数据，从而在显示中改变被击中确定装置作出中确定的游戏者目标的划分部分；以及运动状态改变装置，用于根据被图像改变装置作出改变的游戏者目标划分部分的位置，改变游戏者目标的运动状态。

游戏者目标的每个分区部分还可以进一步由多个子分区部分构成。在这种情况下，图像改变装置以这样一种方式改变游戏者目标图像数据发生装置产生的图像数据，改变击中确定装置已确定的游戏者目标的分区部分的子分区部分。

请注意，游戏者目标通常选择成在平面上是左右对称的。击中确定装置确定游戏者目标的左或右部分被击中。图像改变装置改变击中确定装置确定的被击中一侧的形状。移动状态改变装置以由图像改变装置已改变的一侧与未改变的一侧之间的差异的方式来改变运动状态。

例如，在游戏者目标为飞机形式的形状时，如果假设右翼确定被击中，则图象改变装置改变右翼形状。在这种状态下，如果操纵杆指定游戏者目标移动，则与左翼移动速度相比，移动状态改变装置减慢右翼的移动速度。因此，游戏者目标(飞机)开始以正常形式向右曲线飞行。

在这一方面，如果游戏者目标的分区部分确定被击中，根据受损部分改变了游戏者目标的显示状态，则可以实现动态和有趣的视频游戏。

如果游戏者目标构筑成以多种方式改变运动，则游戏者不得不操纵操作装置对付这种改变，所以提供了一种游戏者目标逼真运动的动态的有趣的视频游戏。

如果游戏者目标根据游戏者目标每个分区部分的影响量来改变运动，则可以提供一种游戏者目标逼真运动的动态和有趣的视频游戏。

通过下面结合附图对本发明的详述，本发明的上述目的与其它目的、特征、各个方面和优点将更清楚.

图 1 是示出本发明一实施例视频游戏系统结构的外观图;

图 2 是本发明一实施例的视频游戏系统的方框图;

图 3 是控制器控制电路 18 的详细电路图;

图 4 是控制器 40 的方框图;

图 5 是示意表示外部 ROM21 整个存储空间的存储图;

图 6 是详细展示外部 ROM21 一部分存储空间的存储图;

图 7 是示意表示 RAM15 整个存储空间的存储图;

图 8 是详细展示 RAM15 一部分存储空间的存储图;

图 9 是表示应用本发明的一例游戏过程的示意图;

图 10 是表示选择图 10 所示游戏场景过程的示意图;

图 11 是表示游戏区域图的视图，用于解释作为应用本发明的一例游戏内容;

图 12 示意表示在图 11 游戏中与同伴通信过程中的信息输出内容;

图 13 表示一例基于图 11 游戏中与同伴通信过程表示的屏上显示器的信息输出;

图 14 表示一例在图 11 游戏中与主角战斗的屏上显示;

图 15 是根据本发明一实施例处理游戏的主流程图;

图 16 是表示过程选择屏上详细处理的子流程图;

图 17 是表示模式转换详细处理的子流程图;

图 18 是说明控制器控制电路 18 与视频游戏机主体之间数据传送的流程图;

图 19 、 20 和 21 是同伴通信处理的子流程图，作为一例辅助游戏进程的信息输出处理;

图 22 和 23 是补充材料供给处理的子流程图，作为另一例辅助游戏进程的信息输出处理;

图 24 、 25 和 26 是同伴目标处理的子流程图;

图 27 是敌方目标处理的子流程图;

图 28 是包含在图 27 的敌方目标处理中的一部分操作步骤;

图 29 是静止目标处理的子流程图;

图 30 是让与处理的子流程图;

图 31 是伴音处理的子流程图;

图 32 是游戏者目标处理的子流程图;

图 33 是游戏者目标处理的子流程图;

图 34 是单向卷动模式下三维空间的 X(-Z)坐标;

图 35 是游戏者目标 60 的外观图;

图 36 是游戏者目标处理的子流程图;

图 37 是全程模式三维空间的 X(-Z)坐标;

图 38 是游戏者目标处理的子流程图;

图 39 是击中确定处理的子流程图;

图 40 是游戏者目标 60 失去部分双翼的外观图;

图 41 是游戏者目标 60 失去部分左翼的外观图;

图 42 是游戏者目标处理的子流程 ;

图 43 是游戏者目标 60 失去大部分左翼的外观图;

图 44 是摄像机处理的子流程图;

图 45 是摄像机处理的子流程图;

图 46 是表示一例屏上显示游戏者目标的示意图;

图 47 是表示另一例屏上显示游戏者目标的示意图;

图 48 是敌方目标击中确定处理的子流程图。

下面说明本发明视频游戏系统的结构和其使用的视频游戏存储媒体。顺便提一下，虽然下述的实施例说明作为专用视频游戏机的情况。但是作为其它应用例子，同样适用于成像处理设备、个人计算机等设备。尽管把操作装置说明为游戏专用控制器，但是只要本发明的视频游戏系统适用于个人计算机一类的图像处理装置，也可采用键盘、鼠标器一类的输入设备。

图 1 是展示本发明一实施例的视频游戏系统结构的外观图。视频游戏系统的结构包括：视频游戏机主体 10， ROM 卡盘 20（作为外部存储器设备）， CRT 显示器 30（作为与主体 10 连接的显示设备），以及作为操作装置（或操作输入装置）的控制器 40。需要时，控制器 40 为可卸安装，配有 RAM 卡盘 50（或振动卡盘 50A ）。

控制器 40 的结构是在壳体 41 上设置多个开关或按钮，而壳体 41 做成可用手抓住的形式。具体说，控制器 40 在壳体 41 的左、右和中央下部装有手柄 41L 、

41C、41R，在其上表面配有操作区。在操作区内，在中央下部装有模拟输入的操纵杆45，在左侧装有十字形数字式方向开关（下称“十字开关”）46，在右侧有多个按钮开关47A－47Z。操纵杆45用于根据操纵杆倾斜的程度和方向指定或输入游戏者目标的移运方向和/或移动速度（或移动量）。十字开关46用于代替操纵杆45作输入，指定游戏者目标的移动方向。多个按钮开关47包括用于指定游戏者目标运动的开关47A与47B、用于切换观看者通过摄像机观察的图像眼的开关47C、起动开关47S、壳体41左上部侧面设置的运动开关47L、壳体41右上部侧面设置的运动开关47R以及手柄41C背面设置的开关47Z。开关47C由排列在上下左右的4只按钮开关47Cu、47Cd、47Cl、47Cr组成，所以除了改变观看者的摄像机眼以外，还可在射击或作战游戏中用来控制移动速度（如加速、减速等）。这类按钮开关47A－47Z可按游戏节目限定功能。

图2是本发明一实施例的视频游戏系统的方框图。视频游戏机10内部配有中央处理单元（下缩写为“CPU”）11和协处理机（实际协处理机：下缩写为“RCP”）12。RCP12包括：用于控制总线的总线控制电路121；用于执行多边形坐标变换、浓淡处理等的图像处理单元（实际信号处理器；下缩写为“RSP”）122；用于把多边形数据光栅扫描成待显示图像并把该图像转换成可在帧存储器上存储的数据形式（点数据）的图像处理单元（实际显示处理机；下缩写为“ROP”）123。把RCP12连接到卡盘连接器13以可卸式安装ROM卡盘20，连接到盘驱动连接器14以可卸苛安装盘驱动器26，并且连接到RAM15。此外，还把RCP12连接到音频信号发生电路16以输出经CPU11处理的伴音信号，并连接到图像信号发生电路17以输出图像信号。另外，RCP12还连接到控制器控制电路18，对一个或多个控制器40A－40D串行传送操作数据，并且/或者为RAM卡盘50传送数据进行扩展。

包含在RCP12中的总线控制电路121对来自CPU11经总线提供的并行信号里的指令作并－串转换，向控制器控制电路18提供串行信号。而且，总线控制电路21把从控制器控制电路18输入的串行信号转换成并行信号经总线向CPU11输出。代表从控制器40A－40D读出的操作状态的数据经CPU11处理后，暂时存入RAM15等里。换言之。RAM15包括一个暂时记忆被CPU11处理的数据的存储区，故被用来通过总线控制电路121平稳地读写数据。

伴音信号发生电路16连接到设置在视频游戏机10背面的连接器195。图像

信号发生电路 17 连接到设置在游戏机 10 背面的连接器 196。连接器 195 能以可断开方式连接到诸如 TV 扬声器等伴音发生器件 32 的连接部分. 连接器 196 能以可断开方式连接到诸如 TV 接收机或 CRT 等显示器 31 的连接部分。

控制器控制电路 18 与设置在视频游戏机 10 正面的控制器连接器（下缩写为“连接器”）191 - 194 相接. 连接器 191 - 194 通过连接插孔以可卸方式连接到控制器 40A - 40D。控制器 40A - 40D 以这种与连接器 191 - 194 相接，使控制器 40A - 40D 与游戏机 10 实现电连接，从而可在两者之间传输/接收或传送数据。

图 3 是控制器控制电路 18 的详细电路图. 控制电路 18 用于在 RCP12 与控制器连接器 191 - 194 之间发送和接收串行数据。控制电路 18 包括数据传送控制电路 181 、发送电路 182 、接收电路 183 和用于暂时记忆传输或接收数据的 RAM184 。数据传送控制电路 181 包括并-串转换电路和串-并转换电路，在数据传送期间转换数据方式，并进一步执行对 RAM184 的读写控制。串 - 并转换电路把 RCP12 提供的串行数据转换成并行数据提供给 RAM184 或发送电路 182 。并 - 串转换电路把 RAM184 或接收电路 183 提供的并行数据转换成串行数据提供给 RCP12 。发送电路 182 把指令（用于控制数据传送控制电路 181 提供的控制器 40 信号读出）与对 RAM 卡盘的写数据（并行数据）转换成串行数据，并提供给分别对应于控制器 40A - 40D 的通道 CH1 - CH4 。接收电路 183 以串行数据方式接收通过相应的通道 CH1 - CH4 输入的控制器 40A - 40D 的操作状态数据和来自 RAM 卡盘 50 的读出数据，把它们转换成并行数据后提供给数据传送控制电路 181 。数据传送控制电路 181 根据从 RCP12 传送的数据、接收电路 183 接收的控制器 40A - 40D 的操作状态数据或者从 RAM 卡盘 50 读出的数据，控制对 RAM184 的写入，并根据来自 RCP12 的指令读出 RAM184 的数据且把它传送给 RCP12 。

RAM184 虽未图示，但它包括存储区 184a - 184h 。区 184a 存储着第一通道的指令，而区 184b 存储着第一通道的传输与接收数据. 同样地，区 184c 存储着第二通道的指令，而区 184d 存储着第二通道的传输与接收数据。区 184e 存储着第三通道的指令，区 184f 存储着第三通道的传输与接收数据。区 184g 存储着第四通道的指令，区 184h 存储着第四通道的传输与接收数据。

图 4 是控制器 40 与 RAM 卡盘 50 的详细电路图. 为检测操纵杆 45 、开关 46

与 47 等的操作状态以把检测的数据传送给控制器控制电路 18，控制器 40 的壳体配备了操作信号处理电路 44 等。该处理电路 44 包括接收电路 441、控制电路 442、开关信号检测电路 443、计数器电路 444、端口控制电路 446、复位电路 447 和或非(NOR)门电路 448。接收电路 441 把诸如控制器控制电路 18 发送的控制信号或对 RAM 卡盘 50 的写数据等串行信号转换成并行信号提供给控制电路 442。当控制器控制电路 18 发生的控制信号是对操纵杆 45 的 X、Y 坐标进行复位的信号时，则控制电路 442 产生一复位信号，通过 NOR 门 448 使计数器 444 内的 X 轴计数器 444X 与 Y 轴计数器 444Y 的计数值复位（O）。

操纵杆 45 包括 X 轴与 Y 轴的光电断续器，以把杠杆倾斜分解为 X 轴与 Y 轴分量而产生在数量上正比于倾斜度的脉冲，从而把脉冲信号分别提供给计数器 444X 和 444Y。当操纵杆 45 以 X 轴向倾斜时，计数器 444X 对响应于倾斜量产生的脉冲数计数；当操纵杆 45 以 Y 轴向倾斜时，计数器 444Y 对响应于倾斜量产生的脉冲数计数。因此，由计数器 444X 与 444Y 的计数值确定的 X 轴与 Y 轴的合成矢量就用来确定游戏者目标或主角或光标的移动方向和坐标位置。顺便说一下，当复位信号发生电路 447 因接通电源而提供复位信号或开关信号检测电路 443 因同时按下预定的两只开关而提供复位信号时，计数器 444X 与 444Y 的计数值也被复位。

开关信号检测电路 443 响应于控制电路 442 以恒定周期（如 1/30 秒间隔的 TV 帧周期）提供的开关状态输出指令，读出根据十字开关 46 和开关 47A - 47Z 按压状态变化的信号，然后把该信号提供给控制电路 442。控制电路 442 响应于控制器控制电路 18 发出的操作状态数据读出指令信号，向发送电路 445 提供对开关 47A - 47Z 预定数据格式的操作状态数据和计数器 444X、444Y 的计数值。发送电路 445 把控制电路 442 输出的并行信号转换成串行信号，经转换电路 43 和信号线 42 把它传送给控制器控制电路 18。控制电路 442 经地址总线、数据总线以及端口连接器 449 连接到端口控制电路 446。当把 RAM 卡盘 50 接到端口连接器 449 后，端口控制电路 446 根据来自 CPU11 的指令执行数据输入/输出（或传输/接收）控制。

RAM 卡盘 50 是通过把 RAM51 接到地址总线与数据总线并把 RAM51 接上电池 52 而构成的。RAM51 是一种 RAM，其容量低于可通过地址总线访问的最大存储容量（如 256K 位）的一半。RAM51 存储了游戏有关的备用数据，即使把

RAM 卡盘 50 从端口连接器 449 上取下，也能利用电池 52 的供电保存好备用数据。

顺便提一下，在通过以高度逼真的方式输出图像或伴音代表伴有冲撞、爆炸等碰撞状态的场合，可以使用配备振动发生电路 53 的 RAM 卡盘 50 或者包含振动发生电路 52 而设有 RAM51 的振动卡盘 50A 。

ROM 卡盘 20 是用装在电路板上的外部 ROM21 构成的，把电路板装在其壳体内部。外部 ROM21 存储有图像数据或节目数据以供游戏等作图像处理，还可按要求存储诸如音乐、效果音响一类的伴音数据或信息。

图 5 是示意表示外部 ROM21 整个存储空间的存储图，图 6 是表示外部 ROM21 存储空间细部（图像显示数据区 24 ）的存储图。外部 ROM21 包括多个存储区（下面在其前面有数据类别名时，把“存储区”缩写为“区”），如图 5 所示，有程序区 22 、角色代码区 23 、图像数据区 24 和伴音存储区 25 ，这样就事先固定存储着各种程序。

程序区 22 存储有例如对游戏执行图像处理所需的程序（执行后面提到的图 15 到图 31 所示流程图功能的程序或符合游戏内容的游戏数据等）。具体说，程序区 22 包括以固定方式为 CPU11 事先存储操作程序的存储区 22a - 22p 。主程序区 22a 为图 15 所示以后说明的游戏存储了主路径处理程序。控制垫数据（操作状态）确定程序区 22b 存储的程序用于处理表示控制器 40 操作状态等的数据。写程序区 22c 存有写程序， CPU11 用它使 RCP12 写入的帧存储器和 Z 缓冲器。例如，写程序区 22c 存有一条程序，用于把色度数据作为基于多个可移动目标或以一个背景显示的背景目标的结构数据写入 RAM15 的帧存储区里（图 7 所示的 152 ），写存有一条把深度数据写入 Z 缓冲区（图 7 中的 153 ）的程序。运动程序区 22d 存有控制程序， CPU11 用它使 RCP12 改变可动目标在三维空间中的位置。摄像机控制程序区 22e 存有摄像机控制程序，以此控制在三维空间中拍摄包括游戏者目标或背景目标的可动目标的位置与方向。过程选择程序区 22f 存有图 16 所示在后面介绍的过程选择子程序。模式转换程序区 22g 存有图 17 所示在后面介绍的模式转换子程序。存储区 22g 存储的程序通过在单向卷动显示与全向（全程）卷动之间转换卷动模式来改变卷动的方向与范围。

通信处理程序区 22h 存有图 19 到图 21 所示在后面介绍的通信处理子程序。补充处理程序区 21i 存有图 22 到图 23 所示在后面介绍的补充处理子程序。游戏

者目标程序区 22j 存有对游戏者操纵的目标作显示控制的程序. 同伴目标程序区 22k 存储的程序（见图 24-图 26）对同伴目标作显示控制，该同伴目标协同游戏者目标展开游戏。敌方目标程序区 22*l* 存储的程序（见图 27 和 28）对攻击游戏者目标的敌方目标作显示控制。背景程序区 22m 存有背景产生程序（见图 29），CPU11 用它使 RCP12 产生三维的背景图片（或过程）。伴音处理程序区 22n 存有产生效果音响或音乐、语音等信息的程序（见图 31）。游戏结束处理程序区 22o 存有在游戏结束时要处理的程序，例如在游戏结束前检测游戏结束的状态或保存游戏状态的备用数据。信息处理程序区 22p 存储的子程序用于处理信息（图 19 到图 21 中的通信处理，处理包括提供图 22 和 23 的补充材料），以便通过显示字符或输出伴音来输出信息，有助于适合持有游戏者目标的位置或环境的操作。

文字代码区 23 是一个记忆多类文字代码的区，它存有相应于某个代码的多类文字点数据。储存在文字代码区 23 中的文字代码数据在游戏中用来对游戏者显示性的注释。在本实施例中，该区用来根据持有游戏者目标的周围环境（如位置、障碍类别、敌方目标类别）或游戏者目标所处的形势而显合理操纵方法或正确时序响应方法的文字信息（或短语）.

如图 6 所示，图像数据区 24 包括存储区 24a - 24f。图像数据区 24 存储每个背景目标和/或可动目标的多个多边形的各别坐标数据、结构数据等，还存有用于以预定位置静止显示这些目标或使目标运动的显示控制程序。例如，存储区 24a 存有显示游戏者目标的程序，存储区 24b 存有显示多个同伴目标 1 - 3 的同伴目标程序，存储区 24c 存有显示多个背景（静止）目标 1 - n1 的背景目标程序，存储区 24d 存有显示多个敌方目标 1 - n2 的敌方目标程序，存储区 24e 存有显示主目标的主目标程序，存储区 24f 则存有用于输出例如图 12 所示在后面介绍的短语或信息的数据。

伴音存储区 25 存储诸如短语、效果音响和游戏音乐等伴音数据，以适合于场景的方式输出伴音的上述信息。

顺便提一下，外部存储装置除了 ROM 卡盘 20 外，可采用诸如 CD - ROM 和磁盘等其它存储媒体取代 ROM 卡盘 20。此时，必要的话应设置盘驱动器（记录再现装置）26 以对 CD - ROM 或磁盘等光盘或磁盘型存储媒体读写各种数据（包括程序数据与图像显示数据）。盘驱动器 26 从同 ROM21 相似的以磁学或光

学方法记忆程序数据的磁盘或光盘中读出数据并把数据传送给 RAM15 。

图 7 是示意表示 RAM15 整个存储空间的存储图，而图 8 是表示 RAM15 存储空间详细部分（图像显示数据区 154 ）的存储图. RAM15 包括各种存储区 150 - 159 。例如， RAM15 包括显示清单区 150 、程序区 151 、用于暂存一帧图像数据的帧存储（或图像缓冲存储）区 152 、用于在帧存储区逐点存储深度数据的 Z 缓冲区 153 、图像数据区 154 、伴音存储区 155 、用于记忆控制垫上操作状态数据的区 156 、工作存储区 157 、同伴数据区 158 和寄存器标志区 159 。存储区 151 - 159 是 CPU11 可通过总线控制电路 121 访问或直接由 RCP12 访问的存储空间，所以它们各自具有按游戏规定的任意容量（或存储空间）。程序区 151 、图像数据区 154 和伴音存储区 155 在有数据传送时，暂时存储一部分数据，例如存储在 ROM21 的存储区 22 、 24 、 25 中的一种游戏全部场景（或阶段）游戏程序某一种过程或阶段所需的游戏程序。若以这种方式把某一场景所需的部分程序数据存储在存储区 151 、 154 、 155 中，则 CPU11 的效率就要比每次由 CPU11 直接从 ROM21 中读出来进行处理的效率更高，这样可提高图像处理速度。

具体讲，帧存储区 152 的存储容量对应于显示器 30 的像元（像素或点）数 X 一个像元的色度数据位数，故它对应于显示器 30 的像元存储每点的色度数据。帧存储区 152 根据三维坐标数据以逐点方式暂时存储可用观察者肉眼看到的某一目标的色度数据，而三维坐标数据表示图像数据区 154 中所存储的一组多个多边形、一个或多个静止目标和/或可动目标，准备以图像处理模式显现在一个背景场中。在显示存储在图像数据区 154 中的各种目标时，其中包括可动目标，即游戏者目标、同伴目标、敌方目标、主目标、背景（或静止）目标等，帧存储区 152 还以显示模式逐点暂存色度数据。

Z 缓冲区 153 的存储容量对应于显示器 30 的像元（像素或点）数 X 一个像元的深度数据位数，故它以某种方式逐点存储对应于显示器 30 像元的深度数据。在图像处理模式中， Z 缓冲区 153 根据三维坐标数据暂存观察者肉眼能看见的目标每一点的深度数据，而三维坐标数据表示一组多个多边形、一个或多个静止目标和/或可动目标。在显示模式中， Z 缓冲区 153 还暂存可动和/或静止目标每一点的深度数据。

在为游戏显示的存储的构成每个静止和/或可动目标的多组多边形中，图像数据区 154 把多边形的坐标数据与结构数据存入 ROM21 ，故在图像处理操作之前，

至少把一个过程或阶段的数据从 ROM21 传送到这里。将参照图 8 说明图像数据区 154 中存储数据的详细情况。

存储在 ROM21 存储区里的部分伴音数据（短语、音乐和效果音响数据）传送到伴音存储区 155，该伴音数据作为通过伴音发生装置 32 产生的伴音数据暂存起来。

控制垫数据（操作状态数据）存储区 156 暂时存储操作状态数据，代表从控制器 40 读出的操作状态。

在 CPU11 执行程序期间，工作存储区 157 暂时存储参数等数据。

同伴数据区 158 暂时存储用于显示控制存储在存储区 22k 里的同伴目标的数据。

寄存器标志区 159 包括多个寄存区 159R 和多个标志区 159F。寄存区 159R 包括：寄存器 R1 - R3，载有对主体、左翼和右翼的损坏量；寄存器 R4，载有对同伴的损坏量；寄存器 R5，载有对敌方（主体）的损坏量；寄存器 R6，载有游戏者目标数；寄存器 7，载有游戏者的寿命计数；寄存器 R8，载有在一个场景中显示的敌方目标数；寄存器 R9，载有静止目标数；寄存器 10，载有游戏过程中的得分点；寄存器 R11 - R1n，载有过程 1-n 的得分点；寄存器 20，载有总得分点；以及寄存器 R21，载有最高点。标志区 159F 是存储标志的区，通过它可知道游戏的进展状态，且包括例如：同伴标志 F1；模式标志 F2，用于鉴别显示范围的模式；短语标志 F31 - F3m，用于存储是否要输出的短语 1-m；游戏结束标志 F4，用于鉴别有无接近游戏结束状态的检测结果；以及击中确定标志 F5。

图 9 表示一例应用本发明的游戏过程，图 10 表示图 9 所示游戏的过程选择屏，图 11 表示说明一例应用本发明的游戏内容的游戏区域图，图 12 示意表示与图 11 游戏中同伴通信过程的信息输出内容，图 13 表示一例根据图 12 游戏中与同伴通信过程所代表的屏上信息显示，图 14 表示一例图 11 游戏中同主角开战状态的屏上显示。

下面参照图 9 - 14 说明视频游戏输出有助于游戏进展的信息的基本情况，它构成了本发明的一个特征。虽然视频游戏的游戏内容由存储在 ROM21 中的程序决定，但在本例中将以射击游戏为例。游戏开始时，显示出图 9 所示的过程。图 9 中，在过程显示区 80 下面给出清除过程显示区 81a - 81e 的显示，表示以逐

个过程为基础的清除状态。屏的上部示出过程得分显示区 82 和高分（顶点）显示区 83 。

在游戏的初始阶段，若选择过程 1 ，就如图 12 那样示出图 11 的起点场景。如图 11 所示，把从起点到模式转换点的长距离选作单向卷动模式的显示区。把单向卷动模式显示区的宽度选成与屏幕尺寸同样宽，可显示在显示器 30 的显示屏 31 上，用于从上到下的卷动显示。在单向卷动模式显示区中，依次显示过程中构成背景图片的目标 71a - 71n（见图 12），诸如楼、树、山、路和天空，代表背景或静止目标。在游戏中途的预定点 A - D 处，出现多个敌方目标 72a - 72n 攻击游戏者目标 60 或阻碍游戏者目标的前行。

在单向卷动模式显示区中，把 A 、 B 、 C 、 D 确定为以显示或伴音方式输出信息（或短语）的地点，让游戏者明白合适的操纵方法，即帮助游戏者目标 60 击退敌方目标 72a - 72n 或巧妙地躲避其攻击。如图 12 所示，在显示区 31a 上显示信息。在显示区 31b 上显现发送该信息的同伴的脸。在显示区 31c 中显示游戏得分，而在显示区 31d 上显示寿命（承受损害的总量）。

图 13 显示表示出作为具体例子的信息。在多个信息中，显示区 31a 显示出程序对某地址设立的信息。在本例中，示出这样一种情况，即根据某人或输出字符的类别与场景用不同的方式以伴音与图像输出短语，由同伴提供的信息找到适合与短语出现相关状况的操纵方法。短语 1 - 9 以优先权确定，这在检测以同一时序产生多个短语的条件时，就产生更高优先级的短语。与信息显示相关的是显示出发送该信息的同伴目标 73 的面貌。信息包括控制方法（对指定减速的信息为“用刹车克服”），此时假定游戏目标 60 是一名战士，并包括操纵方法，说明在控制器 40 上应以何种方式操作哪一只开关来获取该控制方法（信息“ C 按钮向下”表示按压按钮 47Cd ；在上下左右排列的 4 只按钮中，最好显示一种不同色彩的下方按钮）。此外，对于信息显示，需要时也可作声响输出（“用刹车克服”）。在起点 C 产生信息“按二次 Z 或 R ”，说明把开关 47Z 或 47R 按压二次。以这种方式，根据敌方目标的形状或运动，地点 A - D 的信息内容是不同的。如果游戏者操纵操纵杆 45 控制游戏者目标 60 的位置与方向，并根据信息输出操作开关 47A - 47Z 中的某一开关，即便在开关数很多且难以或无法执行合适而快速的开关操作的情况下，他也能容易地执行合适的操作。通过迅速执行指示的操作，可以容易地攻击敌人或躲避危险。游戏者只要未被杀死，很容易写进入

前面的场景。

当游戏者目标 60 到达模式转换点时，显示模式转换成全程模式，能以每一方向卷动。在全程模式中，把主角（主目标）置于可显示区的中央，故游戏者目标 60 可围绕主角 74 对其攻击。在相对于主角 74 的上下左右方向，可在短距离（如几十万）内选择游戏者目标 60 能移动的范围。当游戏者目标 60 移动范围的边界时，改变拍摄游戏者目标 60 的摄像机的方向，从而自动改变游戏者目标 60 的移动方向。此时，在显示屏 31 右下方的地图显示区 31c 中显示一幅缩小的地图，游戏者能方便地知道游戏者目标 60 所在的位置。地图包括显示主角 74 、游戏者目标 60 和同伴目标 73 的符号。

图 15 是本发明一实施例视频游戏系统的主流程图。现参照图 9 - 15 说明沿着图 15 的主流程图的本发明的原理。

接通电源， CPU11 在开始时把视频游戏机 10 置成预定的初始状态。例如，CPU11 把存储在 ROM21 程序区里的游戏程序中的起动程序传送给 RAM15 的程序区 151 ，把诸参数设置成其初始值，之后依次执行图 15 流程图的处理。

在每一帧（ 1/30 秒）执行图 15 的流程操作。在清除过程前，执行步骤 1（图中注为“ S ”）、步骤 2 和步骤 3 - 17 ，之后重复操作步骤 3 - 17 。如果游戏结束而过程清除失败，则执行步骤 18 的游戏结束处理。若过程清除成功，则处理由步骤 16 返回步骤 1 。

即，虽然在步骤 1 显示了游戏过程场景和/或过程选择场景，但若在通电后开始游戏，仍显示出图 9 所示的过程场景。顺便提一下，在清除图 9 所示的过程 1 而进到过程 2 并且过程 2 也被清除以后，就显示图 10 所示的过程选择屏。当在过程选择屏上选择过程时，执行图 16 所示的过程选择子程序（操作步骤 101 - 116 ）。然而，在并不构成本发明的主要部分，因而只示出流程图而不作详细的操作说明。

由于过程 1 的游戏是在起动后立即进行的，所以在步骤 2 对过程进行游戏起动处理。例如，寄存区 159R 和标志区 159F 被清除处理（对寄存区 R6 、 R7 置初始值），并把执行过程 1 （或选择的过程）游戏所需的各种数据从 ROM21 中读出传送给 RAM15 的存储区 151 - 155 。

在步骤 3 ，执行模式转换子程序处理。起动游戏后立即把游戏者目标 60 呈现在图 11 中的起点。然而，起点（ Z 坐标 = 0 ）与模式转换点（ Z 坐标 = 几十

万）之间的时间周期依赖于单向卷动模式。因此，在图 17 的步骤 121 确定游戏者目标不出现在全程模式位置上，故在步骤 122 使标志 F2 复位而转换成单向卷动模式，然后处理进到下一步骤 4 。具体操作将在下面参照图 17 说明。

在步骤 4，执行控制器处理。通过检测操纵杆 45 、十字开关 46 和控制器 40 的开关 47A - 47Z 中任何一个是否操作来执行这一处理。读出对操作状态所检测的数据（控制器数据）并写入读出的控制器数据。下面参照图 18 说明具体的操作。

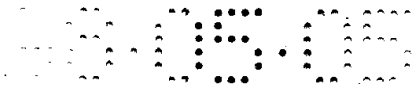
在步骤 5，执行与同伴的通信处理。通过显示或伴音输出告诉合适操纵方法的信息来执行这一处理，这是本发明的一个特征。即，在图 11 所示的地点 A - D ，在单向卷动周期内如图 13 所示作指示或伴音输出的信息或短语，游戏者可知道合适的操纵方法。下面参照图 19 - 21 用一例子来说明具体的操作情况。顺便提一下，要指出，信息的内容或出现条件只是一个举例，具体根据游戏的内容或类别会有所不同，并可根据用途予以修改。

在步骤 6，执行补充处理，从总部提供材料。在这一处理中，即便在游戏者目标 60 被敌人袭击而且机身受损无法正常飞行的情况下，仍由总部或同伴运送帮助游戏者的项目（如修理飞机机翼的零件、火炮、步枪等）。当屏上显示出某一项目时，如果游戏者执行获取该项目的操作（把机身重叠在该项目上，通过射击击中该项目等），则受损部分就能复原，或提供有利于攻击敌人的项目。在这种情况下，由于游戏者需要的项目因游戏者目标受损的情况而不同，故要根据预定的优先级自动确定项目的类别。下面参照图 22 和 23 说明具体操作。

在步骤 7，执行显示游戏者目标 60 的处理。虽然这一处理因游戏者目标 60 存在于单向卷动区或全程而有所区别，但是基本上是一种根据游戏者操作的操纵杆状态和有无敌人攻击来改变方向或形状的处理。例如，通过计算基于存储区 22j 传送程序的改变后多边形数据、存储区 24a 传送的游戏者目标的多边形数据以及操纵杆 45 的操作状态，控制游戏者目标 60 的显示。把色度数据写入对应于多个三角形表面的存储区 154a 的地址，而这些三角形构成了多个这样得到的多边形，从而对其放上由结构数据规定的图案或彩色纸。

在步骤 8，执行摄像机处理。例如，对观看角度进行坐标运算，从而使观看者的肉眼或通过摄像机取景观看的观看位置取某个由游戏者指定的角度。

在步骤 9，执行同伴目标处理。同伴目标是针对在单向卷动区中相对于游戏



者目标的预定位置计算的．例如，当同伴目标在游戏者目标 60 后面飞行时就不予显示，而当游戏者目标 60 减速时，执行计算处理，则把它作为在其前面飞行而作显示。在全程区域内，当同伴目标在游戏者目标 60 前方飞行时，在缩小的地图中用符号一起显示出同伴的机身．当它在后面飞行时，在缩小的地图上只用符号显示。下面参照图 24 到 26 作详细说明。

在步骤 10，执行敌方目标的处理。在这一处理中，通过计算多边形数据显示改变的图像来确定敌方 72a - 72n 的显示位置和/或形状，从而一方面根据部分从存储区 22*l* 与 24d 传送程序判断游戏者目标 60 的运动，另一方面移动攻击游戏者目标 60 或阻碍其行进。据此，敌方目标以某种方式采取行动以对游戏者目标 60 产生一定的作用。下面参照图 27 和 28 详细说明。

在步骤 11，执行背景（静止）目标的处理．在这一处理中，以部分从存储区 22m 传送来的程序和从存储区 24c 传送来的静止目标的多边形数据为基础，经计算确定静止目标 71a - 71n 的显示位置与形状．下面参照图 29 详细说明。

在步骤 12， RSP122 执行让与处理。即在 CPU11 控制下，根据可动目标（诸如敌人、游戏者、同伴）和静止目标（诸如存储在 RAM15 的图像数据区 154 中的背景）的结构数据， RCP12 对图像数据执行变换处理（坐标变换处理与帧存储器让与处理），显示可动目标与静止目标。具体讲，为了附加对目标确定的结构数据所指定的色彩，把色度数据写入对应于三角形平面（用于构成可动目标或静止目标的多个多边形）的存储区 154d 中的地址。下面参照图 30 详细说明。

在步骤 13， RCP12 根据诸如信息、音乐和效果音响等伴音数据执行伴音处理。下面参照图 31 详细说明。

作为 RCP12 在步骤执行让与处理的结果，读出存储在帧存储区 152 中的图像数据，从而在步骤 14 在显示屏 31 上显示出游戏者目标、可动目标、静止目标、敌方目标等。

在步骤 15，通过读出 RCP12 在步骤 13 经伴音处理而得到的伴音数据，输出诸如音乐、效果音响或话音一类的伴音。

在步骤 16，确定过程是否清除（过程清除检测）．如果未清除过程，则在步骤 17 确定游戏是否结束。如果游戏未结束，处理返回步骤 3，于是重复步骤 3 - 17，直到检测到游戏结束的状态．如果对游戏结束的条件进行检测，例如对游戏者原谅的差错数变为某个预定数或者游戏者目标的寿命耗去了某一预定

量，则在后面的步骤 18 以选择性处理继续游戏或保存备用数据来执行游戏结束处理。顺便提一下，在步骤 16，如果检测出清除过程的条件（如主体击倒等），就在步骤 19 执行过程清除处理，接着处理返回步骤 1 。这里，过程清除处理涉及例如在装到相应的过程得分寄存器之前立即装载游戏中获得的寄存器存储的过程得分，把过程得分指示为图 11 中的过程点。如果清除了多个过程，则确定并显示总点数。顺便提一下，根据要求可对过程清除计算的过程得分加津贴点。

下面对每个子程序的具体操作加以说明。

参照图 16，对作为本发明特征的过程选择（主程序的步骤 1 ）作一详述。在步骤 101， CPU11 作显示处理，在过程得分显示区 72a - 72e 上显示过程得分，过程得分由游戏者对每个过程获得并存储在图 9 所示 RAM15 的寄存区 159R 中。具体讲， PCU11 把代表得分的图像数据送入显示清单。此后除非另有说明，否则显示处理参照为对显示清单的图像输入。在步骤 102， CPU11 对过程得分求总，执行显示处理以在总得分显示区 72 中显示总分。在步骤 103， CPU11 作显示处理，以在机身计数量显示区 74 中显示可用的先前目标数。

顺便提一下，除了在步骤 101 、 102 和 103 的显示处理外，过程显示处理或高分显示处理等虽未在图 16 中示出，但都要执行。过程显示处理是由 CPU11 输入图 9 中在过程显示区 80 中所示的这种图像数据。

在步骤 104， CPU11 确定起动开关 47S 是否已按下。如果起动开关 47S 未按下，处理就进入步骤 105 。在步骤 105， CPU11 确定按钮开关 47A 是否已按下。如果开关 47A 未按下，就执行准备在每帧上执行的图像处理与伴音处理。在显示器 30 上显示了预定图像并输出了伴音以后，处理返回步骤 104 。如果开关 47A 已按下，处理进入步骤 110 。

另一方面，在步骤 104，如果起动开关 47S 已按下，处理就进入步骤 106 。在步骤 106， CPU11 对图 10 所示的过程选择窗 75 执行显示处理。过程选择窗 75 用“进行过程”“改变过程”“重来过程”这三项作显示，并以光标（指定装置）可动显示，指定选择哪一项。虽然光标是用图 10 中的箭头表示的，但是可以转换选择项的色彩，或由图像样式指定某一项。

在步骤 107， CPU11 执行移动处理以垂向移动指定装置。具体讲， CPU11 检测操纵杆 45 是前拉还是反推。如果是拉，指定装置就下移，而如果是推，则指定装置上移。在步骤 108， CPU11 确定按钮开关 47A 是否已按下。如果开关

47A 未按下，就对每帧执行图像处理与伴音处理。在显示器 30 上显示预定图像并输出伴音以后，处理返回步骤 107。如果开关 47A 已按下，则处理进到步骤 109。在步骤 109， CPU11 确定指定装置开关 47A 按下选择哪一项，以确定选择的项昌否为“进行过程”。若选择项是“进行过程”，处理进到步骤 110。

在步骤 110， CPU11 把有关同伴目标在整个过程中的数据存入 RAM15 的同伴数据存储区 158。同伴目标数据进行于过程即将清除之前的同伴目标状态。本发明中，同伴目标被构成如在战斗中遭重创就撤离前线。在同伴目标撤离前线的情况下，若损坏严重，则不能指望进入下次过程。若是轻伤，可参加下一次过程，这种能否参与下一次过程的状况就是同伴目标状态。

在步骤 111， CPU11 对下次过程执行初始设定，包括例如帧设置、各种参数设置、把要显示的各种目标数据写 RAM15 等。步骤 111 结束后，处理返回主程序并进到步骤 2。

另一方面，在步骤 109，若选择项不是“进行过程”，处理就进到步骤 112。在步骤 112， CPU11 确定选择项是否为“改变过程”。若选择了“改变过程”，处理进到步骤 113。在步骤 113，如图 9 所示， CPU11 执行设置以选择一个与下接过程不同的过程。例如，把设置成从过程 3 进到过程 4 改为设置成从过程 3 进到过程 7 等。然后处理进行步骤 104。

同时，在步骤 112，若选择项不是“改变过程”，则处理进到步骤 114。在步骤 114， CPU11 将存储在 RAM15 寄存区 159R 里的游戏者目标计数减 1。在步骤 115，从 RAM 的同伴数据存储区 158 读出同伴目标数据。在步骤 116，对即将清除的过程作初始设置。当步骤 111 结束后，处理返回前述程序并进到步骤 2。

现参照图 17 说明模式转换处理子程序的操作（主程序步骤 3）。若游戏者目标达到图 11 中的模式转换点，在步骤 121 确定（或检测）存在全程模式位置，并在步骤 123 确定全程模式的演示（下称“ DEMO ”）处理是否结束。在确定 DEMO 处理未结束，在步骤 124 执行全程模式的 DEMO 图像处理。在步骤 124 执行伴音处理，产生全程模式的 DEMO 伴音，然后处理进到前述步骤 12 的让与处理。

另一方面，若在步骤 123 确定 DEMO 处理已结束，在步骤 126 已完成全程模式的转换处理（把模式标志 F2 改为全程模式），处理就返回主程序。

其优点在于转换卷动范围时不会产生令人陌生的感觉，即屏幕卷动的方向突然从单向卷动模式改为全程模式。而且，与在整个过程期间把卷动范围置于整个范围的情况相比，在单向卷动期间转换卷动范围减轻了 CPU 的负担。而且与在整个过程期间的单向卷动相比，还可以多种方式作卷动显示。这样，可对一种游戏有各种各样的图像表示，可进一步提高游戏兴趣。

现参照图 18 说明控制器处理的子程序操作（步骤 4 ）。在步骤 131 确定对控制器数据有无读请求指令。若没有，在步骤 131 等待产生读请求指令。若确定有读请求指令，在步骤 132 向控制器控制电路 18 提供一指令。控制电路 18 对此作出响应，执行对控制器 40A - 40D 读操作状态数据的处理。在步骤 133，确定控制电路 18 是否对所有的控制器 40A - 40D 读操作状态数据已结束。若未结束，就等待结束。若检测出已结束，则在步骤 134，从控制电路 18 经总线控制电路 121 把每个控制器 40A - 40D 的操作状态数据写入 RAM15 的存储区 156 。

现参照图 19 、 21 说明与同伴作通信处理的子程序( 上步骤 5 )。在步骤 141a 确定游戏者目标是否已到达位置 A 。若确定未到达位置 A，则处理在步骤 141b 、 141c 、 151a 、 151b 、 151c 、 151d 的处理后返回主程序。同时，若在步骤 141a 确定游戏者目标已到达位置 A，在步骤 142a 确定是否有同伴 1 。若第一同伴存在，在步骤 143a 确定是否有正在处理的短衙或信息。若确定有短语正在处理，则在步骤 144a 对短语接通在短语标志 F31 - F3n 中相应的一个标志并作优先级比较，这是选择任一短语必需的。在步骤 145a 确定短语 1 的优先级是否高于在步骤 145a 正在处理的短语。若是高，则处理进到步骤 146a 。在步骤 146a，执行短语 1 的显示处理。例如，短语 1 是第一友方发给游戏者目标的信息( 用刹车克服 )，使它防止受位置 A 出现的敌方攻击。作为一种操作方法，显示的信息命令按下开关 47C 下方的按钮（开关 24Cd ）。在步骤 147a 执行以伴音输出短语 1 的处理。顺便提一下，若在步骤 143a 确定无短语在处理，无需判断优先级，则处理进到步骤 146a 。若在了 142a 确定无第一同伴或在处理的短语的优先级低于在步骤 145a 确定的短语 1，则处理返回主程序。

另一方面，若确定游戏者目标在位置 B 而不在位置 A，则执行步骤 141b - 147b 的操作。步骤 141b - 147b 是输出短语 2，同步骤 141a - 147a 的操作类似。因此，相应的步骤号后接符号 “ b ” 而不是 “ a ”，故省略对它的说明。

同时，在短语输出条件依赖于时间的情况下，例如它依赖于寻找主角的时间

周期 A，则在步骤 141c 确定时间周期 A。当在步骤 142c 确定附近出现第二友方时，执行步骤 143c - 147c 操作。这些步骤 143c - 147c 同第二友方发出信息（图 13 中短语 3）告知如何击倒主角的操作（袭击方法）有关，除了同伴与短语不同外，该操作与步骤 142a - 147a 的操作一样，故省略对它的说明。

关于短语输出条件，第三同伴注视着敌方，与步骤 151a 确定的内容相同，故执行步骤 152a - 156a 的操作。步骤 152a - 156a 的操作是输出信息（图 13 中短语 5），由同伴告知如何战胜主角，除短语有异外，其操作与步骤 142a - 146a 的操作相同。

当短句输出条件是营救第三同伴时，在步骤 151b 确定同样内容，从而执行步骤 152b - 156b 的操作。步骤 152b - 156b 是当营救第三同伴时产生短语 6 的操作，除了短语不同外，其余与步骤 152a - 156a 的操作相同。

同时，在短语输出条件是遭敌方袭击的游戏者目标输出短语 8 的情况下，在步骤 151c 确定同样内容以执行步骤 152c - 156c 的操作。若短语输出条件是输出与主角被击倒的短语 9，则在步骤 151d 确定同样内容以执行步骤 152d - 156d 操作。

如上所述，以显示或伴音输出信息（图 13 例中的短语 1 - 4），从而帮助游戏者作合适的操作。有了合适的告示操作方法，游戏就便于进行下去，即使操作方式较难，也可给游戏者有成就感或感到满意。这样，容易清除场景或过程。若根据游戏的场景或形势以显示和/或伴音方式输出正确的信息（图 13 例中的短语 5 - 9），在游戏中就有高度逼真的表达，增强了游戏娱乐性。

顺便提一下，产生用以帮助游戏者根据信息或形势正确操作的显示或伴音信息并不限于图 13 的实施例，可按游戏类型或内容作适当修改，所以不限于实施例的说明。例如，为简化说明，虽然对开关的操作方法是针对任何一只开关被按下的情况说明的，但是也可确定多次按压同一开关或按压预定组合的多只开关。

现参照图 22 - 23 说明材料补充处理子程序的操作（步骤 6）。在游戏目标来到预定地点或有一项目的位置之前，在步骤 161 确定游戏者目标还未进入该地点。在步骤 163，确定还未设置显示项目的时间周期（T1）（T1 = 0）。在步骤 170 确定未处于呈现标记（项目箱）的条件，而该标记代表有权获得某个项目。在步骤 172 确定未设置项目，处理返回主程序。之后，以帧周期执行主程序处理。

当游戏者目标来到可获得项目的地方时，在步骤 161 确定该项目。在步骤

162，对定时器寄存器设置一恒定时间周期（ T1 ），作为显示一标记告诉处于可获取项目的条件的时间周期．在步骤 163 确定周期 T1 大于 0 ．在步骤 164 作减去单位时间（如 1 秒）的减法（ T1 － 1 ）．在步骤 165 ，显示标记，游戏者可用它在按下项目显示请求开关（如 47Cr ）时获得一项目．在步骤 166 确定项目请求开关是否按下．若确定未按下，执行步骤 170 、 172 ，然后处理返回主程序．对每个帧周期重复步骤 161 、 163 － 166 、 170 、 172 ，于是开关 47Cr 在预定时间周期内等待被按下。

在如上等待操作的重复期间，若在步骤 166 确定已按下显示请求开关，对定时寄存器置 0 （复位），在步骤 168 准备输出代表同伴请求项目的短语．该短语在步骤 14 与 15 以声图方式输出．在步骤 169 ，执行显示标记（项目箱）的处理（项目设置处理），表示处于可获得项目的条件．在步骤 170 ，确定可以显示项目箱的条件．在步骤 171 ，执行显示项目箱的处理．当在步骤 172 确定正在显示项目箱时，在下一步 173 确定游戏者是否已执行获取项目箱的操作（如向项目箱射击或把游戏者目标重叠在项目箱上等操作）．若确定已获得项目箱，则在步骤 173 － 180 作处理，根据游戏者目标的状况提供要求的项目．例如，若在射击游戏中游戏者目标是飞机，则在步骤 174 确定机翼是否处于预定状态．若不存在预定的机翼，则在步骤 175 提供一机翼作为补充项目．若存在预定的机翼，在步骤 176 确定寿命或损伤承受量是否大于恒定值（ 128 ）．若未确定大于，在步骤 177 提供恢复寿命的项目．若寿命大于恒定值（ 128 ），在步骤 178 确定有无双管炮（双管）．若确定有，在步骤 179 提供双管。若确定存在，在步骤 180 提供炸弹。

以这种方法，根据游戏者目标的状态提供有利于游戏者推进游戏的项目，从而有助于游戏者使游戏继续推进前面的场景或清除过程．这样，游戏者易于获得成就感或对游戏满意．而且，游戏者在游戏时能在实际控制飞机时感到好像接受到命令，或得到帮助而飞行，提高了游戏的兴趣．顺便提一下，提供的项目视游戏的类别或内容而异，因而游戏软件开发者可参照本例的技术思路作出各种修改。

现参照图 32 、 33 、 36 、 38 、 39 和 42 的流程说明处理游戏者目标的子程序．首先在步骤 301 ， CPU11 读出存储在控制垫数据区里的操纵杆数据并作校正．具体讲，删去操纵杆中部的数据，使杆来到中央位置（如半径为 10 计数）时数据为“ 0 ”．这样使操纵杆数据正确假定为“ 0 ”，尽管因操纵杆制造或游

戏者手指轻微抖动而有误差．而且，在操纵杆可操作范围内的外周部分对预定范围作了校正．具体讲，在游戏中校正成不输出不希望的数据部分。

在步骤 302， CPU11 确定准备供游戏使用的操纵杆数据 Xj、 Yj．因为步骤 301 确定的数据是计数器 444 的值，所以要转换成便于游戏处理的形式。具体讲，杆不倾斜时 Xj 为“ 0 ”，在 - X 轴向（向左）倾斜最大时为“ + 60 ”，而在 + X 轴向（向右）倾斜最大时为“ - 60 ”．杆不倾斜时 Yj 为“ 0 ”，在 + Y 轴向（向前）倾斜最大时为“ + 60 ”，而在 - Y 轴向（向后）倾斜最大时为“ - 60 ”．

在步骤 303， CPU11 在三维游戏空间设置游戏者目标的基本速度为 As0．As0 是游戏者目标在 1 帧内前进的游戏空间距离，可按游戏的过程或场景自由设置。在步骤 304， CPU11 读出存储在控制垫区中的按钮开关 C1 数据，确定游戏者是否按下开关 C1．开关 C1 用作提速开关，用于提高游戏中游戏者目标的前进速度．按下开关 C1 时，游戏者目标例如在飞机的情况下作图像处理，加强发动机喷射和轰鸣。若确定游戏者正在按开关 C1，处理进到步骤 305．在步骤 305，CPU11 处理向控制器的振动发生电路 53 提供振动发生信号，结果，在按下开关 C1 时，控制器就振动．不按开关 C1 时，停止振动．控制器在提速期间以这种方式振动，增强了游戏的逼真感，便令游戏者增加兴趣。

在步骤 306， CPU11 作 As = As0 + Asα 运算． As 为游戏者目标的速度，Asα 为提速期间游戏者目标提高的速度。该运算增大了游戏者目标下一帧内在三维位置中的移动距离。接着处理进到步骤 309．

另一方面，若在步骤 304 确定游戏者不在按开关 C1，处理进到步骤 307．在步骤 307， CPU11 读出存储在控制垫数据区中的有关开关 47Cd 的数据，确定游戏者是否按下开关 47Cd．开关 47Cd 作用刹车开关。在游戏中减慢游戏者目标的前进速度。按下开关 47Cd 时，游戏者目标例如为飞机的情况下作发动等图像处理，产生发动轰鸣声．若确定游戏者正在按开关 47Cd，处理进到步骤 308．在步骤 308， CPU11 像步骤 305 那样作振动处理。

在步骤 309， CPU11 作 As = As0 - Asβ 运算， Asβ 是使用刹车期间游戏者目标减慢的速度。该运算减短了游戏者目标下一帧三维位置中的移动距离。然后处理进到步骤 310．

同时，在步骤 307，若确定游戏者目标不在按开关 47Cd，则处理进到步骤 310．在步骤 310， CPU11 确定是否是全程模式。具体讲， CPU11 检测 RAM15

标志区 159f 中的全程模式标志是置位还是复位. 若全程模式标志置位，在全程模式中确定当前游戏模式. 若全程模式标志处于复位状态，则把当前游戏模式定为单向卷动模式。若确定不是全程模式，处理进到步骤 320 。

在步骤 320， CPU11 按下列方程操作. 注意, 在本说明书中, 乘法标为“ * ”。

θ x1 = 0.68 * Xj

θ Y1 = 0.68 * Yj

Xs = - As * Sin θ x1

Zs = - As * Con θ x1

Ys = - As * Sin θ y1

下面参照图 34 说明θ x1 、 θ x1 、 Xs 、 Zs 、 Ys 的含义。 θ x1 是游戏者目标 60 的 As 方向矢量在 XZ 坐标（从上面观看的游戏者目标 60 的平面坐标系）上相对于 - Z 轴向的角度，操纵杆向左斜，其值在 + 向增大，操纵杆右斜，其值在-向增大、由方程可知，值θ x1 在 - 40.8 (0.68* 60)到 40.8 (0.69 * (-60))增减，正比于杆的倾斜度。由于游戏者目标 60 向 As 方向行进，所以左右范围的方向变化为 40.8 (0.68 * 60)到 40.8 ( 0.68 * (-60))度。例如，该方向是机身机头方向，这里把图 35 中的飞机用作游戏者目标 60 。在使用坦克时，它可以是炮的方向，在使用动物时，它又可以是脸的方向.

θ y1 是游戏者目标 60 的 As 方向矢量在 YZ 坐标( 在左到右平面中观看游戏者目标时的平面坐标系）上相对于 - Z 轴方向的角度，操纵杆前倾时其值在 + 向增大，操纵杆后倾时其值在-向减小。由方程可知，值θ x1 在 - 40.8 (0.68 * 60)到 40.8 ( 0.68 * (-60))增减，正比于操纵杆的倾斜度。因为游戏者目标 60 向 As 方向前进，故它在 40.8 (0.68 * 60)到 40.8 ( 0.68 * (-60))度范围内垂直改变方向。

Xs 、 Zs 、 Ys 分别为 As 的 X 轴分量、 Y 轴分量和 Z 轴分量。

在步骤 321， CPU11 确定是否有游戏者目标 60 的左边部分。由于本例用飞机作为说明游戏者目标的一个例子，故确定有无左翼。注意，若用坦克作为游戏者目标，则用履带代替机翼。若用动物，则是肢体。若确定有游戏者目标 60 的左部，则处理进到步骤 322 。在步骤 322， CPU11 确定是否有游戏者目标 60 的右部。若确定有右部，处理进到步骤 323 。

在步骤 323， CPU11 按下列方程操作。

Xa = Xa + Xs

Ya = Ya + Ys

Za = Za + Zs

该运算对前一帧上的坐标（ Xa、 Ya、 Za ）（上述方程的右边）增加了游戏者目标的速度（ Xa、 Ya、 Za ），这样就确定了后继帧的坐标（ Xa、 Ya、 Za ）（上述方程的左边）。于是处理进到图 36 的步骤 330。

另一方面，当在步骤 321 确定游戏者目标 60 无左部时，处理进到步骤 324。CPU11 在步骤 324 按下列方程操作。

Xs = Xs - 2.5

Ys = Ys - 2.5

该运算减小了在 X 轴向与 Y 轴向的速度. 因此，在游戏进行中，显示的游戏者目标 60 好像被拉向左下方向. 例如，即使操纵杆不倾斜，则在面向屏幕观看时，显示的游戏者目标 60 好像慢慢地移向左下方。

同时，若在步骤 322 确定游戏者目标 60 无右部，处理进到步骤 325。在步骤 325， CPU11 按下列方程操作。

Xs = Xs + 2.5

Ys = Ys + 2.5

该运算提高了在 X 轴向的速度，但降低了 X 轴向的速度。因此，在游戏进行时，若面对屏幕看，显示的游戏者目标 60 好像被拉向右下方。例如，即使操纵杆不倾斜，也会以逐渐称向右下方的方式显示游戏者目标 60 。接着处理进到步骤 323。

另一方面，若在步骤 310 确定 CPU11 不处于全程模式，则处理进到步骤 330。

首先说明单向卷动模式与全程模式的差别。单向卷动模式的一个坐标系如图 34 所示，而全程模式有图 37 所示的两种坐标系。对全程模式的各坐标系说明如下。图 37 的 X1(- Z1)坐标是用于表示其中有多种目标的三维的的坐标，在游戏中，这些目标总以固定方式设置。（本例中，该坐标称为“游戏空间坐标”。）X2(- Z2)坐标则是一种基于游戏者目标的运动而改变的坐标。由于 - Z2 轴向对应于下面要说明的摄像机处理的摄像机拍摄方向，所以 Z2 轴代表垂直于显示屏的方向。（本例中，称为“游戏者坐标”。）在此游戏者坐标中，游戏者目标 60 通常执行类似于在单向卷动模式中的操作。以此种方式应用两种坐标系，就能即使用单向卷动模式系统，又能设计一种能以每一方向卷动的全程模式系统。因

此，即使改变了模式，操作方式几乎不变，故游戏者容易参与游戏。

在步骤 330， CPU11 按下列方程操作。

θ x1=0.68*Xj

θ y1=0.68*Yj

θ x2=0.02*θ x1+θ x2(0.03*θ x1 每帧增加)

Xs=-As*sin(θ x1+θ x2)

Zs=-As*cos(θ x1+θ x2)

Ys=-As*sin θ y1

现参照图 37 说明θ x1 、θ y1 、θ x2 、 Zs 、 Zs 和 Ys 。θ x1 是游戏者目标 60 的 As 方向矢量在 XZ 坐标( 从上面以平面观看游戏者目标 60 时的游戏者坐标系 ) 上相对于 - Z2 轴向的角度，其值在操纵杆左倾时在 + 向增大，右倾时在-向减小。

θ y1 是游戏者目标的 As 方向在在 YZ 坐标 ( 从左到右平面观看游戏者目标 60 时的游戏者坐标系 ) 上相对于 - Z2 轴向的角度，其值在操纵杆前倾时在 + 向增大，后倾时在-向减小。

θ x2 是游戏者目标 60 在 XZ 坐标( 从上面的平面观看游戏者目标 60 时的游戏者坐标系 ) 上 - Z1 轴向与 - Z2 轴向之间的角度，在每一帧上，其值在增减正比于θ x1 。即，其值在操纵杆左倾时在 + 向增大，右倾时在-向增大。由方程可知，值θ x2 与杆的倾斜度呈比例地变化，在每一帧上的增减范围为 - 224( = 0.03 * 0.68 * ( - 60)) 到 1.224( = 0.03 * 0.68 * 60)。

Xs 、 Zs 、 Ys 分别是 As 在游戏空间坐标中的 X 轴分量、 Y 轴分量和 Z 轴分量。

在步骤 332， CPU11 确定游戏者目标 60 是否存在右部，若确定有右部，处理进到步骤 333 。

在步骤 333， CPU11 确定目标 60 是否存在右部，若确定有右部，处理进到步骤 334 。在步骤 334， CPU11 类似于步骤 323 按下列的操作。

Xa = Xa + Xs

Ya = Ya + Ys

Za = Za + Zs

该运算在前一帧上把游戏者目标 60 在游戏空间坐标系中的速度 ( Xs 、

Ys 、 Zs ）加到游戏空间坐标（ Xa 、 Ya 、 Za ），从而确定后继帧上的游戏人坐标（ Xa 、 Ya 、 Za ）。接着处理进到图 38 的步骤 350 。

另一方面，若在步骤 332 确定游戏者目标 60 无左部，则处理进到步骤 335 。CPU11 在步骤 335 按下列方程操作。

Xs = Xs - 2.5 * Cos θ x2

Zs = Zs - 2.5 * Sin θ x2

Ys = Ys - 2.5

该运算在游戏者空间中减小了 X 轴向和 Y 轴向的速度。因此，在游戏中， 若从面对屏幕看， 显示的游戏者目标 60 好像被拉向左下方。例如即使杆不倾斜，面对屏幕观看时， 显示的游戏者目标 60 逐渐移向左下方。

若在步骤 333 确定游戏者目标 60 无右部，则处理进到步骤 336 。在步骤 336 ，CPU11 按下列方程操作。

Xs = Xs + 2.5 * Cos θ x2

Zs = Zs - 2.5 * Sin θ x2

Ys = Ys - 2.5

该运算在游戏空间内提高了 X 轴向速度而减低了 Y 轴的速度。因此，在游戏中， 面对屏幕观看时， 显示的游戏者目标 60 好像被拉向右下方。例如， 即使杆不倾斜， 显示的目标 60 仍逐渐向右下方。于是处理进到步骤 334 。

顺便提一下，为简化起见，虽然以图作了说明，从而控制器的角度直接反映在每帧的游戏者目标上，但是具体讲，在从 0.68 * Xj 推演出 θ X1 时，要作滤波处理。同样地，从 0.68 * Yj 推演出 θ y1 也要作滤波处理。现详细说明这一滤波处理，在多个帧上， CPU11 让 θ x1 值逐渐接近 0.68 * Xj ，充在处理过这些帧的某一时刻得到 θ x1 = 0.68 * Xj ， 并不在一帧上使值 0.68 * Xj 立刻变成 θ x1 。例如，把 0.68 * Xj 除以 Nf （预定的自然数），并在 Nf 帧周期内把 0.68 * Xj / Nf 加到 θ x1 ，由此得到 θ x1 = 0.68 * Xj 。从 0.68 * Yj 里推演出 θ y1 的滤波处理类似于从 0.68 * Xj 里导出 θ x1 的滤波处理。利用这种滤波处理，可防止游戏者目标 60 因突然操作操纵杆而发生极度移动以及可怕地改变场景，否则要加重游戏者观察的负担而难以控制。

在步骤 350 ， CPU11 对目标 60 作击中确定，下面用步骤 351 - 354 作一详述。步骤 351 - 354 是常规击中确定，后面谈的同伴目标击中确定也相似。

在图 39 的步骤 351， CPU11 确定 ABS （ OBJ2X - OBJ1X )是否≤OBJ1r。OBJ1 是要确定击中的目标，本例中为游戏者目标 60 。 OBJ2 是移向 OBJ1 的目标，本例中为攻击目标，诸如同伴目标、敌方目标、静止目标、敌方目标发射的激光目标。 OBJ1X 是 OBJ1 的 X 坐标值，而 OBJ2X 是 OBJ2 的 X 坐标值。 OBJ1X 与 OBJ2X 可以在游戏空间坐标上或游戏者坐标上，只要它们是公共坐标系上的 X 坐标值就行。 ABS( )表示在( )内给定数的绝对值。在 OBJ1 认为是立方体的情况下， OBJ1r 是表示立方体一面半长的值，换言之， OBJ1r 是代表 OBJ1 击中确定范围的值。若 ABS （ OBJ2X - OBJ1X ）≤ OBJ1r 成立，进程到达步骤 302 。

在步骤 352， CPU11 确定 ABS （ OBJ2Y - OBJ1Y )是否 ≤ OBJ1r。 OBJ1Y 是 OBJ1 的 Y 坐标值，而 OBJ2Y 是 OBJ2 的 Y 坐标值。 OBJ1Y 与 OBJ2Y 可以在游戏空间坐标上或游戏者坐标上，只要它们在公共坐标系上具有 Y 坐标值就行。若 ABS （ OBJ2Y - OBJ1Y ）≤ OBJ1r 成立，处理进到步骤 353 。在步骤 353 ，CPU11 确定 ABS （ OBJ2Z - OBJ1Z ）≤ OBJ1r 是否成立。 OBJ1Z 是 OBJ1 的 Z 坐标值，而 OBJ2Z 是 OBJ2 的坐标值，二者可在游戏空间坐标上或游戏者坐标上，只要它们在公共坐标系上具有 Z 轴坐标值就行。如果 ABS （ OBJ2Z - OBJ1Z ）≤ OBJ1r 成立，处理进到步骤 354 。

在步骤 354， CPU11 确定 OBJ2 与 OBJ1 被击中，在 RAM15 的标志区 159F 中设置击中确定标志，本例中的游戏者目标 60 由三个目标组成，即主体目标 61 （下称主体 61 ）、右翼目标 60L 和右翼目标 60R ，通过输入到显示清单，可在显示器 30 上显示各个目标。同时，如图 35 所示，游戏者目标 60 共有 4 个确定点，位于主体 61 上部、主体 61 下部、左翼 62 靠近主体部分和右翼靠近主体部分，所以它们各有 1 点。这些点在步骤 351 - 354 接受击中确定，确定是否应对每点设置击中标志。接着处理返回前一程序。

同时，若在步骤 351 确定 ABS(OBJ2x-OBJ1x)不是⩽ OBJ1r ，处理返回前一程序。

另一方面，若在步骤 352 确定 ABS(OBJ2y-OBJ1y)不是单元⩽ OBJ1r ，处理返回前一程序。

另一方面，若在步骤 353 确定 ABS(OBJ2z - OBJ1z)不是⩽ OBJ1y ，处理返回前一程序。

另一方面，若结束了步骤 350 ，处理进到步骤 360 。

在步骤 360， CPU11 确定在立体 61 下部游戏者目标 60 是否被另一目标击中。具体讲， CPU11 检测是否在 RAM15 的标志区 159F 中设置击确定标志。若主体 61 的下部被另一目标击中，处理进到步骤 361 。 CPU11 在步骤 361 按下列方程操作。

$$R1 = R1 - 40$$

R1 代表主体 61 承受的伤害程度，本例中其初始值为 255 。当主体 61 下部受损时， R1 减 40 。例如，受损 7 次，则 R1<0 。

在步骤 362， CPU11 按下述方程操作。

R2(或 R3)=R2(或 R3) - 15

R2 代表左翼 62 能承受的损伤程度，在本例中其初始值为 60 。同样地， R3 代表右翼 63 能承受的损伤程度，初始值为 60 。在该步骤，左右随意选择，以让选择的机翼承受更大损伤。当左翼 63 或右翼 63 受伤时， R2 或 R3 减 15 。例如，受伤 4 次得 R2(或 R3)=0 。

在步骤 363， CPU11 确定游戏者目标 60 的主体 61 上部是否被另一目标击中，若被击中，处理进到步骤 364 。

CPU11 在步骤 364 确定 R1 是否 $\leqslant 0$ 。换言之，确定游戏者目标 60 的受伤程度是否能把游戏继续下去。若 R1 不是 $\leqslant 0$ ，处理进到步骤 365 。

在步骤 365， CPU11 把主体 61 输入显示清单。主体 61 是构成游戏者目标 60 中部的目标。

在步骤 366， CPU11 确定 R2 与 R3 是否 $\leqslant 0$ 。换言之，确定左翼 62 和右翼 63 的损伤程度能否忍受。若 R2 与 R3 不是 $\leqslant 0$ ，处理进到步骤 367 。

在步骤 367 、 CPU11 确定 R2 是否 $\leqslant 0$ 。换言之，确定左翼 62 的损伤程度是否为左翼能忍受。若 R2 不是 $\leqslant 0$ ，处理进到步骤 368 。

在步骤 368， CPU11 确定 R3 是否 $\leqslant 0$ 。换言之，确定右翼 63 的损伤程度是否为右翼能忍受。若 R3 不是 $\leqslant 0$ ，处理进到步骤 369 。

在步骤 369 、 CPU11 把完整的左翼目标 60L 输入显示清单。运用这种登录，显示器 30 能显示图 35 所示的游戏者目标 60 。于是处理返回前一程序进到步骤 8 。

另一方面，若在步骤 366 有 R2 与 R3 $\leqslant 0$ ，处理进到步骤 370 。在步骤 370 ，

CPU11 把无左翼 62 的左翼目标 60L 和无右翼 63 的右翼目标 60R 输入显示清单。运用这一输入，显示器 30 显示出失去左翼 62 和右翼 63 的游戏者目标 60 作为左右翼部分，如图 40 所示，接着处理返回前一程序进到步骤 8 。

同时，若在步骤 367 有 R2 ≤ 0，处理进到步骤 371 。在步骤 371， CPU11 把失去左翼 62 的左翼目标 60L 和完整的右翼目标 60R 寄存到显示清单。这一登录使显示器 30 显示出以失去左翼 62 的游戏者目标 60 作为左翼部分，不过有正常的右翼，如图 41 所示。于是处理返回前一程序进到步骤 8 。

另一方面，若在步骤 368 有 R3 ≤ 0，步骤进到 372 。在步骤 372， CPU11 把失去右翼 63 的右翼目标 60R 和完整的左翼目标 60L 输入显示清单。这一输入使显示器 30 把失去右翼 63 的游戏者目标 60 显示为右翼部分但是有正常的左翼，同图 41 的游戏者目标 60 相反。于是处理返回前一程序进到步骤 8 。

另一方面，若在步骤 364 有 R1 ≤ 0，处理进到步骤 373 。在步骤 373， CPU11 对游戏者目标 60 执行爆炸处理。在爆炸处理中，跳过多帧的其它处理，在显示器 30 上显示爆炸目标 60 的场景。之后， RAM15 中的寄存区 159 的值减 1 ，处理进到步骤 17 。若游戏未结束，从游戏空间中的预定位置恢复游戏。

同时，在步骤 363 ，若主体 61 的上部未被另一目标击中，处理进到步骤 374 。

在步骤 374， CPU11 执行在多个帧上把游戏者目标 60 向上移动的处理，这一处理表示目标 60 的下部好像与另一目标碰撞而推开。处理进到步骤 364 。

另一方面，若主体 61 的下部在步骤 360 未被另一目标击中，处理进到步骤 370 。

在步骤 375， CPU11 确定目标 60 的主体 61 上部是否被另一目标击中，若被击中，处理进到步骤 376， CPU11 按下述方程操作。

R1=R1 - 40

在步骤 377， CPU11 下述方程操作。

R2(或 R3)=R2(或 R3) - 15

在步骤 378， CPU11 在多个帧内执行使目标 60 下移的处理，这一处理表示目标 60 的上部好像与另一目标相撞而推开。于是处理进到步骤 364 。

另一方面，若另一目标在步骤 375 未击中主体 61 的上部，处理进到步骤 380，CPU11 确定主体 61 和目标 60 的右翼目标 60R 是否未被另一目标击中而只是左翼目标 60L 被另一目标击中。若只是左翼目标 60L 被另一目标击中，处理进到步骤

381， CPU11 按下述方程操作。

R2=R2 - 20

然后处理进到步骤 365 。

另一方面，若在步骤 380 只是左翼目标 60L 未被另一目标击中，处理进到步骤 382， CPU11 确定是否主体 61 与目标 60 的左翼目标 60L 未被另一目标击中而只是右翼目标 60R 被另一目标击中。若只是右翼目标 60R 被另一目标击中，处理进到步骤 383， CPU11 按下述方程操作。

R3=R3 - 20

处理进到步骤 365 。

同时，若在步骤 382 只是右翼目标 60R 未被另一目标击中，则处理进到步骤 365 。

现在说明处理戏者目标的另一个实施例。本例具有左右翼目标 60L 和 60R ，各翼由三部分而不是二部分组成。具体讲，左右翼各由三部分形成，在显示器 30 上显示时，受损只是断脱外部。如图 35 所示，左右翼目标 60L 和 60R 各自包括左翼 62 与 64 及右翼 63 与 65 。例如，在显示器 30 上显示图像，其左翼目标 60L 在受轻伤时，例如 0<R2 ⩽ 30 ，就如图 41 所示失去左翼 62 ，而受重伤时(如 R2 ⩽ 0)就如图 43 所示失去左翼 64 。对右翼 63 与 65 同样作这种处理。

在执行图 33 的步骤 321 - 325 和图 36 的步骤 332 ~ 336 时，根据左右翼目标 60L 与 60R 的状况，向游戏者目标 60 的左右提供二级运动控制，能以多种运动方式控制游戏者目标 60 。这样，为游戏者进一步提供游戏的真实感或成就感。

现对示出图 15 的步骤 8 子程序流程图的图 44 和 45 作一详细。在步骤 500 ，CPU11 确定是否在 RAM15 的标志区 159F 中设置了全程模式标志。若未设置全程模式标志，处理进到步骤 501， CPU11 按下述方程操作。

Xc=(Xa-X0) * 0.8

Xc 是摄像机拍摄游戏者目标 60 的 X 坐标位置，类似于图 34 。摄像机是假设的摄像机，以某个角度确定显示在显示器 30 上的三维空间内的目标。 Xo 是偏离值，即在游戏者目标 60 运动的 X 坐标中心与三维空间 X 坐标原点之间存在的偏离，该值主要在过程转移时应用。本例中 Xo 取 0 。

在步骤 502， CPU11 按下述方程操作。

Yc=(ya-Y0) * 0.8

Yc 是摄像机拍摄游戏者目标 60 的 Y 坐标位置。 Yo 是目标 60 移动的 Y 坐标中心与三维空间 Y 坐标原点之间存在的偏离值，该值主要在过程转移时应用，本例中 Y0 取 0 。

在步骤 503， CPU11 确定游戏者是否在按按钮开关 C1 。若游戏者不在按开关 C1 ，处理进到步骤 504 。

在步骤 504， CPU11 确定游戏者是否在按按钮开关 47Cd ，若不在按开关 47Cd ，处理进到步骤 505， CPU11 按下述方程操作。

Zc=Za+400

Zc 是摄像机拍摄游戏者目标 60 的 Z 轴值。这样， Zc 代表从游戏者目标位置在 Z 轴向移动 400 后的 Z 坐标位置。

在步骤 506 ，以卷动方向设置摄像机拍摄方向。如图 34 所示在 - Z 轴向设置。

另一方面，当游戏者在步骤 503 按下开关 C1 时，处理进到步骤 507， CPU11 按下述方程操作。

Zc=(Za+abs(Za))+400

abs( )代表()内给定数的绝对值。 Za 是游戏者目标 60 离开摄像机的值。以这种方式对 Zc 加上 abs(Z α )，使显示器 30 显示的游戏者目标好象从屏幕的纵深方向迅速飞行。接着处理进到步骤 506 。

同时， 当游戏者在步骤 504 按下开关 47Cd 时， 处理进到步骤 508， CPU11 按下述方程操作。

Zc=(Za-abs(Z β ))+400

Z β 是游戏者目标 60 使用刹车而摄像机逼近目标 60 的值。以这种方式由 Zc 减去 abs(Z β )，使显示器 30 显示的游戏者目标 60 好像被迅速拉回屏幕的该方向。处理进到步骤 506 。

另一方面， 若在步骤 500 已在标志区 159F 中设置了全程模式标志，处理进到图 55 步骤 510 。

在步骤 510， CPU11 确定游戏者是否在按下开关 C1 ，若不在按开关 C1 ，处理进到步骤 511 。

在步骤 511， CPU11 确定游戏者是否在按下开关 47Cd ，若不在按开关 47Cd ，处理进到步骤 512， CPU11 按下列方程操作。

Xr=2.0*θ x1*cos θ x2

Zr=2.0*θ x1*sin θ x2

Xc=Xa+400*sin θ x2+Xr

Zc=Za+400*cos θ x2+Zr

Yc=0.8*Ya

同图 37 相似， 2.0 * θ x1 是从游戏者空间中游戏者目标 60 后面使摄像机在(- x)轴向偏离的值。 Xr 是由游戏者空间 X 坐值转换的游戏空间 X 坐标值。 Zr 是由游戏者空间 Z 坐标值转换的游戏空间 Z 坐标值。以这种方式，摄机能以朝向游戏者目标 60 倾斜的方向移动位置且正比于目标 60 的倾斜度。在显示器 70 上作一显示例子，游戏者目标 60 在直飞时，显示在屏幕左右方向的中心，如图 46 所示。然而，当目标 60 像图 47 那样左移时，就显示在屏幕左右方向的右边。同样地，当目标 60 以右方向移动时，显示在屏幕左右方向的左边。这样显示有利于瞄准敌人。因此，游戏者容易瞄准敌人，乐于参加游戏。在步骤 513， CPU11 把摄像机拍摄方向置成 - Z2 轴向，如图 32 所示。

另一方面，在步骤 510，游戏者正在按开关 C1，处理进到步骤 514， CPU11 按下列方程操作。

Xr=2.0*θ x1*cos θ x2

Zr=2.0*θ x1*sin θ x2

Xc=Xa+(400+abs(Z α))*sin θ x2+Xr

Zc=Zx+(400+abs(Z α))*cos θ x2+Zr

Yc=0.8*Ya

于是处理进到步骤 513。

同时，当游戏者在步骤 511 按开关 47Cd 时，处理进到步骤 515， CPU11 按下列方程操作。

Xr=2.0*θ x1*cos θ x2

Zr=2.0*θ x1*sin θ x2

Xc=Xa+(400-abs(Z α))*sin θ x2+Xr

Zc=Zx+(400-abs(Z α))*cos θ x2+Zr

Yc=0.8*Ya

接着，处理进到步骤 513。

现参照图 24 ~ 26 说明同伴目标处理子程序的操作。在步骤 201 确定有无第一同伴，若有，在步骤 202 对第一同伴目标作处理。之后，在步骤 203 确定有无第二同伴，若有，在步骤 204 对第二同伴目标作处理。同样地，在步骤 205 、 206 确定有无第三同伴并对第三同伴目标作处理。这里，除了同伴类型有异外，步骤 202 、 204 、 206 所示的对第一到第三同伴目标的处理均相同，它们可用图 25 和 26 的子程序处理(步骤 211 ~ 230)具体实现。

即，在步骤 211 确定该处理不是暂停处理，而在步骤 212 执行移动处理，对第一到第三同伴中任一个作移动。在步骤 213 确定距离是否在可攻击敌人的范围内。若在可攻击距离范围内，在步骤 214 执行攻击敌人目标的处理(计算和显示发射集束炸弹的处理)。在步骤 215 确定有无友方被敌人追踪，若确定有追踪，则确定 他是否在敌人可攻击距离之内。若敌人在可攻击距离范围内，在步骤 217 执行指示短语 5(如“帮助”)的处理，并在步骤 218 执行短语 5 的伴音输出处理。顺便提一下，若在步骤 215 和 / 或 216 的确定不是同一个(No.)，则在步骤 219 确定游戏者目标是否帮助任何一个同伴目标。若确定要帮助，在步骤 220 执行短语 6(如“我们要帮助！”)的显示处理。在步骤 221 执行短语 6 的伴音输出处理。

在步骤 222 ，当同伴被敌攻击时，作击中确定(如下面要介绍的在步骤 254 确定同伴是否被击中)。在步骤 223 确定他是否被敌人的子弹击中。若被击中，在步骤 224 作从寄存器 R4 值中减去同伴受伤速度处理。在步骤 225 确定寄存器 R4 中受伤承受量是否为 100 或更小。若在 100 以上，处理进到步骤 229 ，若不大于 100 ，则处理进到步骤 226 。在步骤 226 执行显示友方受一定程度损伤或更大损伤须停止战斗返回基地的信息的处理，并在步骤 227 作伴音输出处理。在步骤 228 确定是否结束了停战处理。若未结束，在步骤 229 执行把处理的友方目标输入显示清单的处理。若已结束，则关闭同伴标志 F1 ，处理返回主程序。

现参照图 27 说明敌方目标处理(步骤 10)子程序的操作。在步骤 241 ，寄存器 R5 置 1 ，暂时存储敌方目标数。在步骤 242 根据寄存器 R5 的值确定有敌方目标。在步骤 243 执行子程序(后面说明图 28)处理有多少敌方目标。于是在步骤 244 对寄存器 R5 加 1 。在步骤 245 确定结束的是否是以程序设置的数目显示所有敌方目标的处理。若全部处理都结束，处理返回步骤 242 以重复步骤 242 ~ 245 的处理。

现参照图 28 详细说明对一个敌方目标的处理。在步骤 251 确定敌方目标不

处于爆炸处理中．在步骤 252 对寄存器(E)存储编号的敌方目标作移动处理．在步骤 253 确定游戏者目标或同伴目标是否在射程内．若在射程内，则在步骤 254 进行处理，对射程内出现的游戏者或同伴目标实施攻击。

同时，在步骤 255，在游戏者或友方目标攻击敌方目标的情况下作击中确定．在步骤 56 ，确定游戏者或同伴目标发射的集束炸弹是否击中敌方目标．若检测到击中了，则执行把敌方目标的受伤量减去击中的处理(寄存器 R5 减 1)，并向游戏者提供点数(把按击倒敌人确定的点数加到寄存器 R10 值里的处理)．在步骤 258 确定受伤量是否变为 0 或更小(R5=0)．若不低于 0(R5 ＜>0)，在步骤 261 把处理中的敌方目标输人到显示清单．反之，若小于 0(R5 ＜ 0)，在步骤 259f 进行处理，即把敌方目标炸掉而消失．若在步骤 260 确定已结束爆炸处理，则在步骤 262 关闭已受游戏者目标攻击的敌方目标的标志，于是处理返回主程序。

图 48 的流程图详细列出了处理图 28 中敌方目标的步骤 255 。

在普通游戏机中，确定击中敌人的范围是固定的，故在游戏者攻击敌人时，出现的敌人小难以攻击和击中，使游戏难以进行，减少了游戏者的兴趣．与此相反，本发明的敌人击中确定，其敌方目标击中确定范围增大了，且与敌方目标和游戏者目标 60 之间的距离成正比，使游戏者能攻击并击中甚至隐现的敌方目标。因此，本发明避免使游戏过难，让游戏者作游戏时有适中的难度，令游戏感到满意。

在步骤 600， CPU11 按下述方程运算。

RAD=(sqrt((OBJ2x-OBJ1x) ^ 2+(OBJ2y-OBJ1y) ^ 2+(OBJ2z-OBJ1z) ^ 2)/50

RAD 是正比于被确定击中的两目标间距离而增大的数， sqrt( )是表示（）内给定数方根的函数．符号“ ^ ”是第一所述数与下一所述数乘幂的符号．因此，“ 2 ”表示平方．符号“/”表示除法． OBJ1 是被确定为击中的目标，这里指敌方目标． OBJ2 是朝 OBJ1 来的目标，这里指诸如由游戏者目标 60 发射的激光目标等袭击目标。在 OBJ2 是游戏者目标 60 、同伴目标、静止目标或同伴目标发射的袭击目标，击中确定与图 37 所示相同。

在步骤 601， CPU11 确定 RAD 是否>200，若 RAD 不是>200，处理进到步骤 602 。

在步骤 602， CPU11 确定 ABS(OBJ2x-OBJ1x)是否＜ OBJ1r+RAD ，若是，

处理进到步骤 603 。

在步骤 603， CPU 确定 ABS(OBJ2y-OBJ1y)是否≤ OBJ1r+RAD 若成立，处理进到步骤 604 。

在步骤 604， CPU11 确定 ABS(OBJ2z-OBJ1z)是否≤ OBJ1r+RAD，若成立，处理进到步骤 605 。

在步骤 605， CPU11 确定击中 OBJ2 或 OBJ1，并在 RAM15 的标志区里设置击中确定标志。

处理返回前一程序。

另一方面，若在步骤 602 中 ABS(OBJ2x-OBJ1x) ≤ OBJ1r+RAD 不成立，处理返回前一程序。

另一方面，若在步骤 603 中 ABS(OBJ2y-OBJ1)≤ OBJ1r+RAD 不成立，处理返回前一程序。

另一方面，若在步骤 604 中 ABS(OBJ2z-OBJ1z)≤ OBJ1r+RAD 不成立，处理返回前一程序。

同时，若在步骤 601 中 RAD>260，处理进到步骤 606， CPU11 运算如下。

RAD=200

使 RAD 不超出 200 是为了避免容易不顾远离游戏者目标 60 存在敌方目标而作出错误的确定。若不对 RAD 设一上限，则游戏者目标释放的袭击目标会击中在屏上小得难以鉴别的敌方目标，失去了真实感。

现参照图 29 说明狰处理子程序的操作(步骤 11)。在步骤 271， 静止目标寄存器(R9)置 1 。在步骤 272， 把寄存器(R9)规定的静止目标输入显示清单。在步骤 273， 寄存器 R9 加 1 。在步骤 274 确定结束的处理是否是以程序设置的数量显示全部静止目标。若全部处理还未结束，处理返回步骤 272， 重复步骤 272 - 274 的处理。若全部处理都已结束，处理返回主程序。

现参照图 30 说明让与处理子程序的操作(步骤 12)。在步骤 281 作坐标变换处理。坐标变换处理是在 RCP12 控制下，把诸如动物、游戏者等可动目标以及同伴与存储在 RAM15 的图像数据区 154 里诸如背景等静止目标的多个多边形坐标数据变换成摄像机的观察者眼睛坐标。具体讲，通过运算，把构成多个可动目标与静止目标的多边形数据从绝对坐标变换成摄像机坐标数据，从而变为由摄像机观察的图像。在步骤 282， 对帧存储器作让与处理。作这一处理时，把根据结构

数据确定的色度数据以逐点方式(即在变换成摄机坐标后构成被多边形坐标包围的目标的一个三角形平面上)写入图像缓冲区 152 。此时，写入靠得更近的目标的色度数据，从而根据每个多边形的深度数据优先显示面对此目标(靠近此目标)的目标。据此，把对应于色度数据写的点的深度数据写到 Z 缓冲区 153 中的对应地址。之后，处理返回主程序。

对每一帧在固定时间周期内执行步骤 281 和 282 的操作。然而，要依次对构成在一个场景中显示的多个目标的多边形作处理，且重复到对要在一个场影中显示的所有目标完成处理.

现参照图 31 说明伴音处理子程序的操作(步骤 13)．在步骤 291 ，确定伴音标志是否接通，若确定已接通，在步骤 292 选择要输出的伴音数据。在步骤 293 对选择的伴音数据作读处理，然后处理返回主程序，顺便提一下，由准备以伴音输出的伴音发生电路 16 对这样读出的伴音信息数据作数 / 模转换。

虽然对本发明作了详细说明，但是这些说明显然是示例性的，并非对本发明的限制，所以本发明的精神和范围只受所附权利要求书的限制.

# 说 明 书 附 图

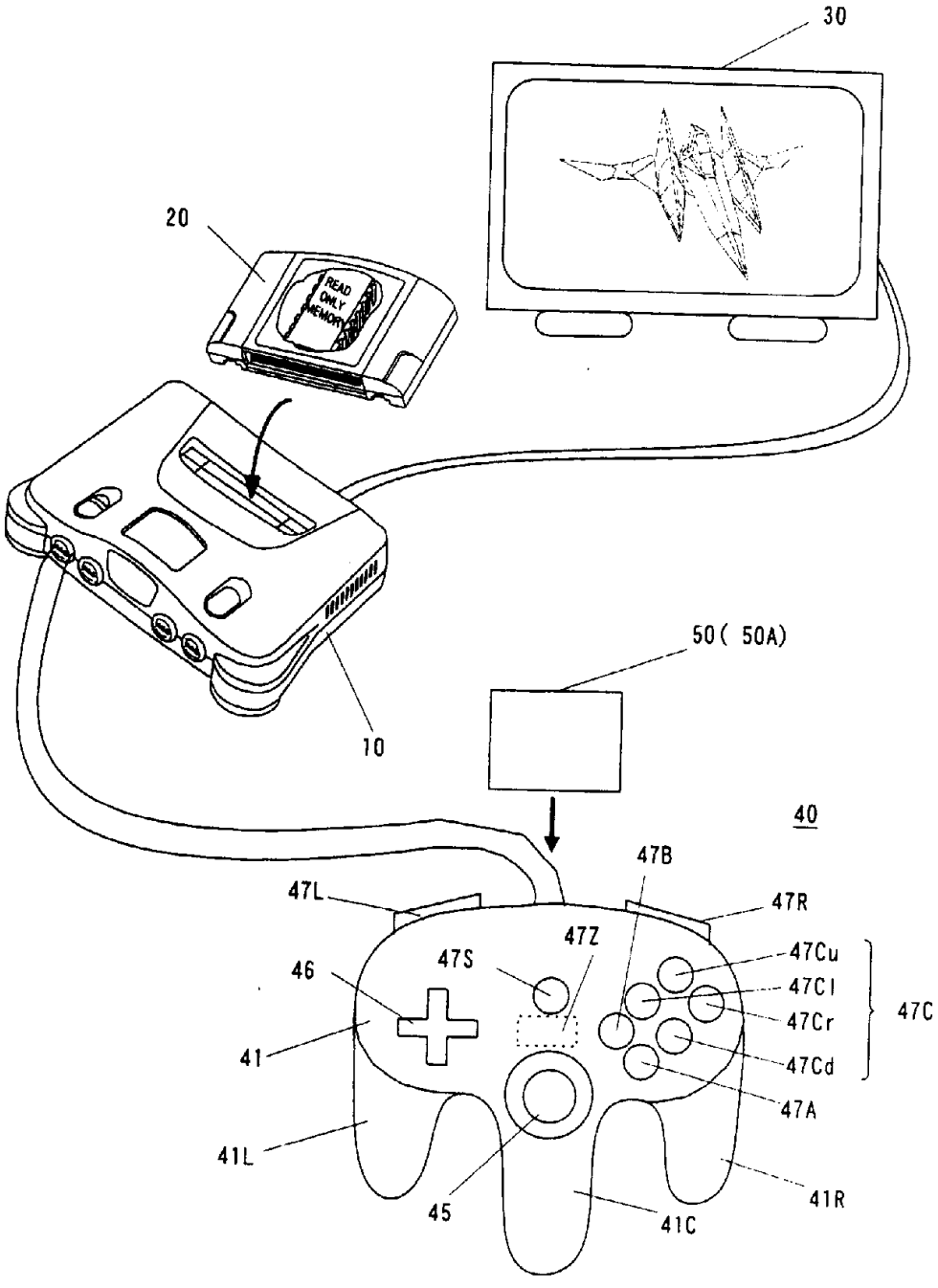

30
20
READ ONLY MEMORY
10
50( 50A)
40
47B
47L
47R
47Z
47S
46
47Cu
47Cl
47Cr
47Cd
47C
47A
41
41L
45
41C
41R

图 1

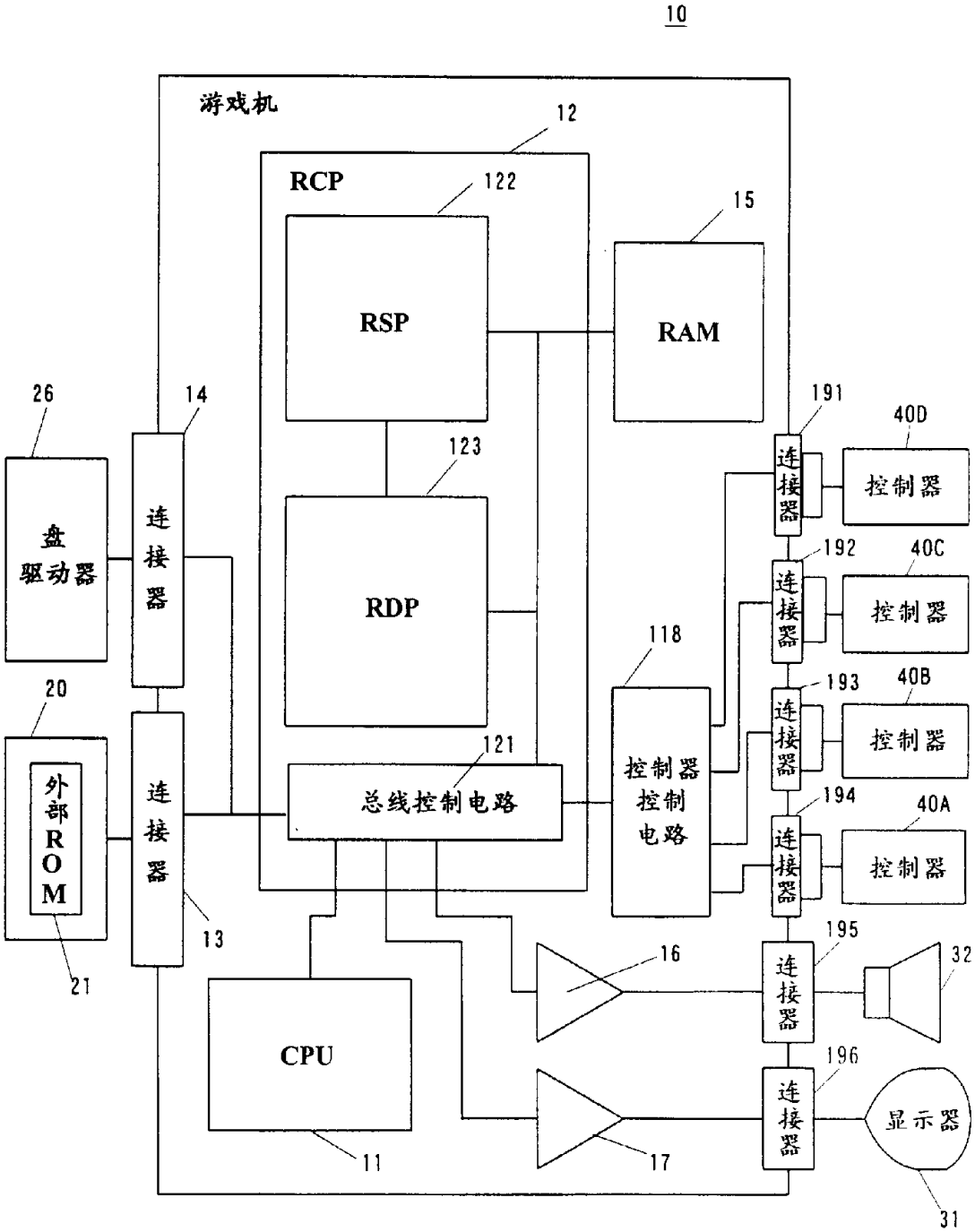

10
游戏机
12
RCP
122
RSP
15
RAM
123
RDP
121
总线控制电路
118
控制器控制电路
26
盘驱动器
14
连接器
20
外部ROM
21
13
CPU
11
16
17
191
192
193
194
195
196
连接器
40D
40C
40B
40A
控制器
32
显示器
31

图 2

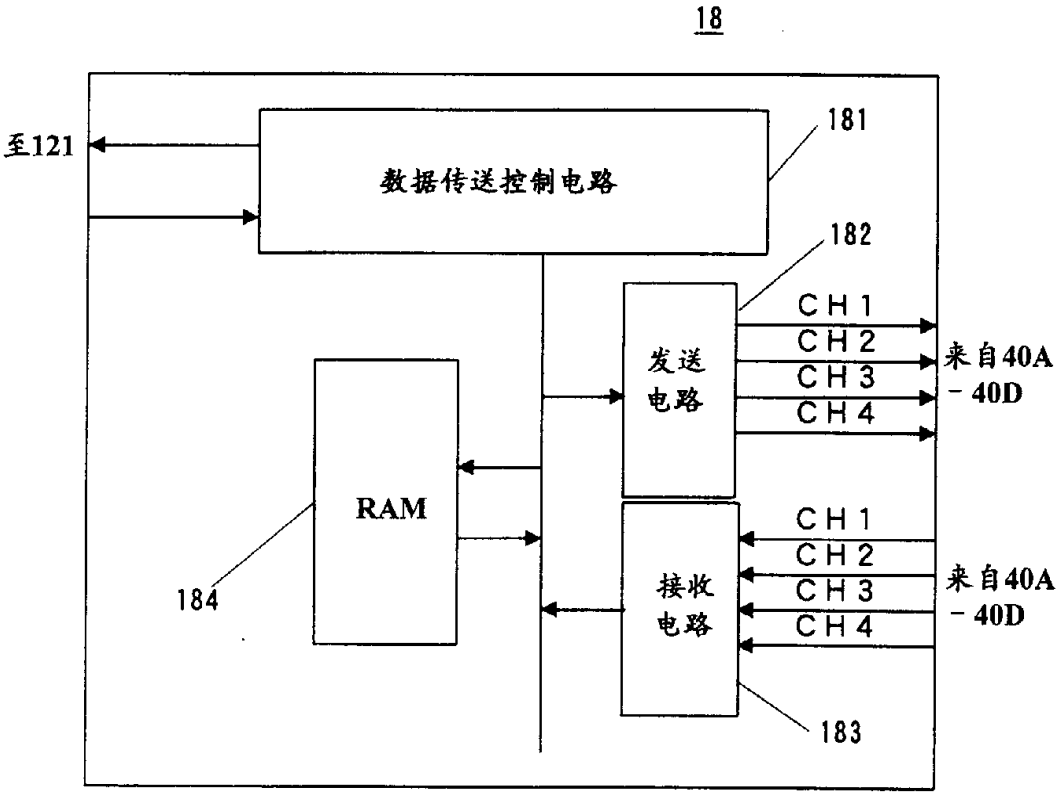

18
至121
数据传送控制电路
181
182
发送
电路
CH1
CH2
CH3
CH4
来自40A
－40D
RAM
184
接收
电路
CH1
CH2
CH3
CH4
来自40A
－40D
183

图 3

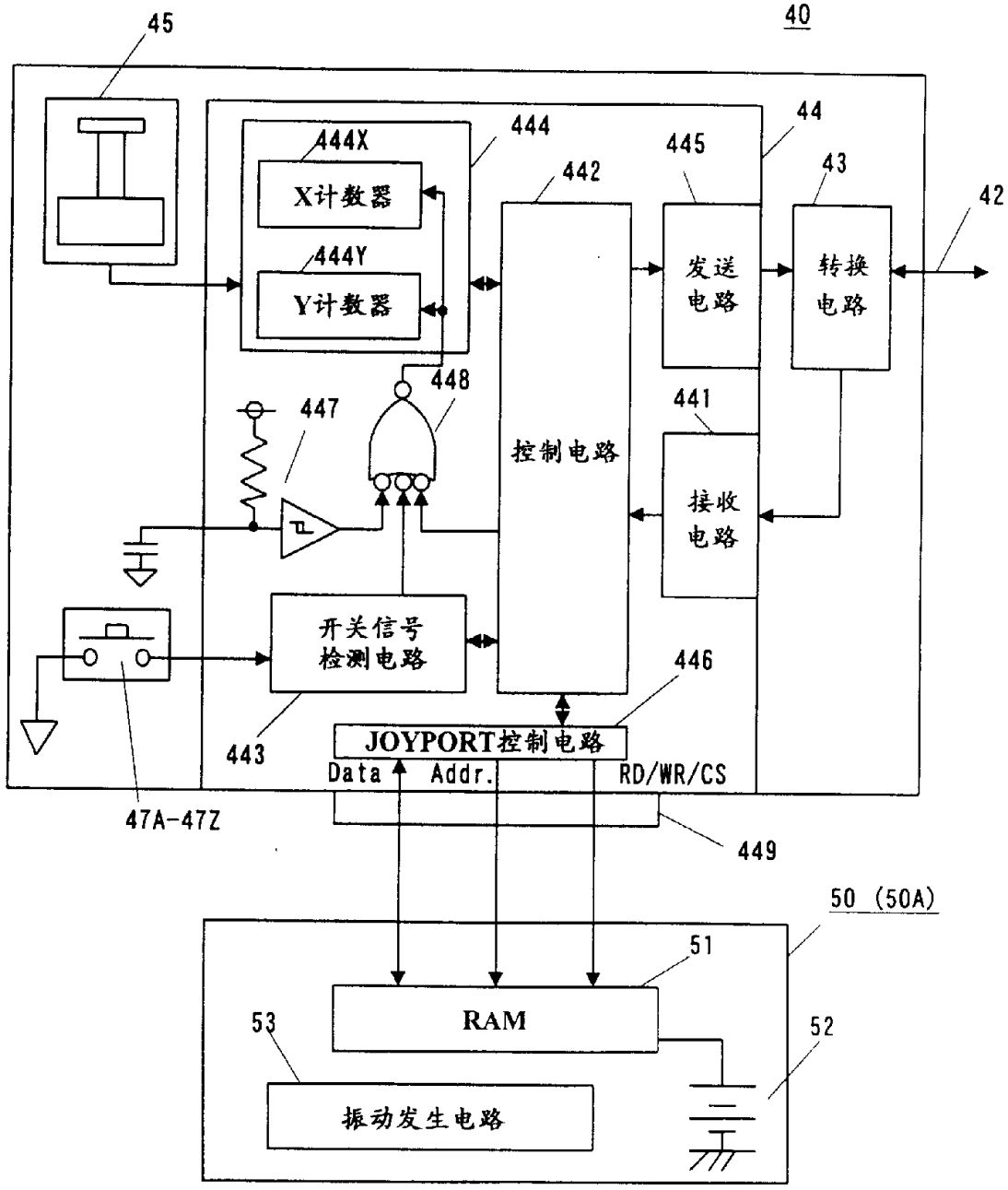

45
40
44
444
444X
X计数器
444Y
Y计数器
442
445
43
42
发送
电路
转换
电路
448
447
441
控制电路
接收
电路
开关信号
检测电路
446
443
JOYPORT控制电路
Data
Addr.
RD/WR/CS
47A-47Z
449
50 (50A)
51
RAM
53
52
振动发生电路

图 4

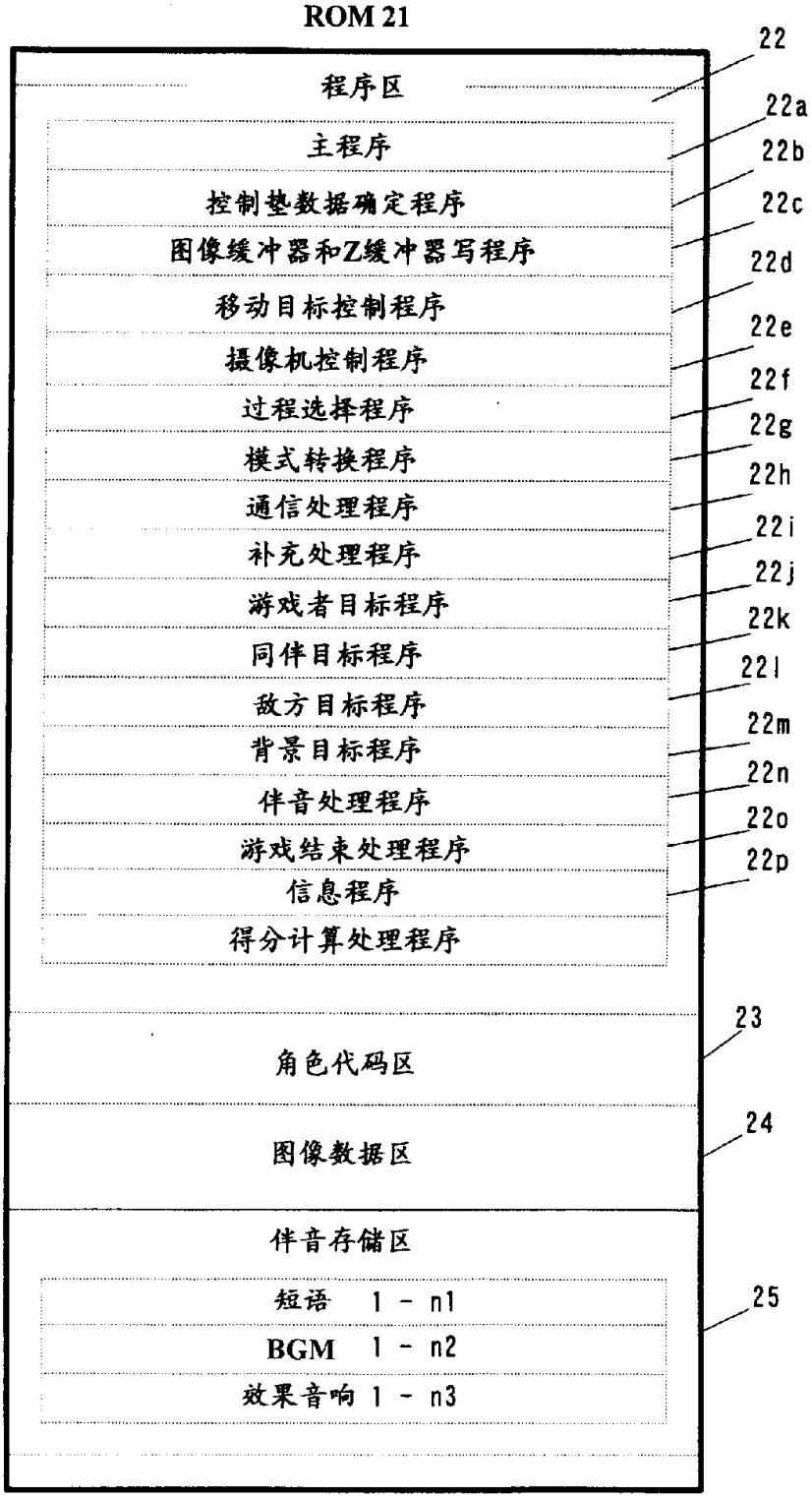

ROM 21
22
程序区
22a
主程序
22b
控制垫数据确定程序
22c
图像缓冲器和Z缓冲器写程序
22d
移动目标控制程序
22e
摄像机控制程序
22f
过程选择程序
22g
模式转换程序
22h
通信处理程序
22i
补充处理程序
22j
游戏者目标程序
22k
同伴目标程序
22l
敌方目标程序
22m
背景目标程序
22n
伴音处理程序
22o
游戏结束处理程序
22p
信息程序
得分计算处理程序
23
角色代码区
24
图像数据区
伴音存储区
短语 1 - n1
BGM 1 - n2
效果音响 1 - n3
25

图 5

图像数据区24的详细情况

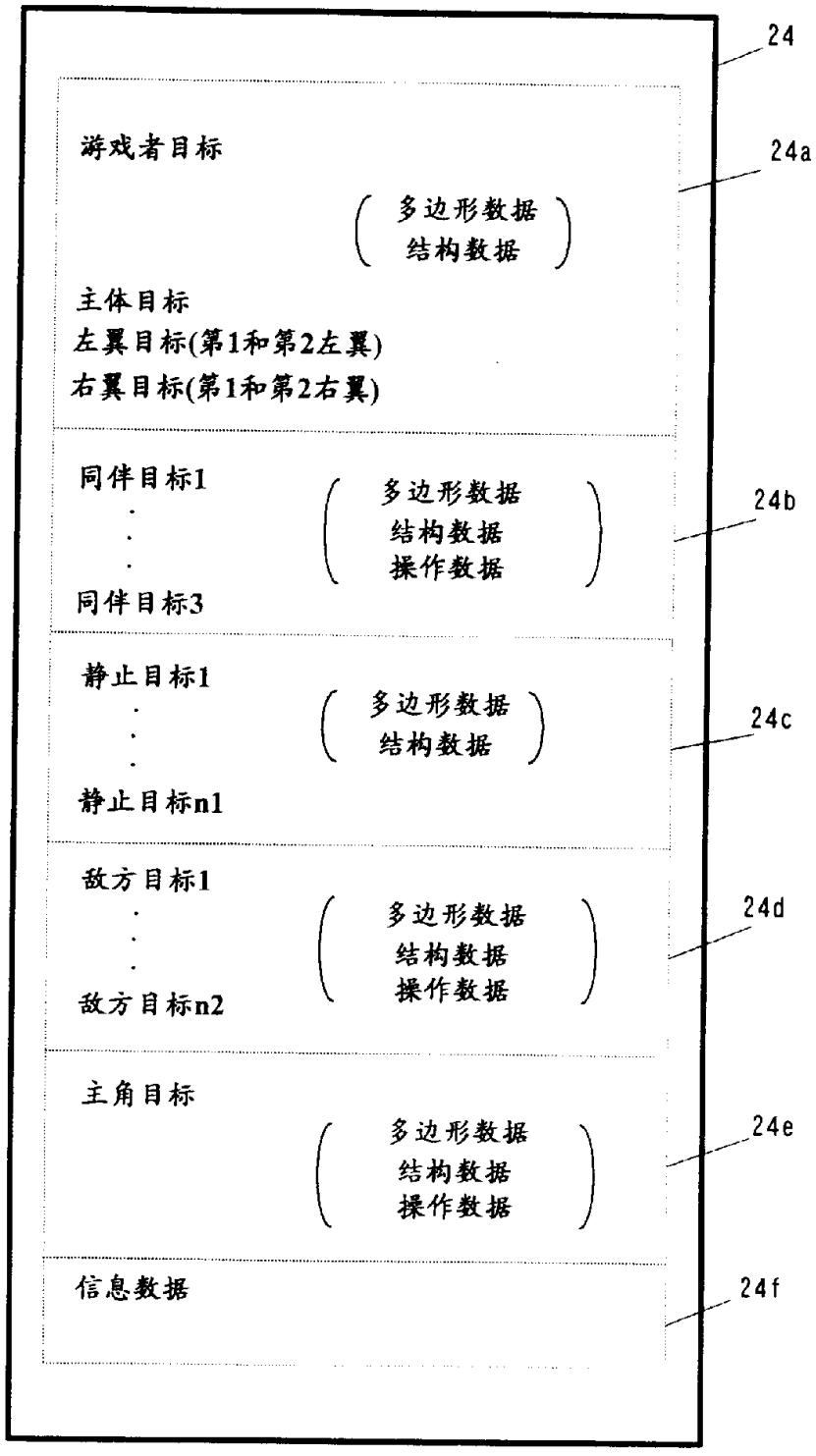



图 6

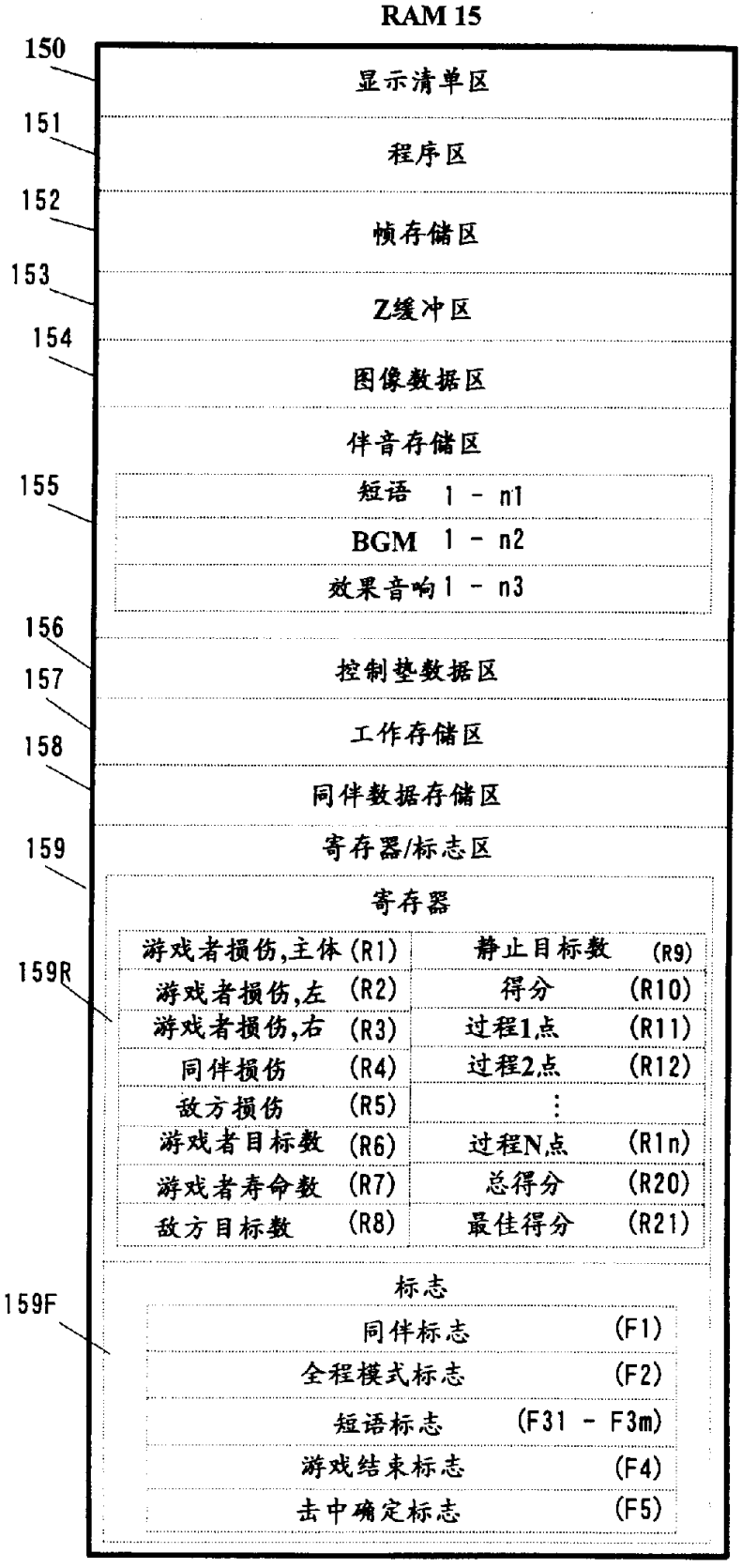

RAM 15
150
显示清单区
151
程序区
152
帧存储区
153
Z缓冲区
154
图像数据区
伴音存储区
155
短语 1 - n1
BGM 1 - n2
效果音响 1 - n3
156
控制垫数据区
157
工作存储区
158
同伴数据存储区
159
寄存器/标志区
寄存器
159R
游戏者损伤,主体 (R1)
游戏者损伤,左 (R2)
游戏者损伤,右 (R3)
同伴损伤 (R4)
敌方损伤 (R5)
游戏者目标数 (R6)
游戏者寿命数 (R7)
敌方目标数 (R8)
静止目标数 (R9)
得分 (R10)
过程1点 (R11)
过程2点 (R12)
过程N点 (R1n)
总得分 (R20)
最佳得分 (R21)
标志
159F
同伴标志 (F1)
全程模式标志 (F2)
短语标志 (F31 - F3m)
游戏结束标志 (F4)
击中确定标志 (F5)

图 7

图像数据区154的详细情况

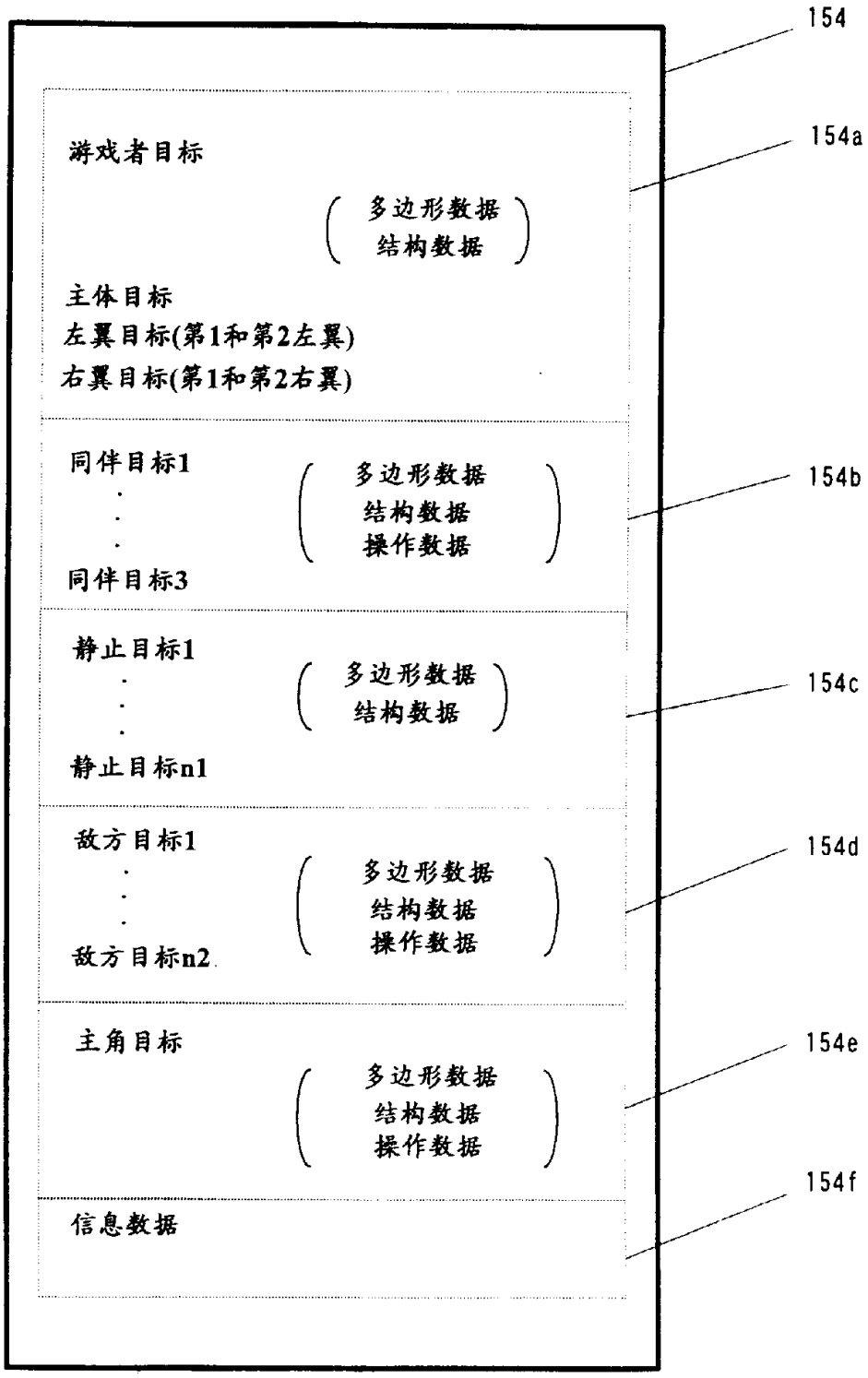



图 8

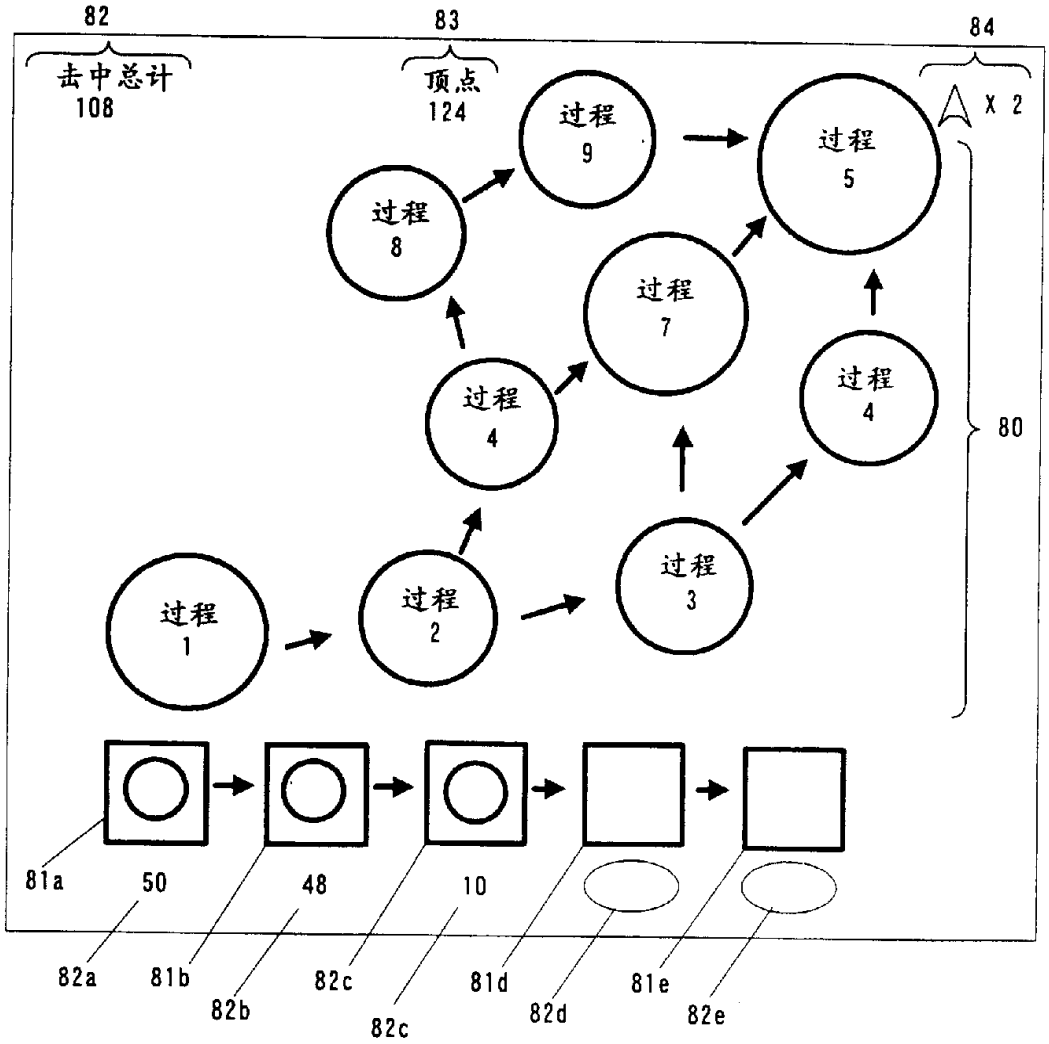

82
击中总计
108
83
顶点
124
84
X 2
过程
1
过程
2
过程
3
过程
4
过程
4
过程
5
过程
7
过程
8
过程
9
80
81a
50
82a
81b
48
82b
82c
10
82c
81d
82d
81e
82e

图 9

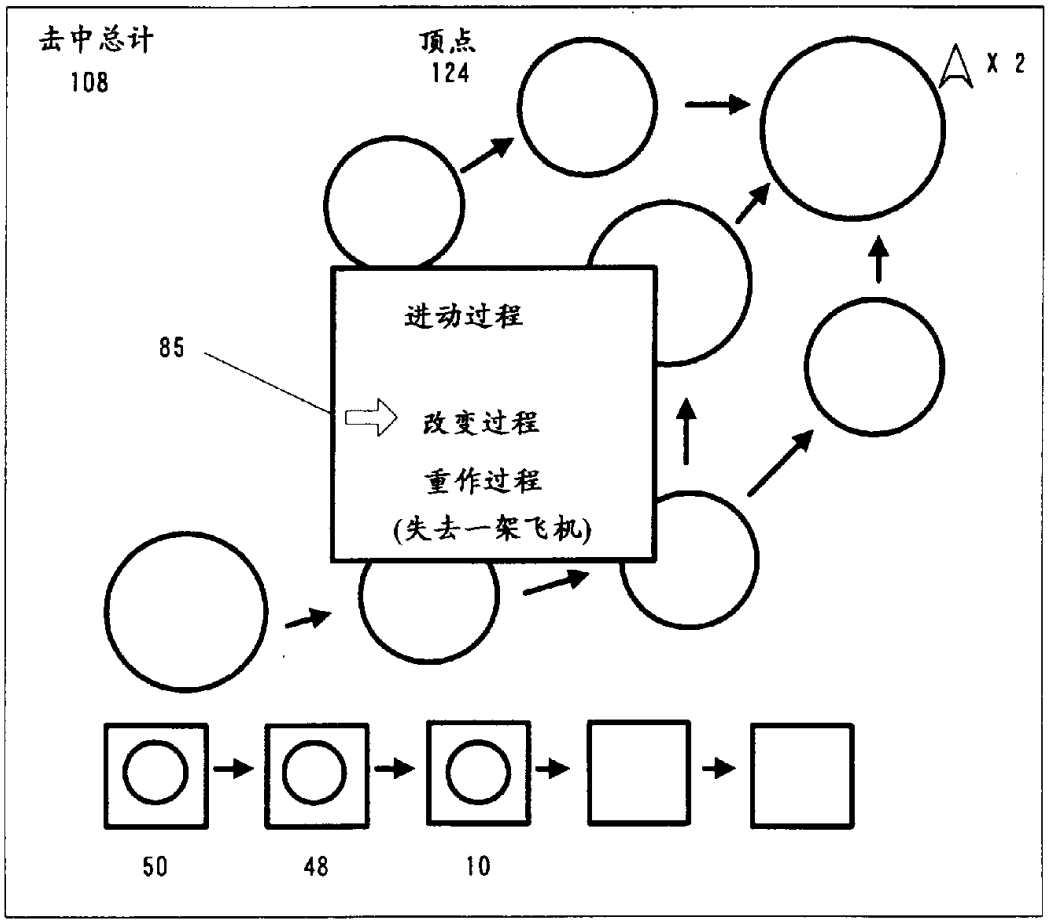



图 10

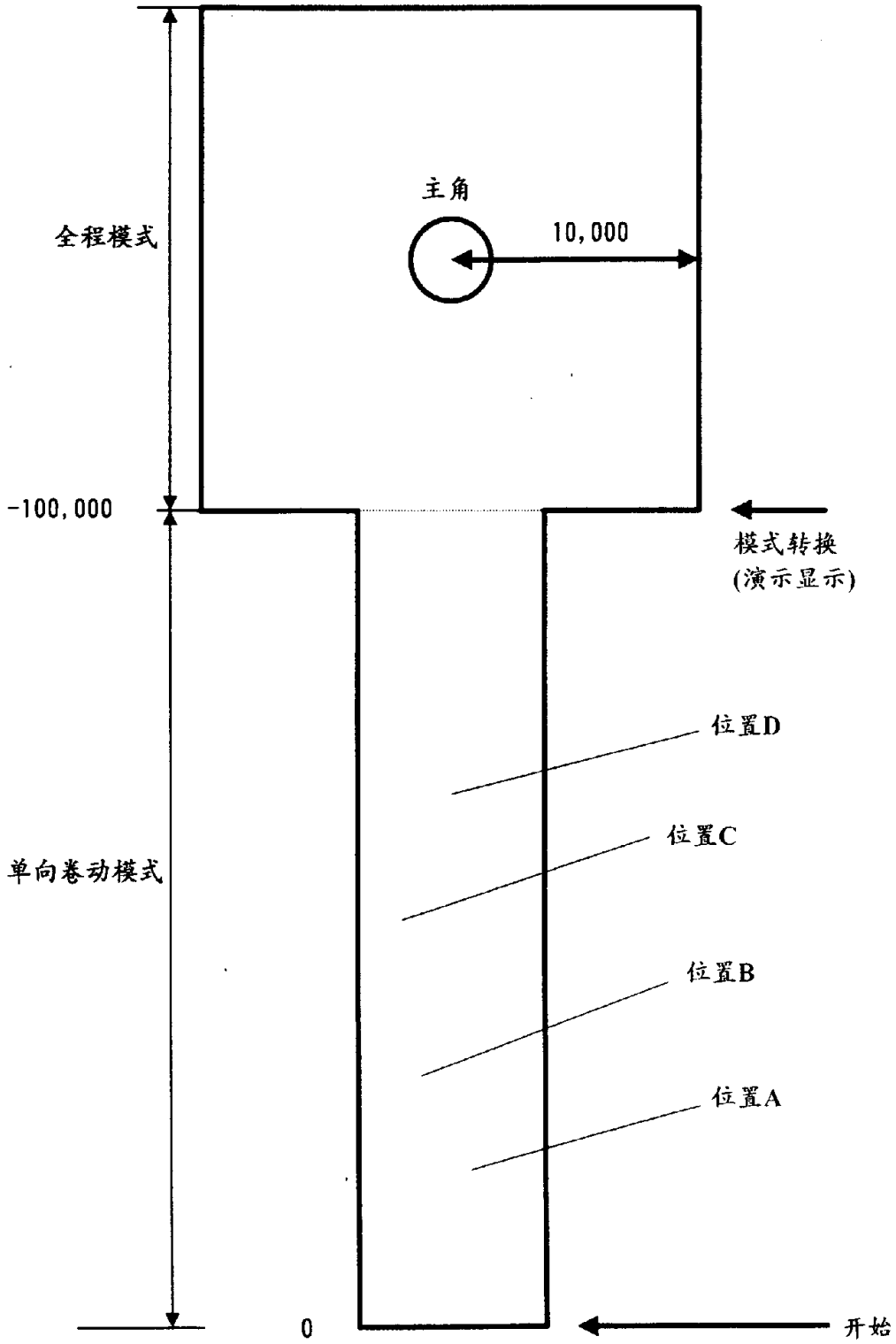



图 11

98.05.05

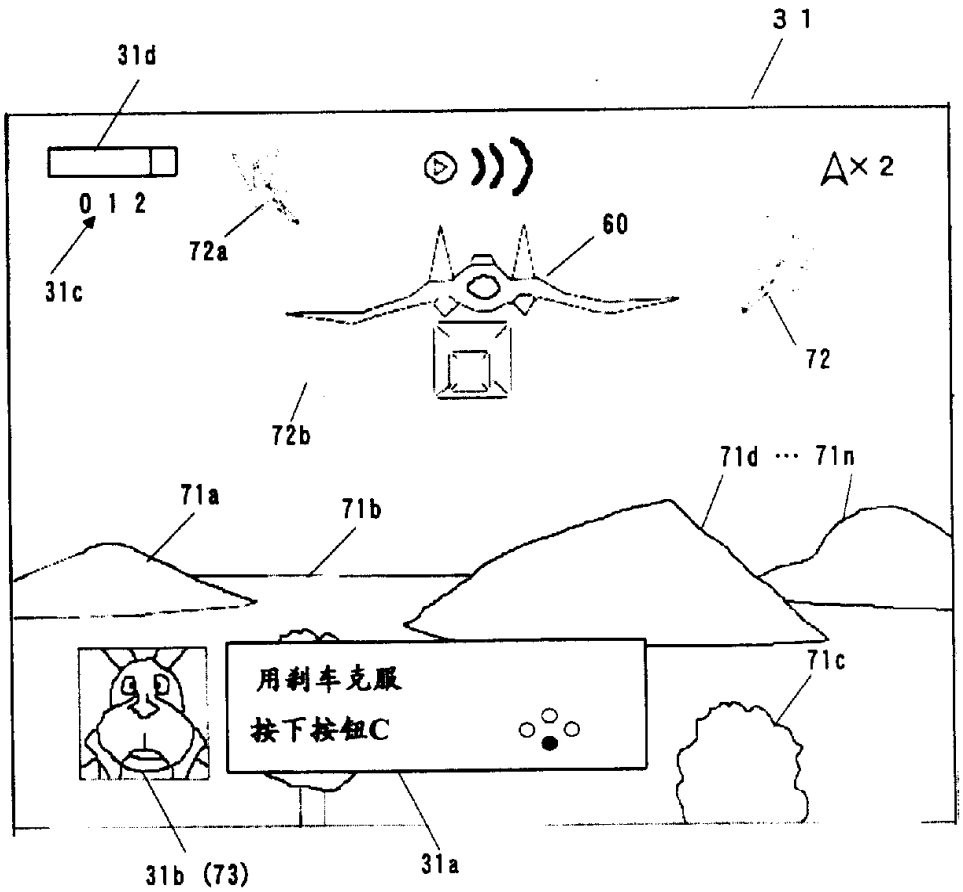

31
31d
0 1 2
31c
72a
60
A× 2
72
72b
71d … 71n
71a
71b
71c
用刹车克服
按下按钮C
31b (73)
31a

图 12

| | 人 | 短语 | 条件 | 优先级 |
|---|---|---|---|---|
| 短语 1 | 同伴 1 | 用刹车克服,C按钮向下 | 位置A | 1 |
| 短语 2 | 同伴 2 | 提速通过,C按钮向左 | 位置B | 2 |
| 短语 3 | 同伴 3 | 瞄准背后 | 寻找主角的时间A | 2 |
| 短语 4 | 同伴 4 | 用滚动抵抗,按两次Z或R | 位置C | 1 |
| 短语 5 | 同伴 5 | 帮助 | 敌方瞄准 | 1 |
| 短语 6 | 同伴 6 | 我们被监视了 | 营救 | 1 |
| 短语 7 | 同伴 3 | 机智地使用炸弹 | 位置D | 1 |
| 短语 8 | 英雄 | UWAHHHH…… | Bd ≦ 0 | 3 |
| 短语 9 | 主角 | GWAHHHH…… | 敌人的 Ed ≦ 0 | 0 |

图 13

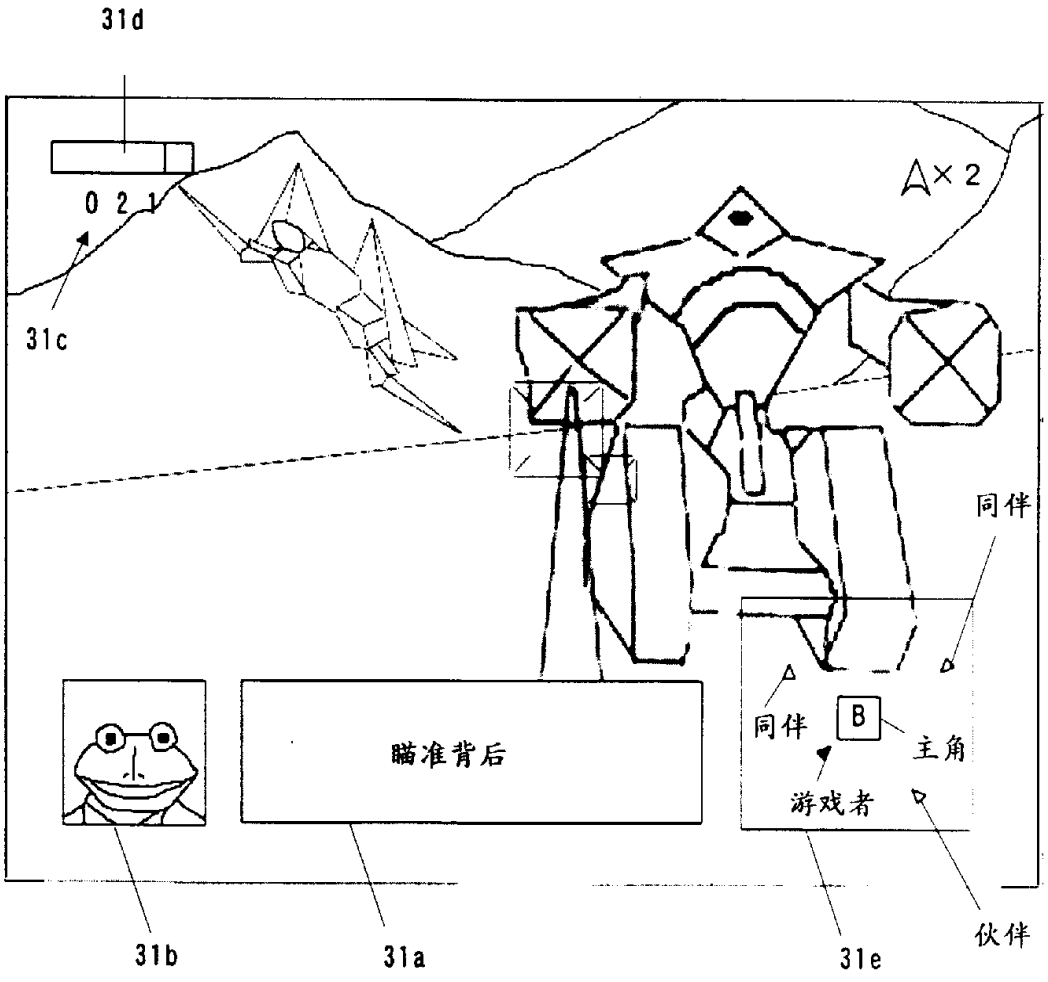



图 14

整套流程图

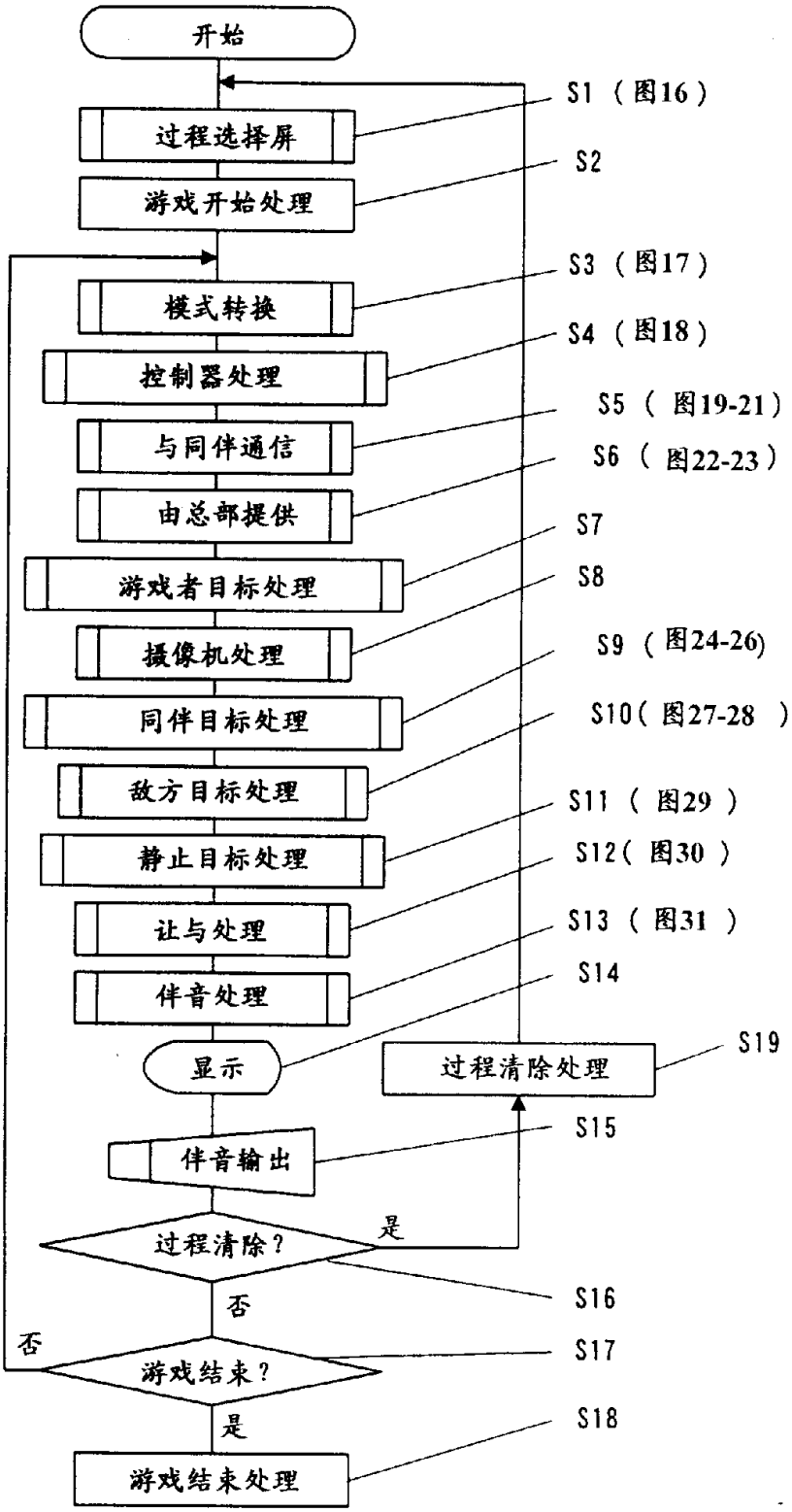

开始
过程选择屏
S1（图16）
游戏开始处理
S2
模式转换
S3（图17）
控制器处理
S4（图18）
与同伴通信
S5（图19-21）
由总部提供
S6（图22-23）
游戏者目标处理
S7
摄像机处理
S8
同伴目标处理
S9（图24-26）
敌方目标处理
S10（图27-28）
静止目标处理
S11（图29）
让与处理
S12（图30）
伴音处理
S13（图31）
显示
S14
过程清除处理
S19
伴音输出
S15
过程清除？
是
否
S16
游戏结束？
否
S17
是
游戏结束处理
S18

图 15

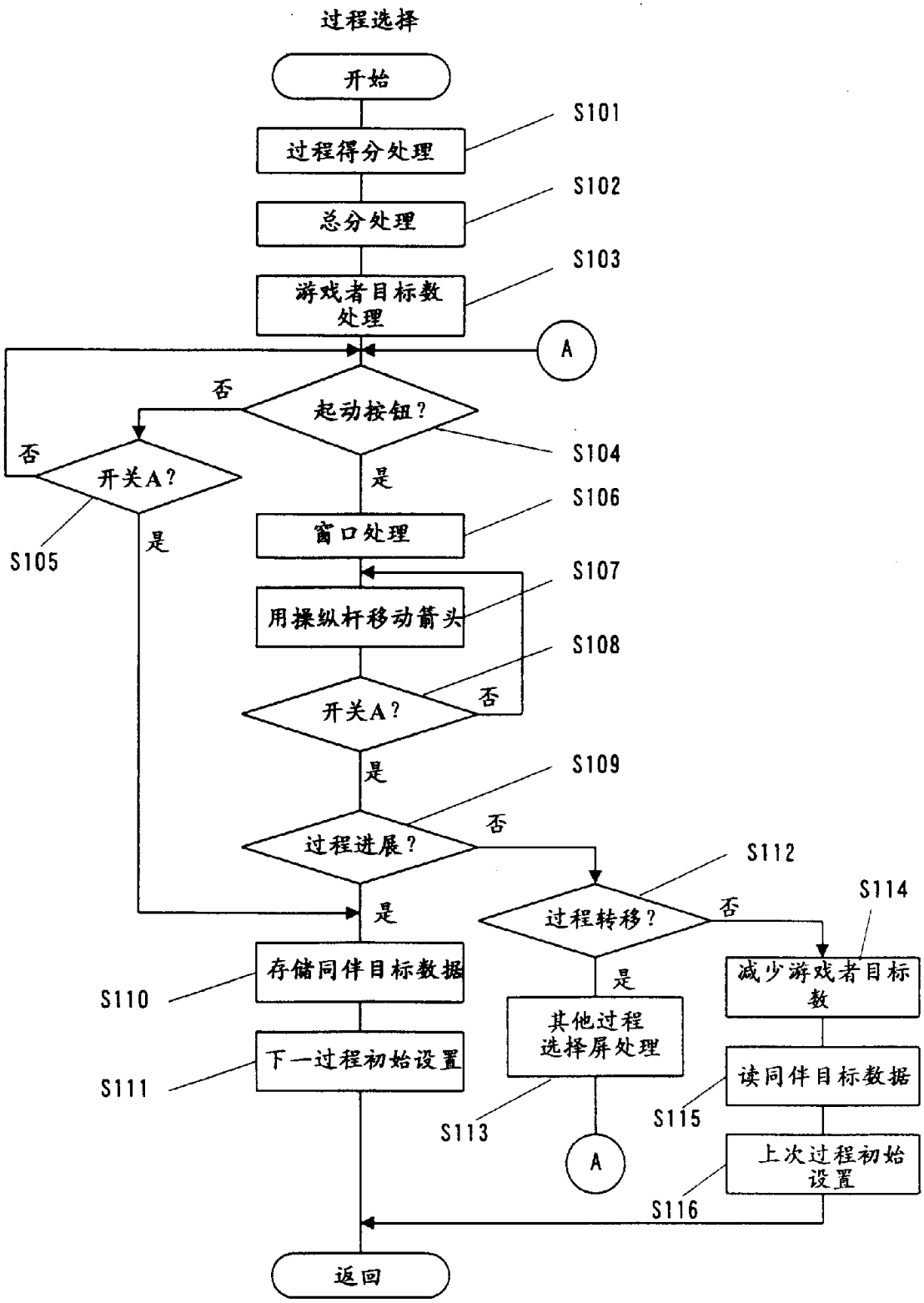

过程选择
开始
S101
过程得分处理
S102
总分处理
S103
游戏者目标数处理
A
否
起动按钮?
S104
否
开关A?
是
S105
是
S106
窗口处理
S107
用操纵杆移动箭头
S108
开关A?
否
是
S109
过程进展?
否
S112
过程转移?
否
S114
减少游戏者目标数
是
其他过程选择屏处理
S113
A
读同伴目标数据
S115
上次过程初始设置
S116
是
存储同伴目标数据
S110
下一过程初始设置
S111
返回

图 16

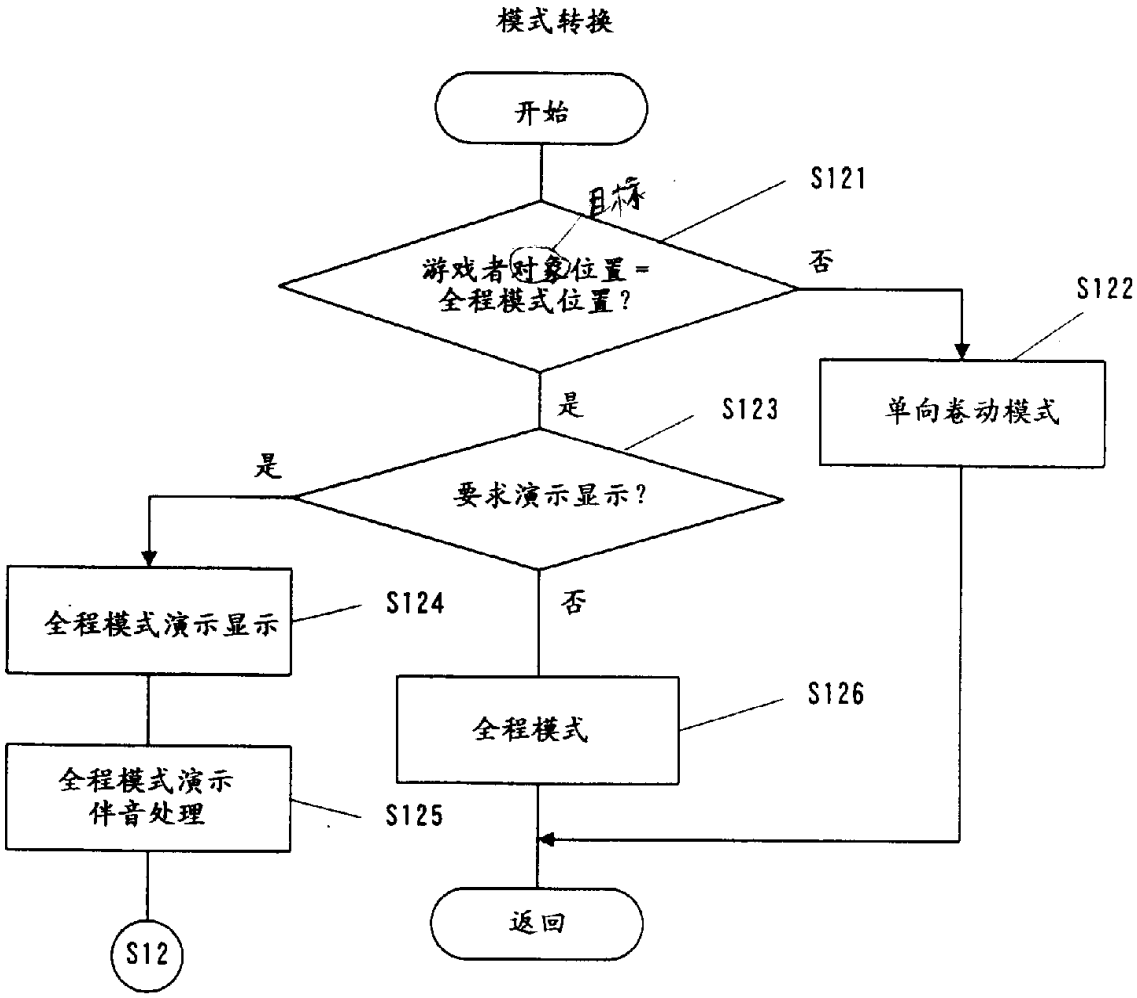

模式转换
开始
S121
游戏者对象位置＝
全程模式位置？
否
S122
单向卷动模式
是
S123
要求演示显示？
是
否
S124
全程模式演示显示
S125
全程模式演示
伴音处理
S126
全程模式
返回
S12

图 17

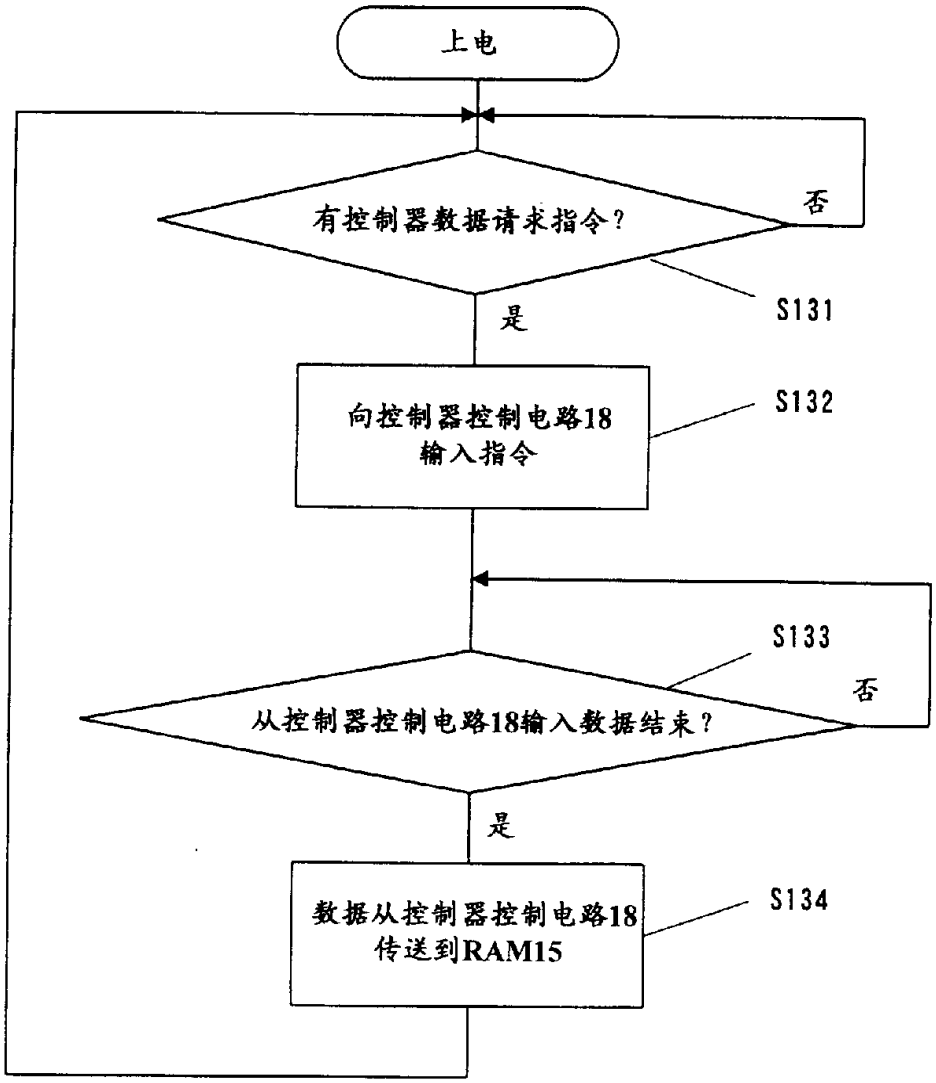

上电
有控制器数据请求指令？
否
S131
是
向控制器控制电路18
输入指令
S132
S133
否
从控制器控制电路18输入数据结束？
是
数据从控制器控制电路18
传送到RAM15
S134

图 18

与同伴通信1

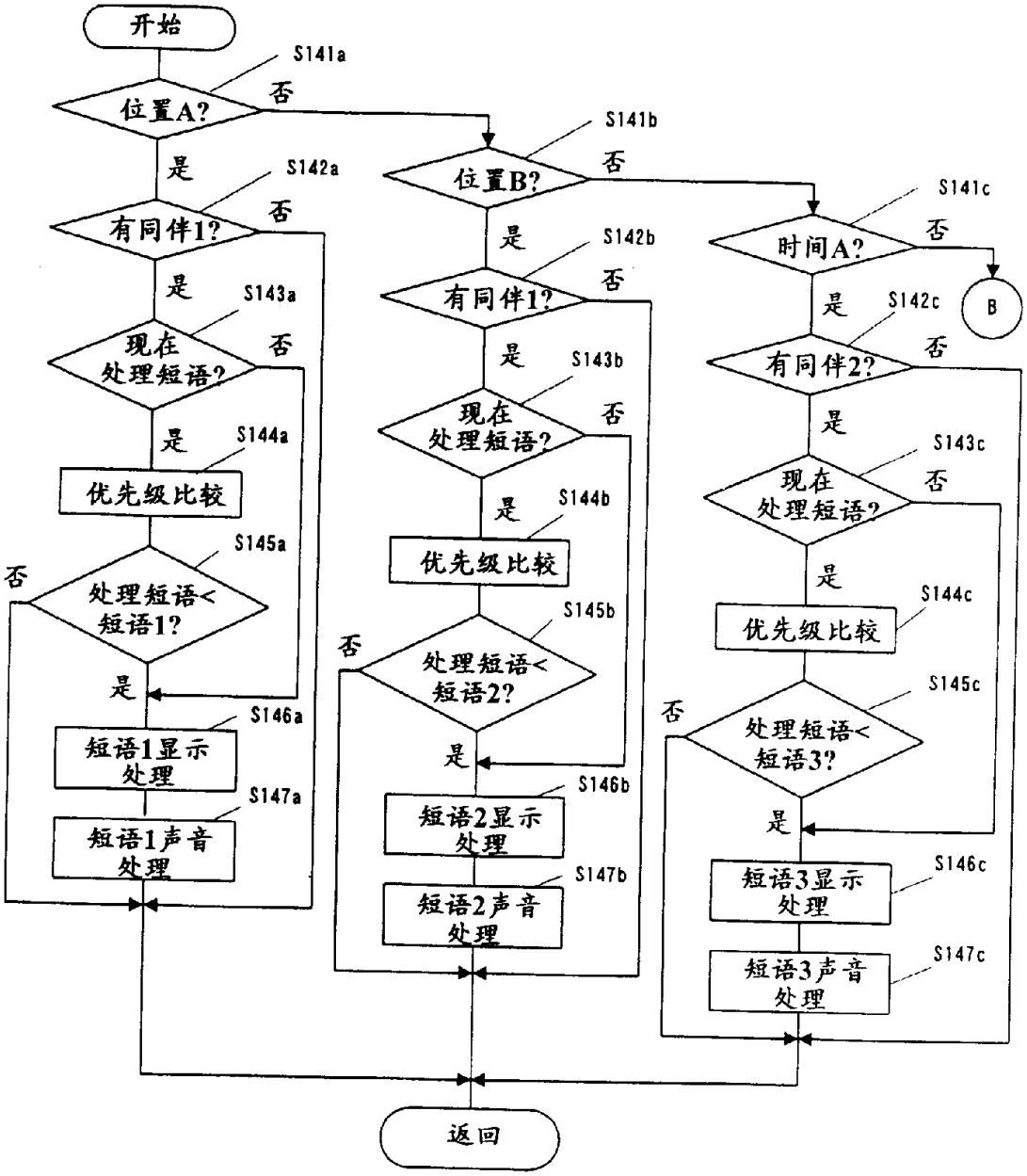

开始
S141a
位置A?
否
是
S142a
有同伴1?
否
是
S143a
现在处理短语?
否
是
S144a
优先级比较
S145a
处理短语<短语1?
否
是
S146a
短语1显示处理
S147a
短语1声音处理
S141b
位置B?
否
是
S142b
有同伴1?
否
是
S143b
现在处理短语?
否
是
S144b
优先级比较
S145b
处理短语<短语2?
否
是
S146b
短语2显示处理
S147b
短语2声音处理
S141c
时间A?
否
是
B
S142c
有同伴2?
否
是
S143c
现在处理短语?
否
是
S144c
优先级比较
S145c
处理短语<短语3?
否
是
S146c
短语3显示处理
S147c
短语3声音处理
返回

图 19

与同伴通信2

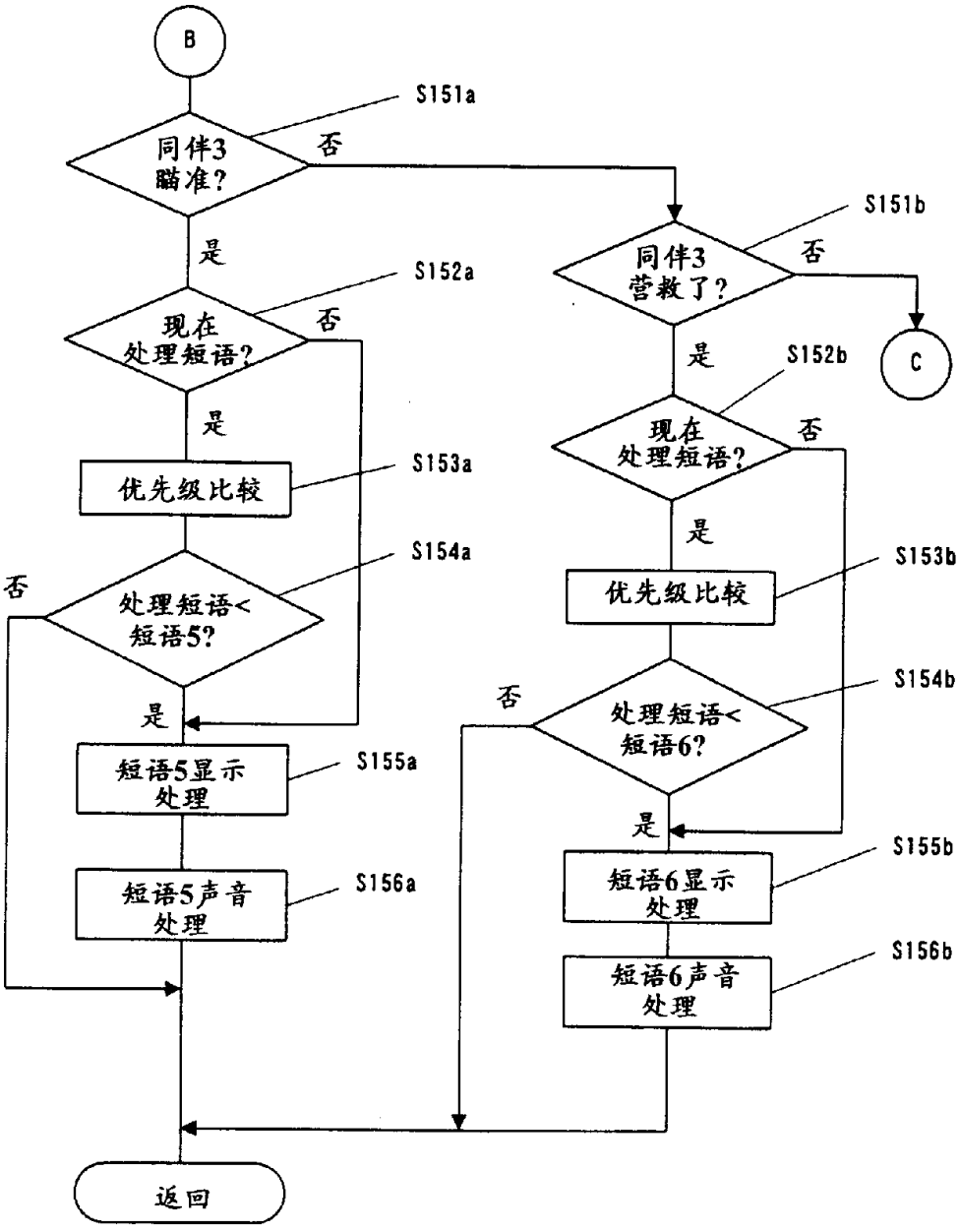

B
S151a
同伴3
瞄准?
否
是
S152a
现在
处理短语?
否
是
S153a
优先级比较
S154a
处理短语<
短语5?
否
是
S155a
短语5显示
处理
S156a
短语5声音
处理
返回
S151b
同伴3
营救了?
否
是
C
S152b
现在
处理短语?
否
是
S153b
优先级比较
S154b
处理短语<
短语6?
否
是
S155b
短语6显示
处理
S156b
短语6声音
处理

图 20

与同伴通信3

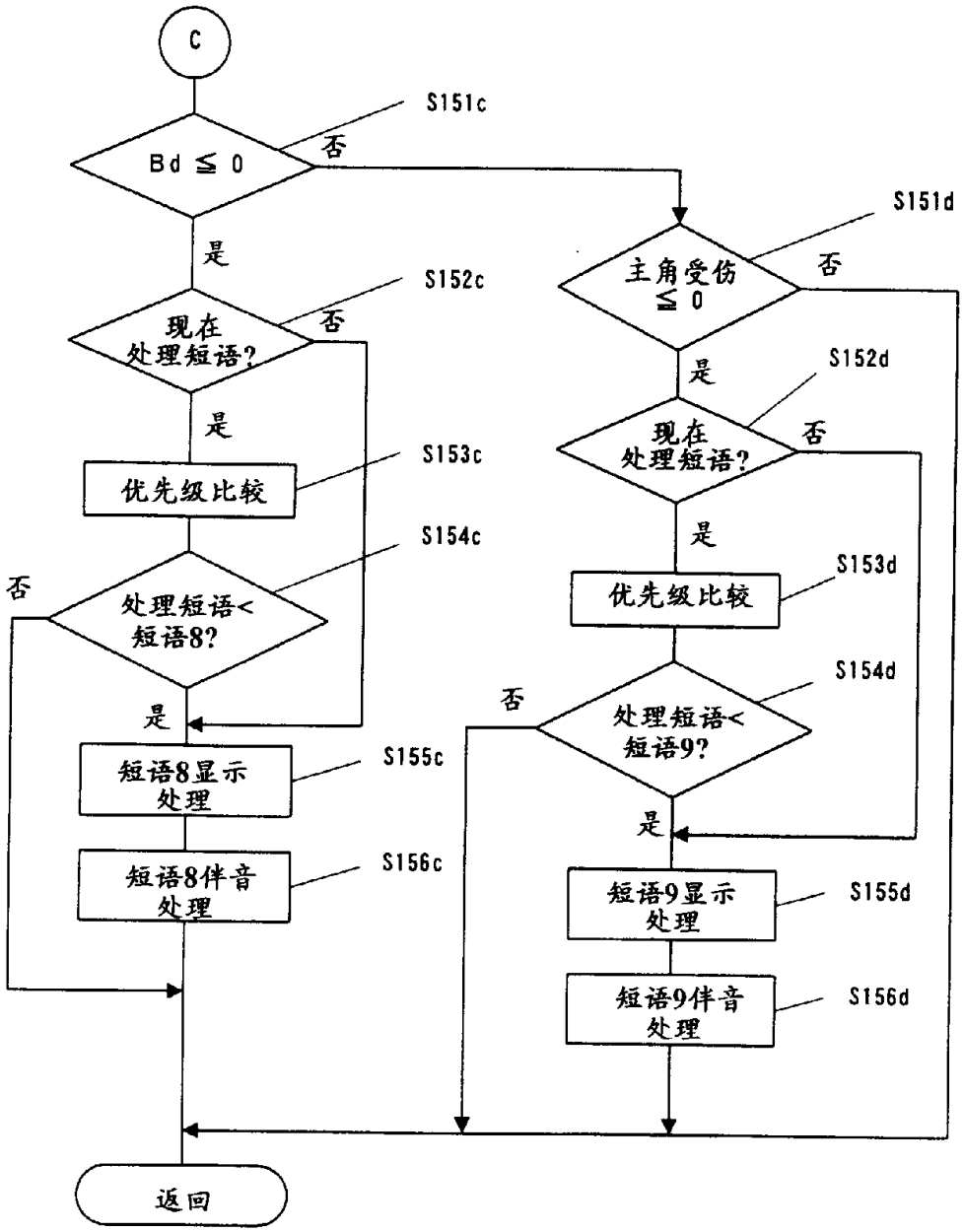

C
S151c
Bd ≦ 0
否
是
S152c
现在
处理短语?
否
是
S153c
优先级比较
S154c
处理短语<
短语8?
否
是
S155c
短语8显示
处理
S156c
短语8伴音
处理
S151d
主角受伤
≦ 0
否
是
S152d
现在
处理短语?
否
是
S153d
优先级比较
S154d
处理短语<
短语9?
否
是
S155d
短语9显示
处理
S156d
短语9伴音
处理
返回

图 21

材料补充处理1

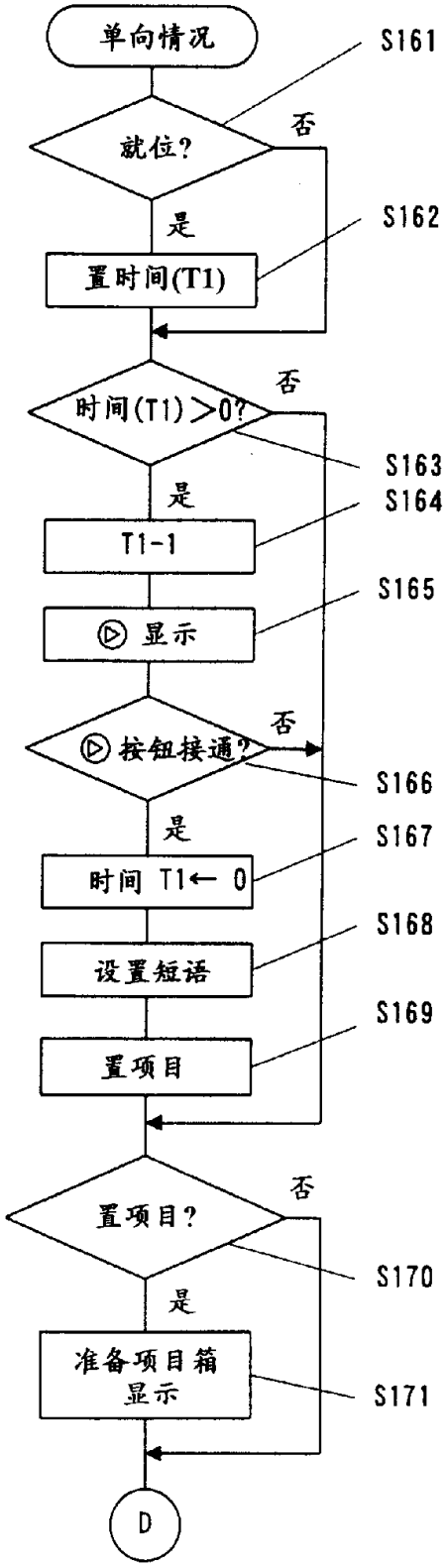

单向情况
S161
就位?
否
是
S162
置时间(T1)
时间(T1)＞0?
否
S163
是
S164
T1-1
S165
显示
按钮接通?
否
S166
是
S167
时间 T1← 0
S168
设置短语
S169
置项目
置项目?
否
S170
是
准备项目箱
显示
S171
D

图 22

材料补充处理2

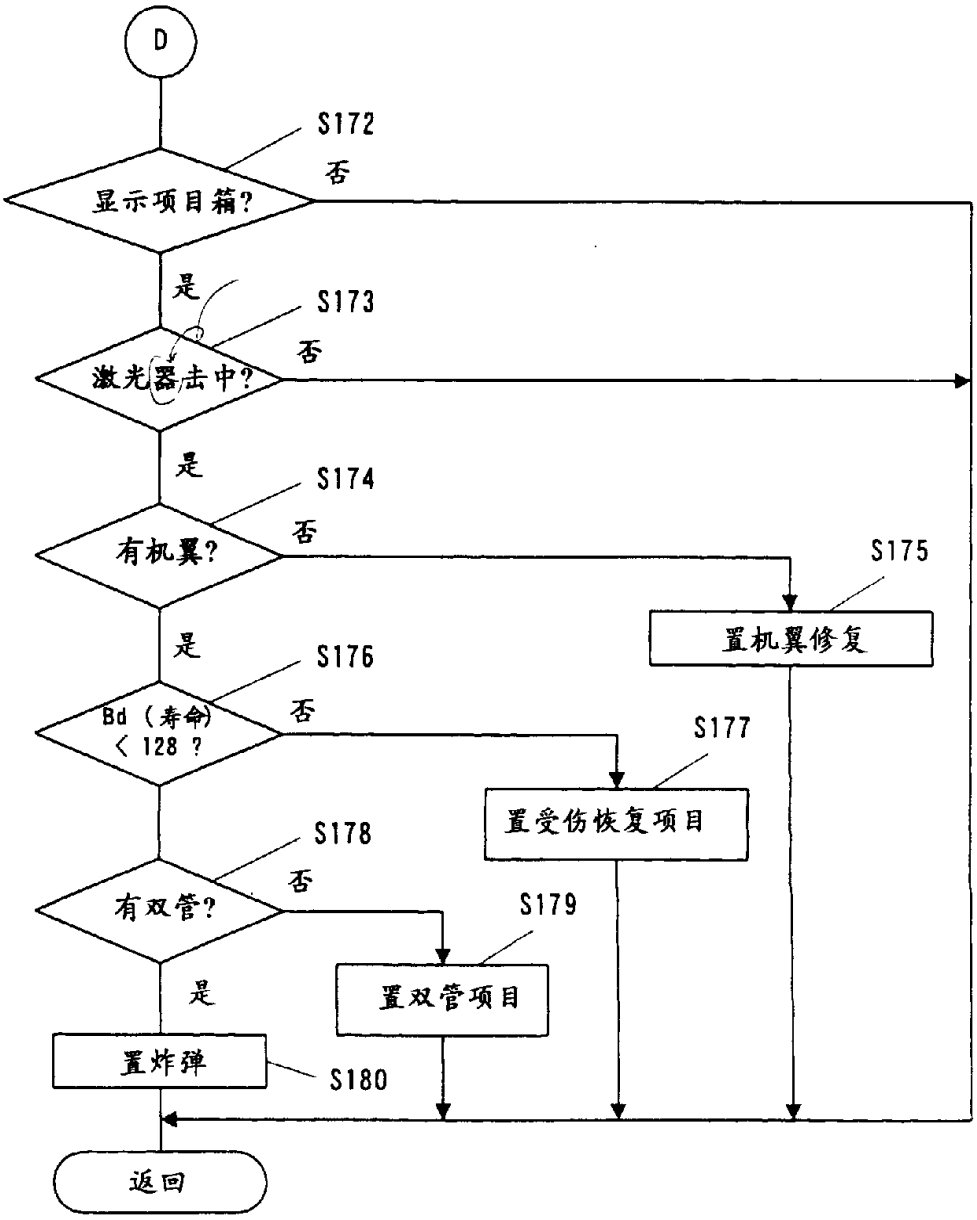

D
S172
显示项目箱?
否
是
S173
激光器击中?
否
是
S174
有机翼?
否
S175
置机翼修复
是
S176
Bd（寿命）
< 128 ?
否
S177
置受伤恢复项目
S178
有双管?
否
S179
置双管项目
是
置炸弹
S180
返回

图 23

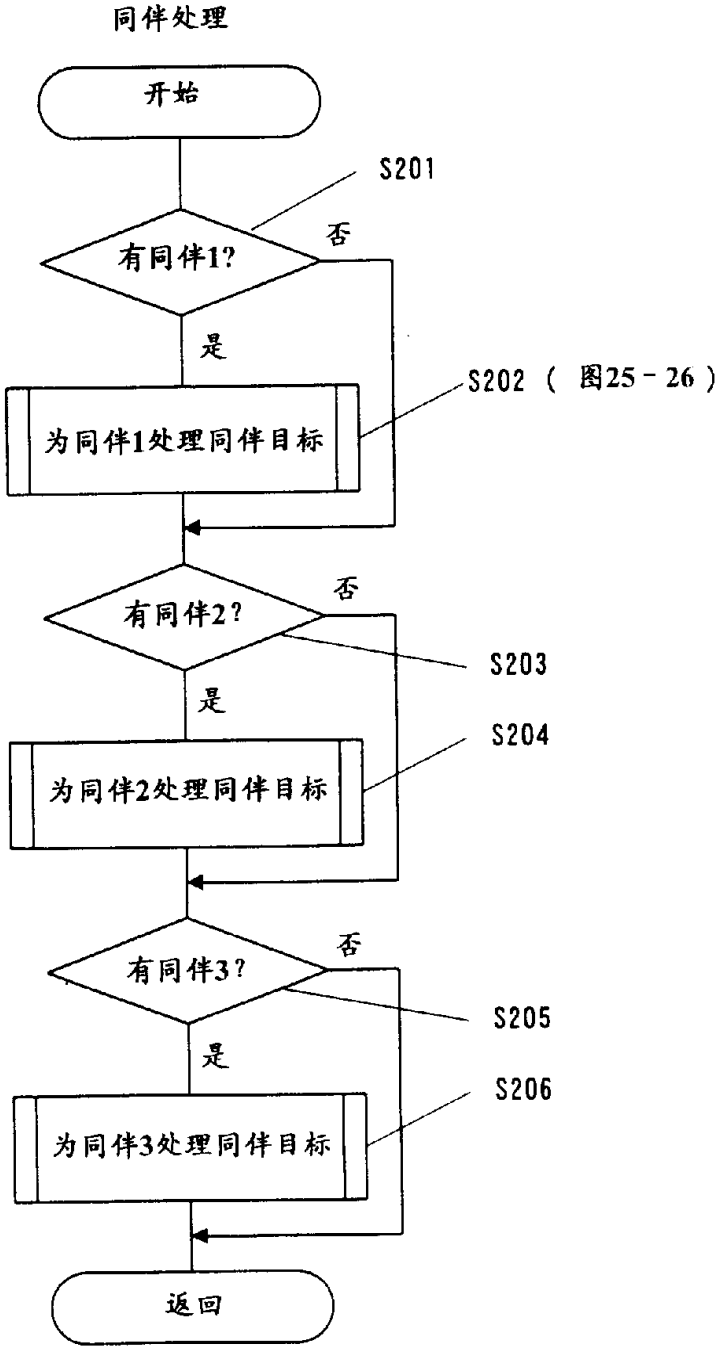

同伴处理
开始
S201
有同伴1?
否
是
S202 （ 图25－26 ）
为同伴1处理同伴目标
有同伴2?
否
S203
是
S204
为同伴2处理同伴目标
有同伴3?
否
S205
是
S206
为同伴3处理同伴目标
返回

图 24

同伴目标处理1

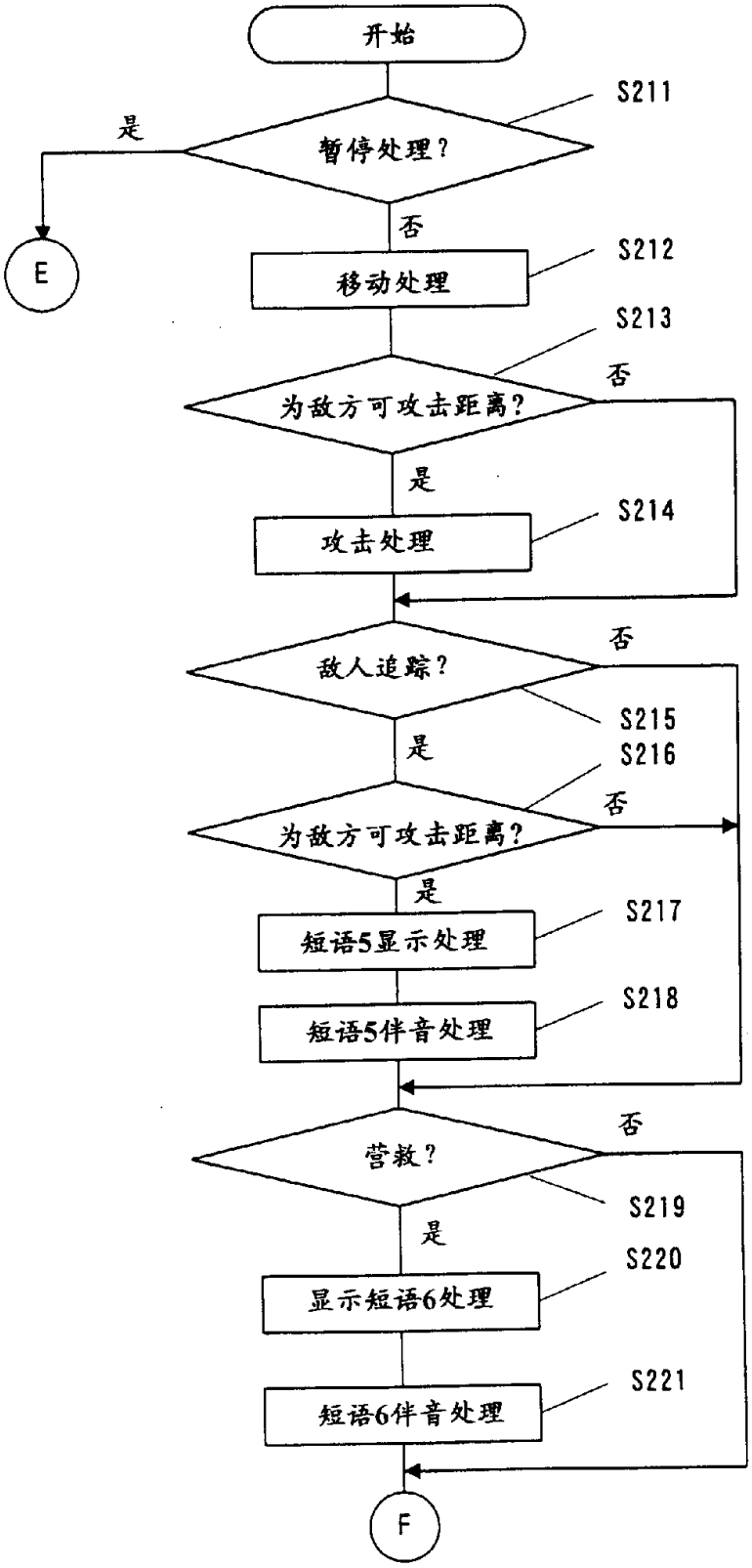

开始
S211
是
暂停处理?
否
E
S212
移动处理
S213
否
为敌方可攻击距离?
是
S214
攻击处理
否
敌人追踪?
S215
是
S216
否
为敌方可攻击距离?
是
S217
短语5显示处理
S218
短语5伴音处理
否
营救?
S219
是
S220
显示短语6处理
S221
短语6伴音处理
F

图 25

同伴目标处理2

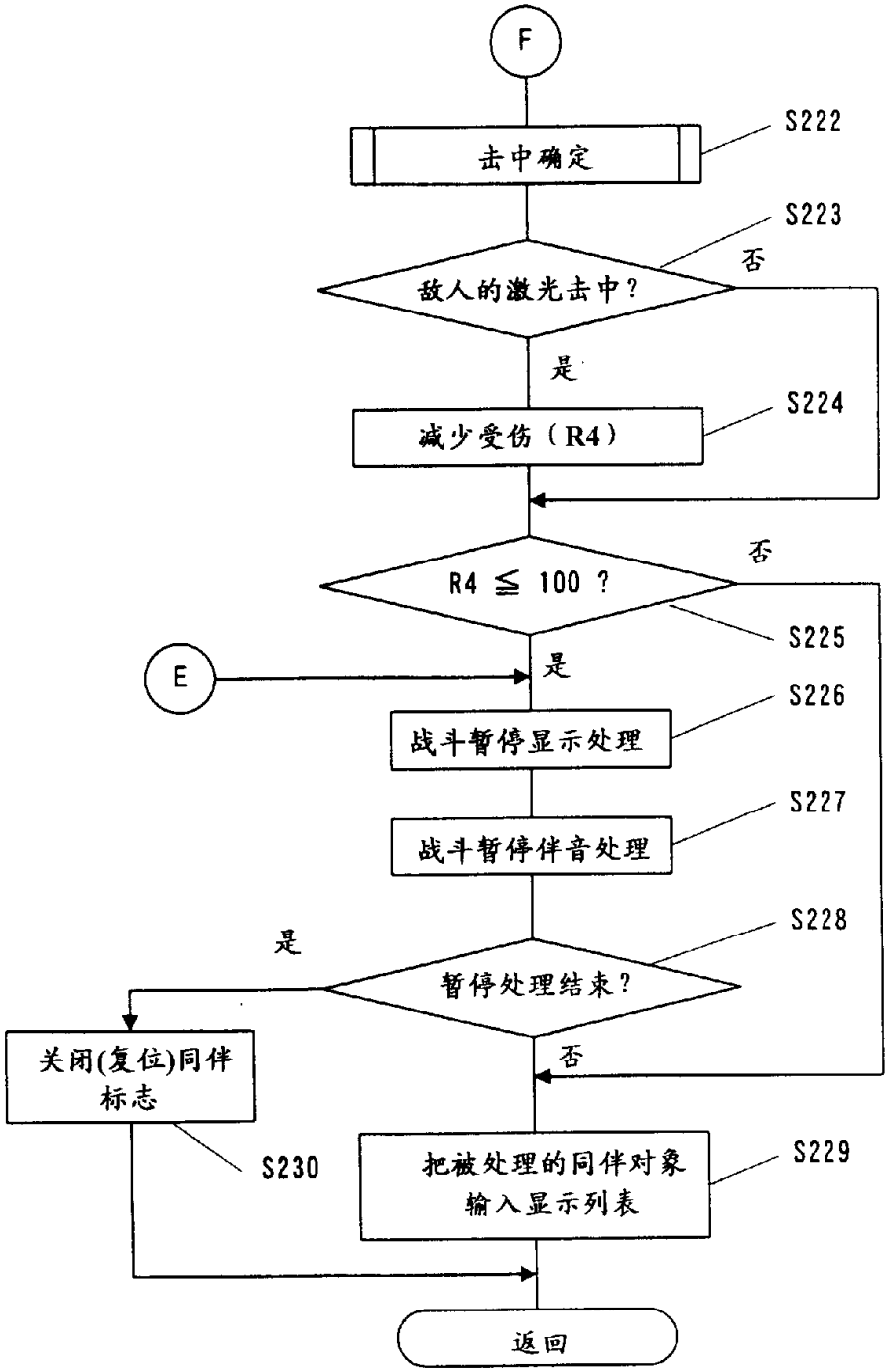

F
击中确定
S222
敌人的激光击中？
S223
否
是
减少受伤（R4）
S224
R4 ≦ 100 ？
否
S225
E
是
战斗暂停显示处理
S226
战斗暂停伴音处理
S227
暂停处理结束？
S228
是
否
关闭(复位)同伴标志
S230
把被处理的同伴对象输入显示列表
S229
返回

图 26

敌方目标处理1

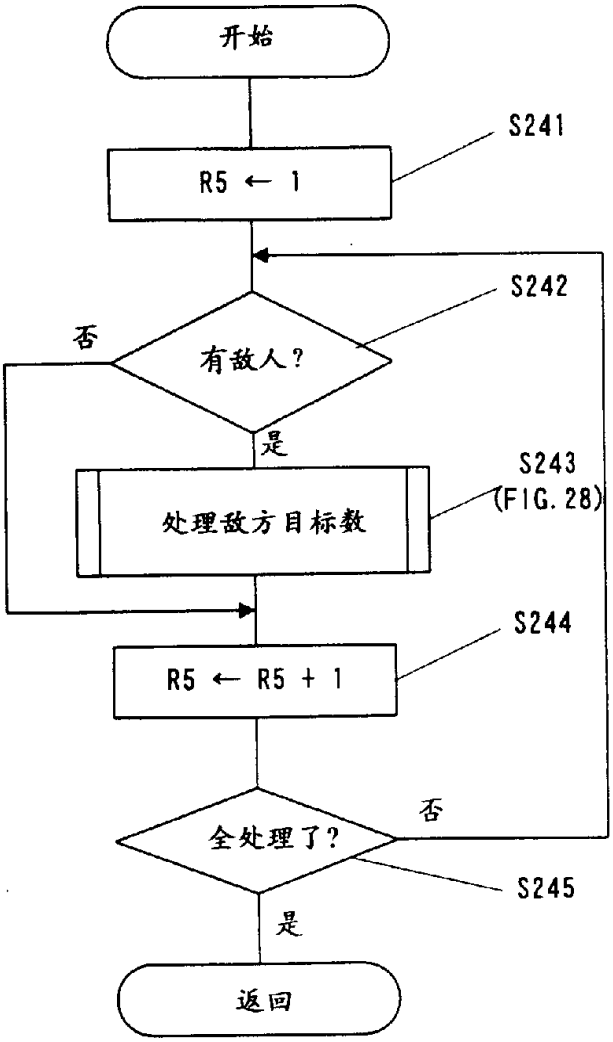

开始
S241
R5 ← 1
S242
否
有敌人?
是
S243
(FIG. 28)
处理敌方目标数
S244
R5 ← R5 + 1
否
全处理了?
S245
是
返回

图 27

敌方目标处理2

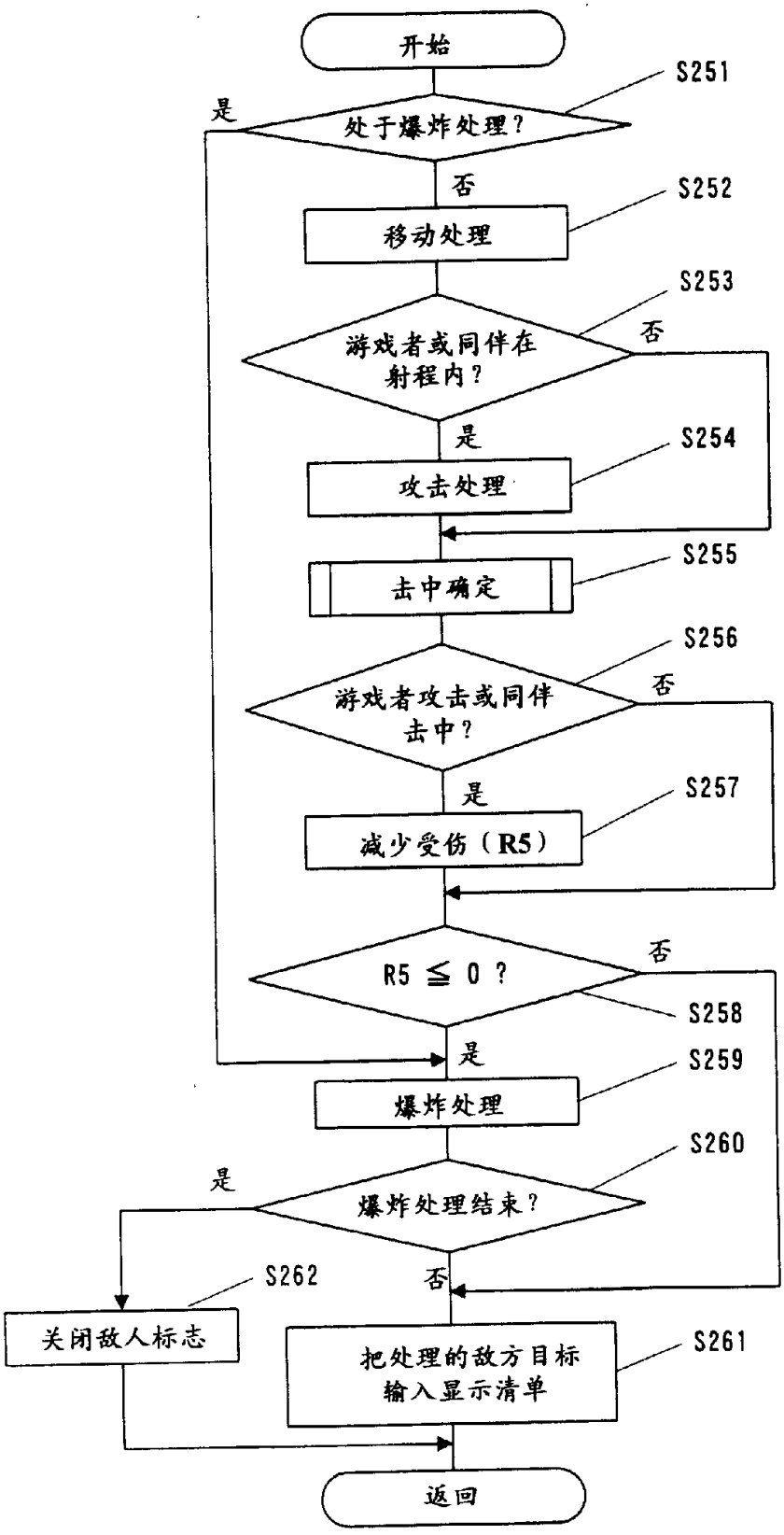

开始
S251
处于爆炸处理？
是
否
S252
移动处理
S253
游戏者或同伴在射程内？
否
是
S254
攻击处理
S255
击中确定
S256
游戏者攻击或同伴击中？
否
是
S257
减少受伤（R5）
R5 ≦ 0 ？
否
S258
是
S259
爆炸处理
S260
是
爆炸处理结束？
否
S262
关闭敌人标志
S261
把处理的敌方目标输入显示清单
返回

图 28

静止目标处理

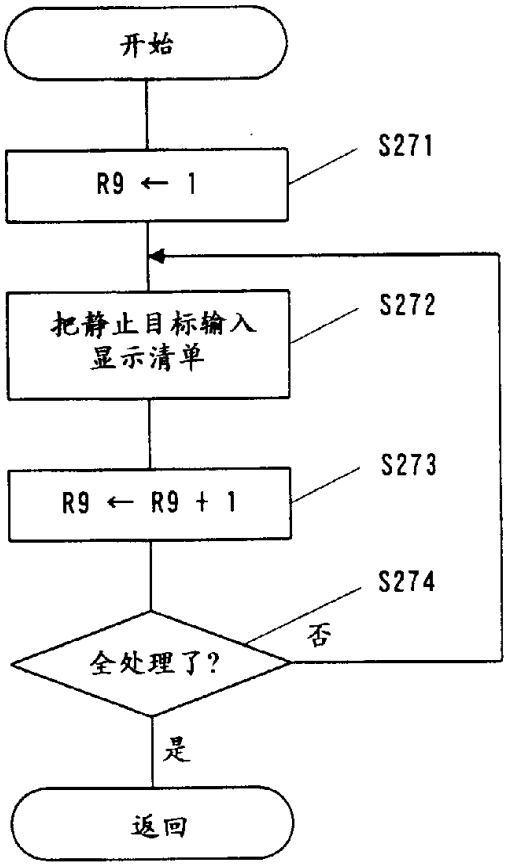



图 29

让与处理

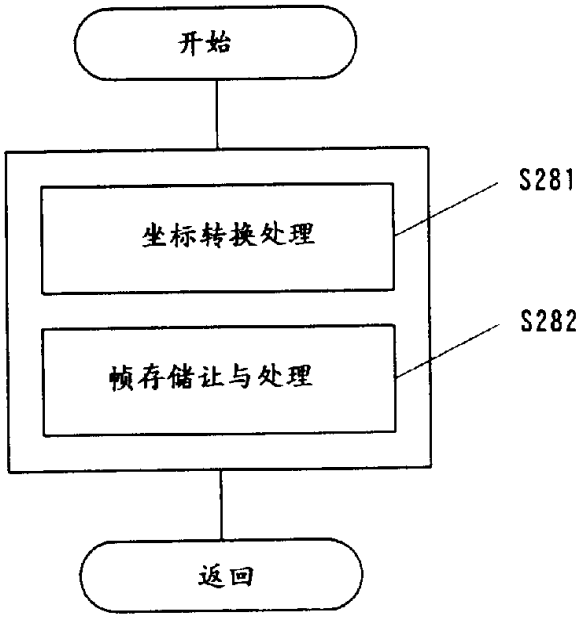



图 30

伴音输出处理

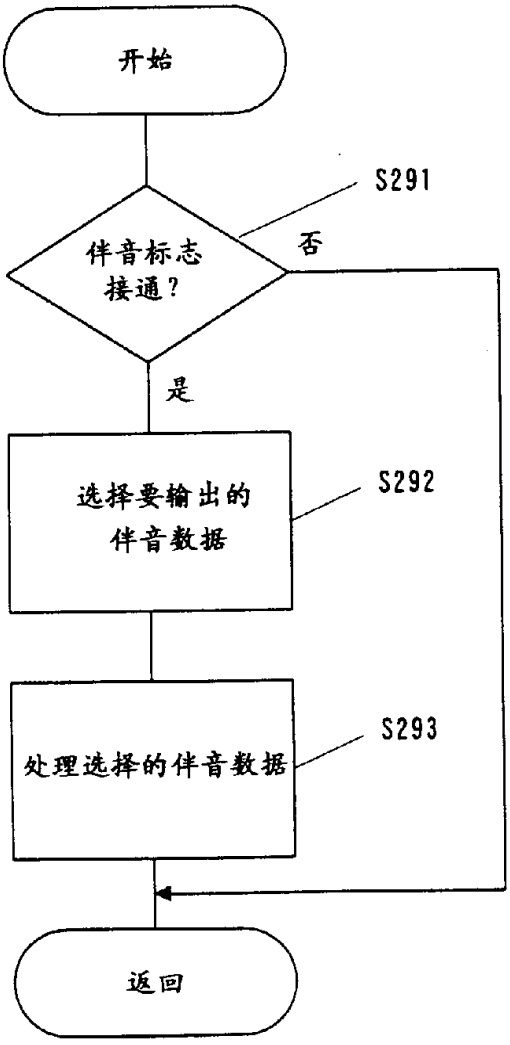



图 31

游戏者目标处理1a(位置)

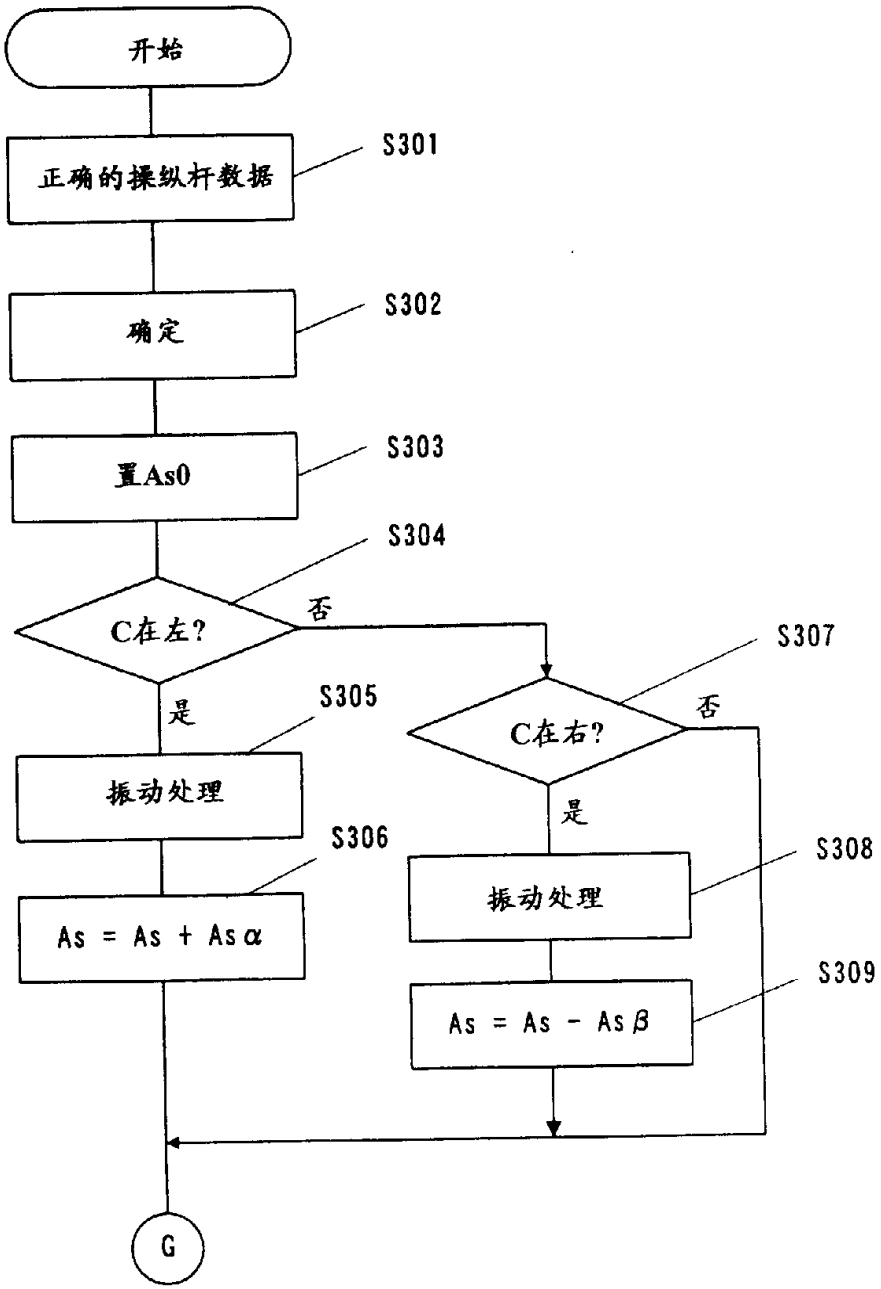



图 32

游戏者目标处理1b(位置)

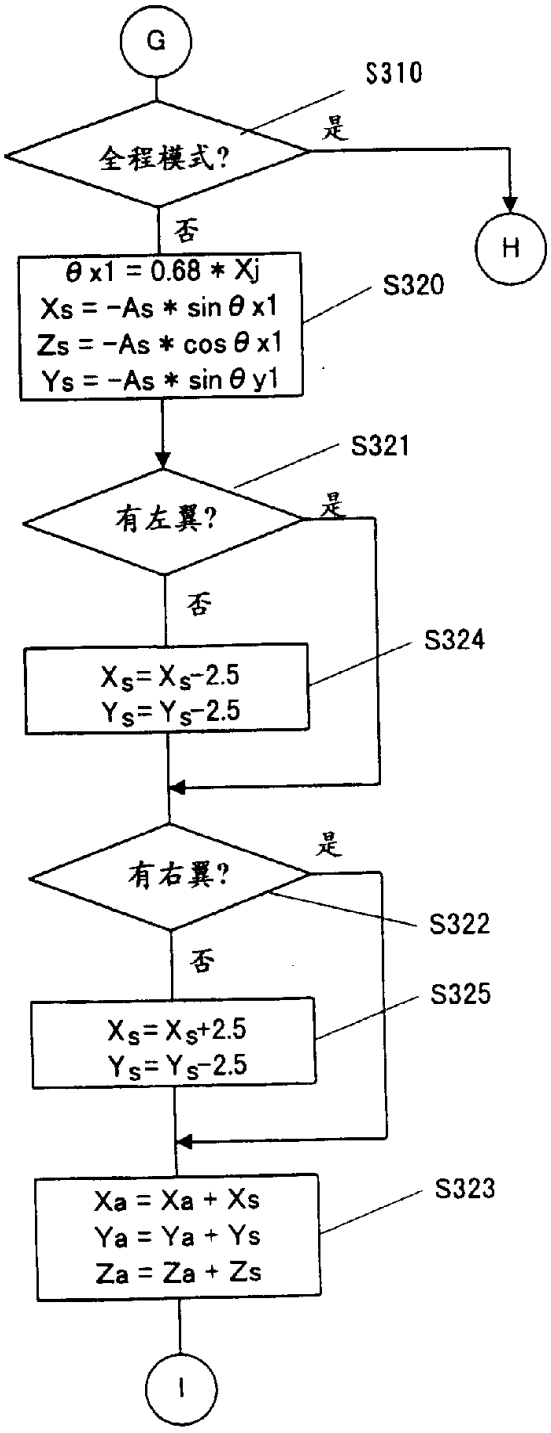
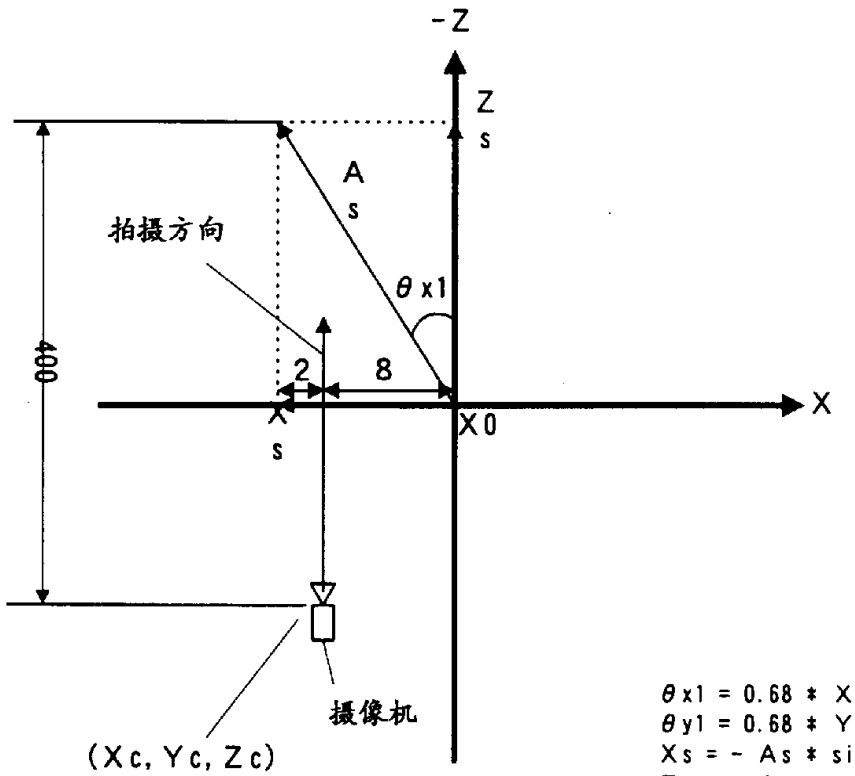



图 33

单向卷动模式




$$\theta x1 = 0.68 * Xj$$
$$\theta y1 = 0.68 * Yj$$
$$Xs = -As * \sin\theta x1$$
$$Zs = -As * \cos\theta x1$$
$$Ys = -As * \sin\theta y1$$

$$Xc = (Xa - X0) * 0.8$$
$$Yc = (Ya - Y0) * 0.8$$
$$Zc = (Za + abs(Z\alpha)) + 400$$
$$Zc = (Za - abs(Z\beta)) + 400$$
$$Zc = Za + 400$$

图 34

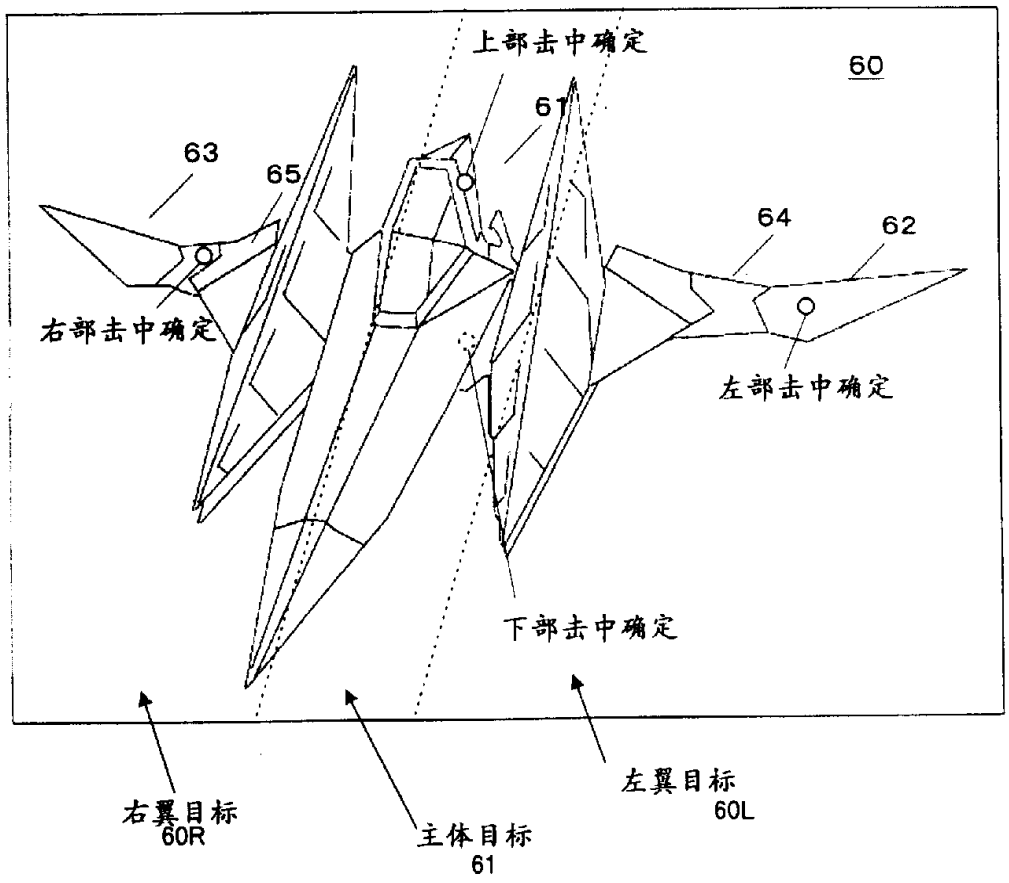

上部击中确定
60
61
63
65
64
62
右部击中确定
左部击中确定
下部击中确定
右翼目标
60R
主体目标
61
左翼目标
60L

图 35

游戏者目标处理1c(位置)

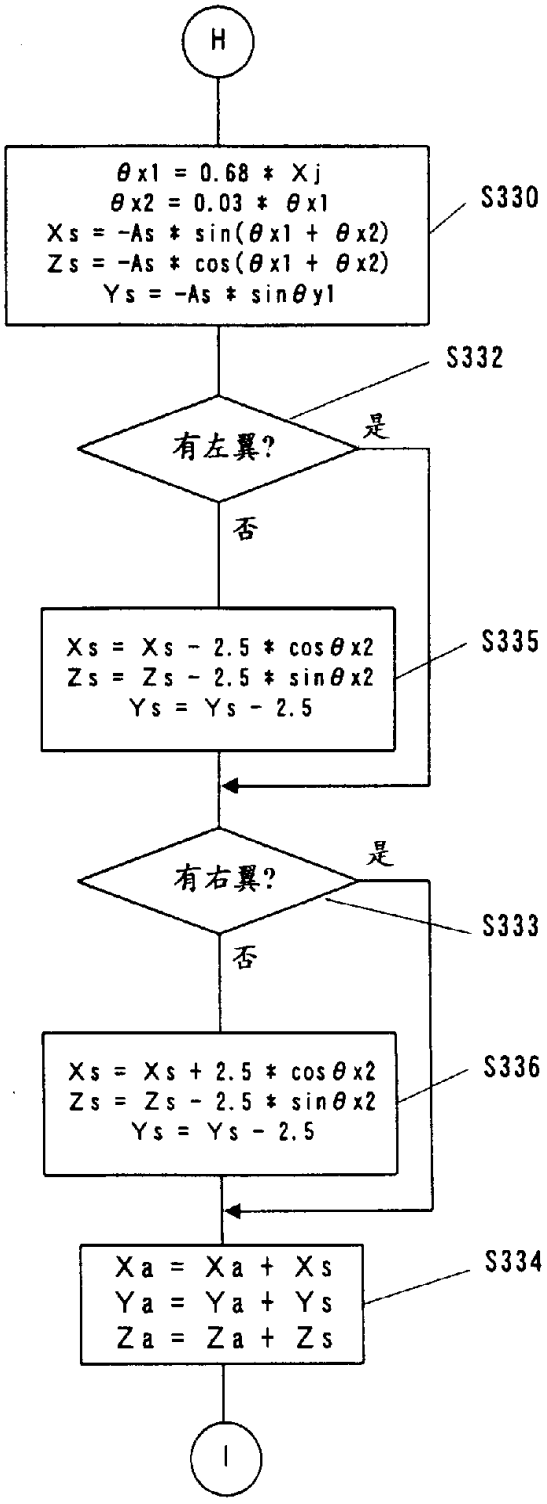
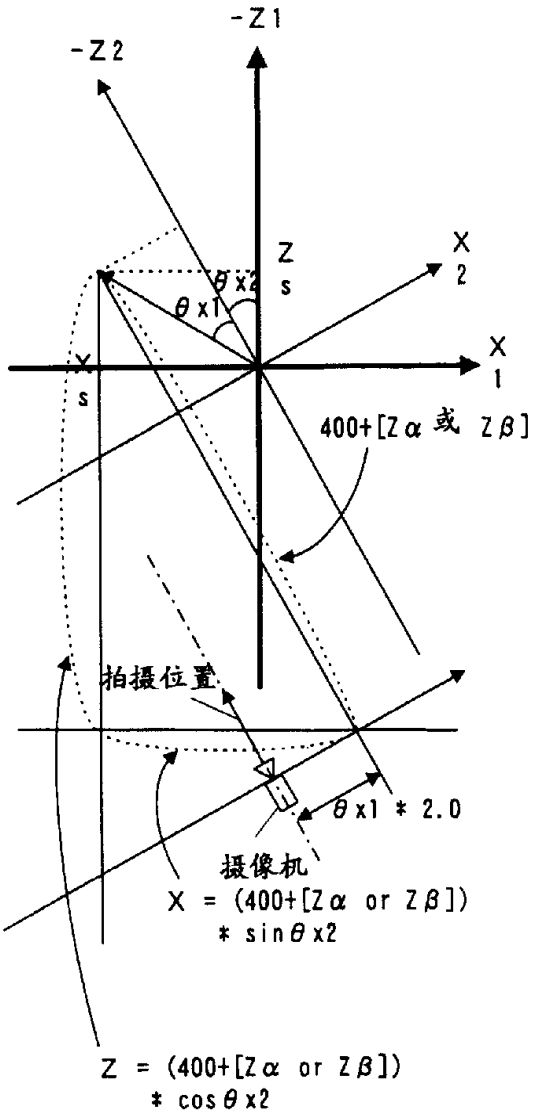



图 36

全程模式




角度

$\theta x1 = 0.68 * Xj$
$\theta y1 = 0.68 * Yj$
$\theta x2 = 0.03 * \theta x1 + \theta x2$
( 前一帧值 )

速度

$Xs = - As * \sin(\theta x1 + \theta x2)$
$Zs = - As * \cos(\theta x1 + \theta x2)$
$Ys = - As * \sin\theta y1$

位置

$Xa = Xa + Xs$
$Zs = Za + Zs$
$Ys = Ya + Ys$

Xa, Ya, Za : 三维坐标位置
Xs, Ys, Zs : 三维坐标速度

摄像机位置

$Xc = Xa + (400+[0 \text{ or } abs(Z\alpha) \text{ or } - abs(Z\beta)]) * \sin\theta x2 + Xr$
$Zc = Za + (400+[0 \text{ or } abs(Z\alpha) \text{ or } - abs(Z\beta)]) * \cos\theta x2 + Zr$
$Yc = Ya * 0.8$

$Xr = \theta x1 * 2.0 * \cos\theta x2$
$Zr = \theta x1 * 2.0 * \sin\theta x2$
$Z\alpha = As\alpha * \cos\theta x1$
$Z\beta = As\beta * \cos\theta x1$

图 37

游戏者目标处理2a(击中确定)

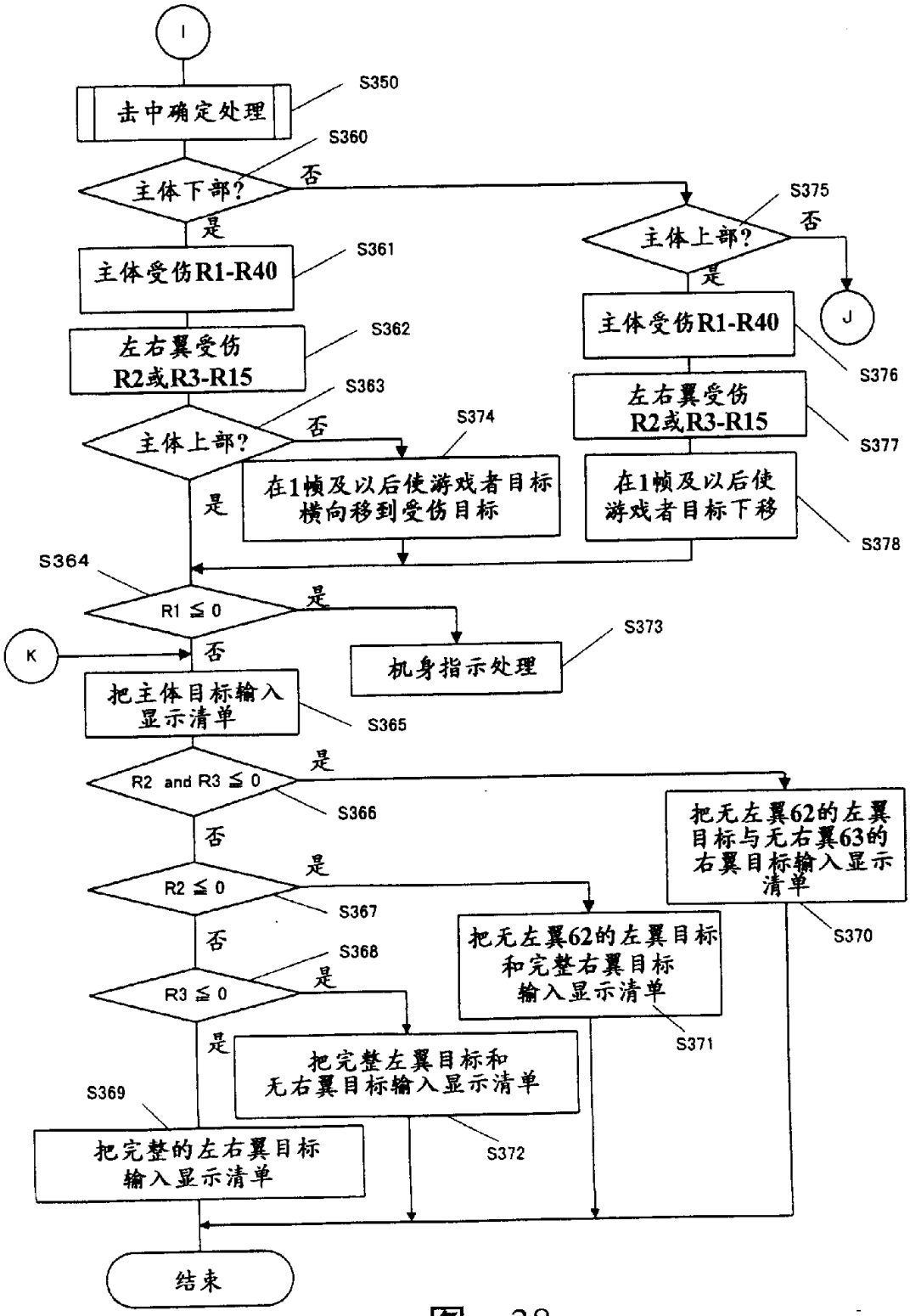



图 38

击中确定(标准)

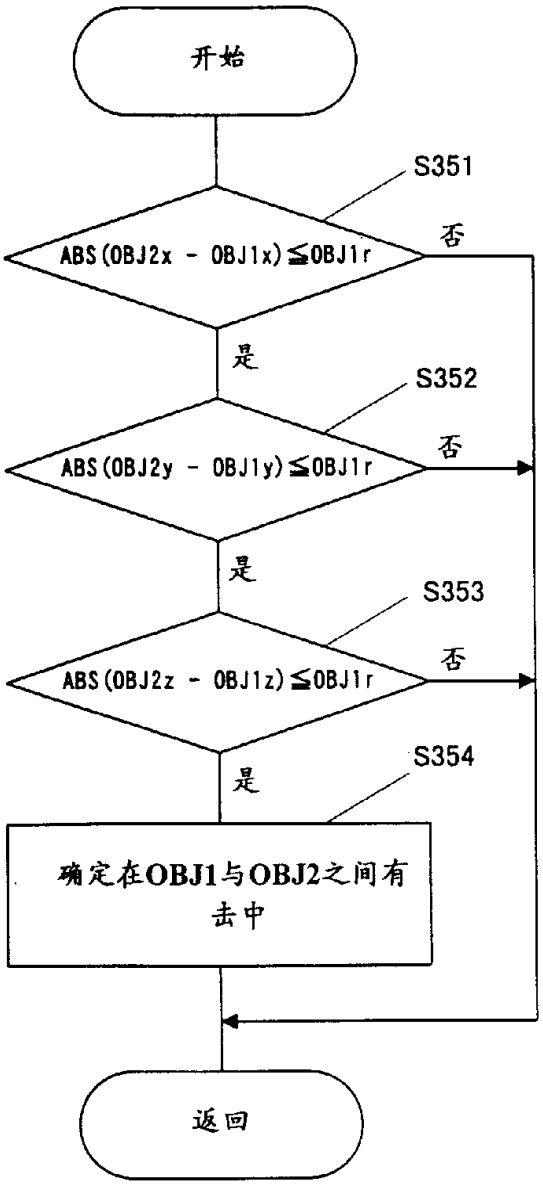

开始
S351
ABS(OBJ2x - OBJ1x)≦OBJ1r
否
是
S352
ABS(OBJ2y - OBJ1y)≦OBJ1r
否
是
S353
ABS(OBJ2z - OBJ1z)≦OBJ1r
否
是
S354
确定在OBJ1与OBJ2之间有击中
返回

图 39

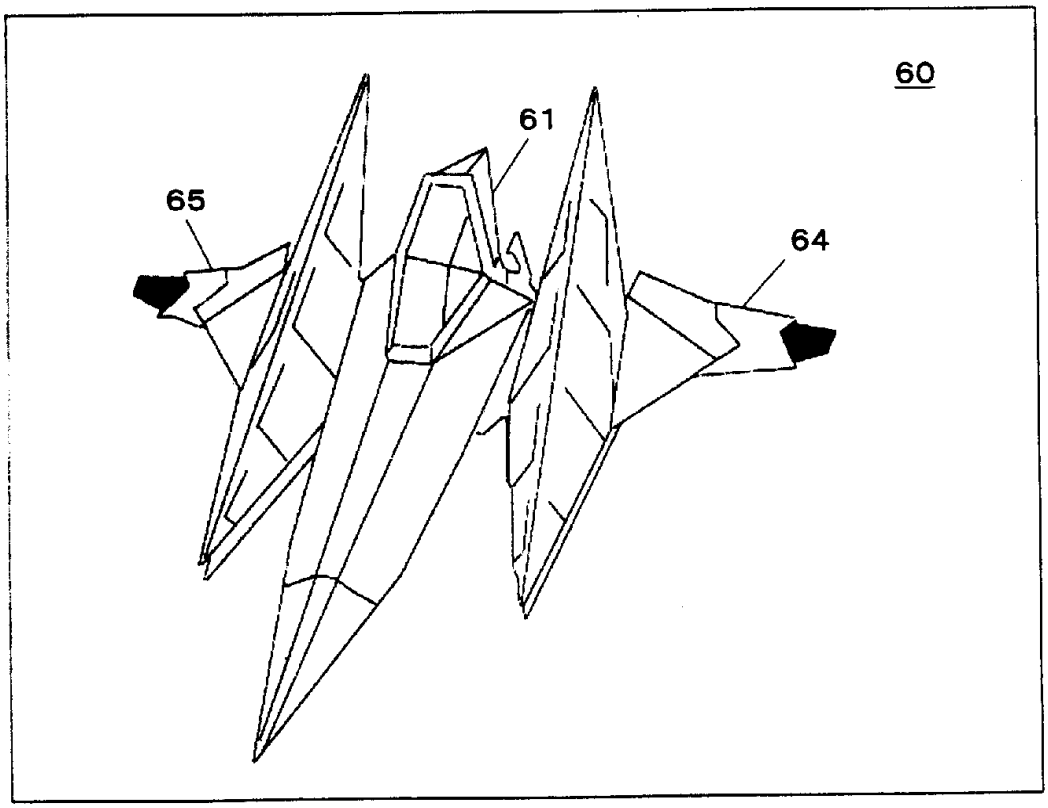



图 40

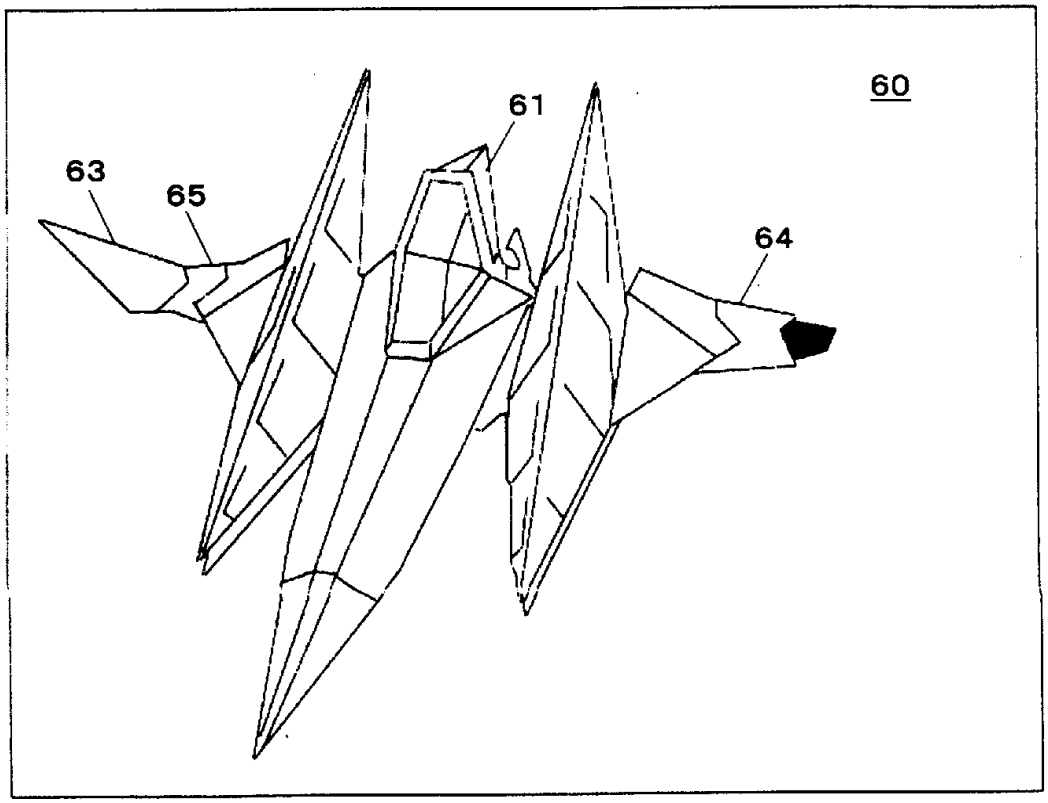



图 41

游戏者目标处理(击中确定)

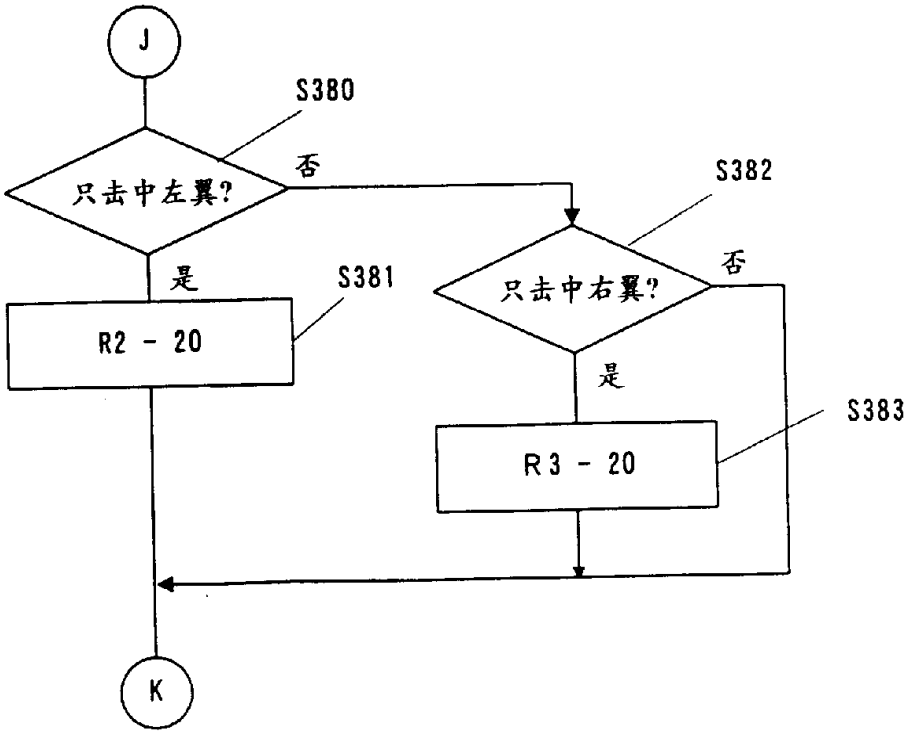



图 42

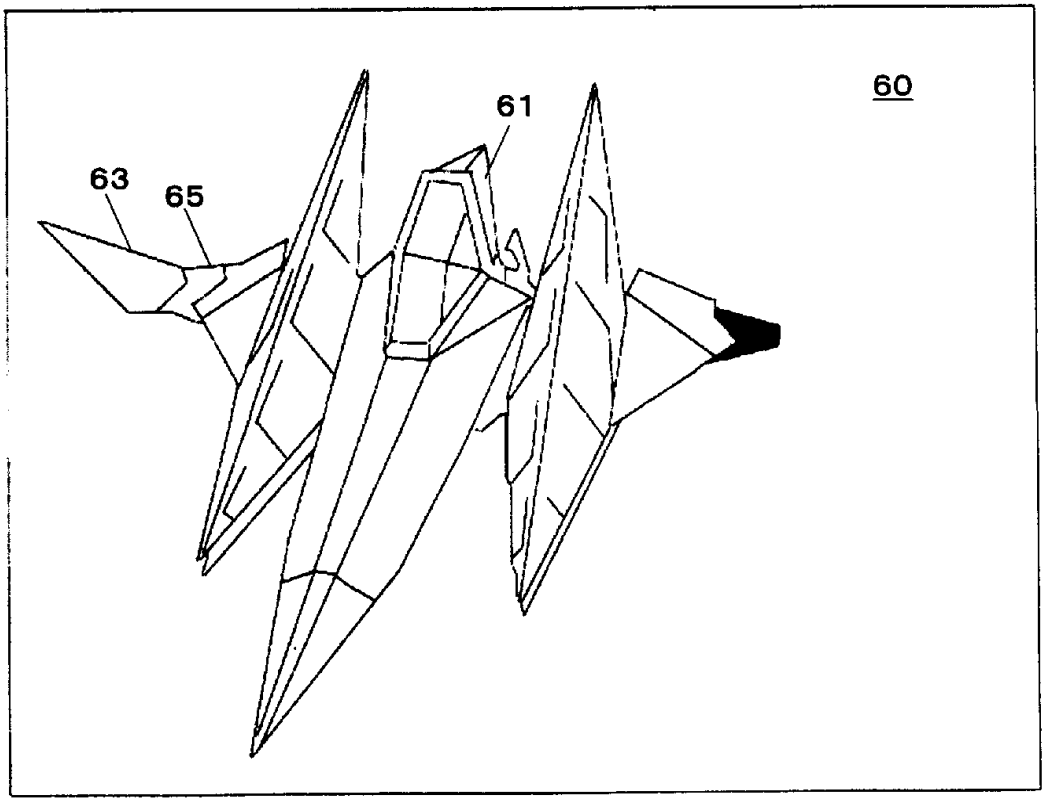



图 43

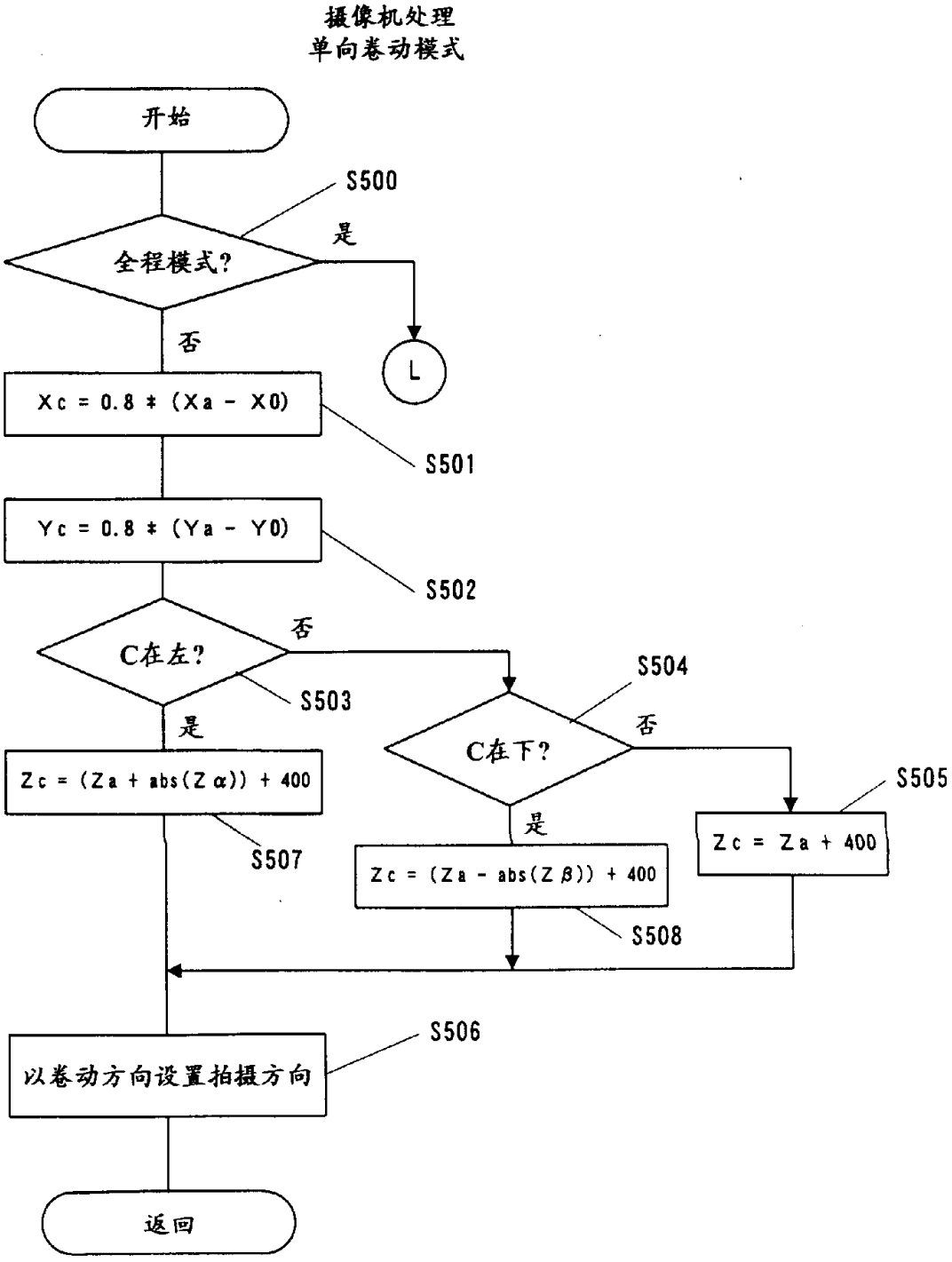

摄像机处理
单向卷动模式
开始
S500
全程模式?
是
否
L
Xc = 0.8 * (Xa - X0)
S501
Yc = 0.8 * (Ya - Y0)
S502
C在左?
否
S503
是
S504
C在下?
否
是
Zc = (Za + abs(Zα)) + 400
S507
Zc = (Za - abs(Zβ)) + 400
S508
Zc = Za + 400
S505
以卷动方向设置拍摄方向
S506
返回

图 44

摄像机处理
全程模式

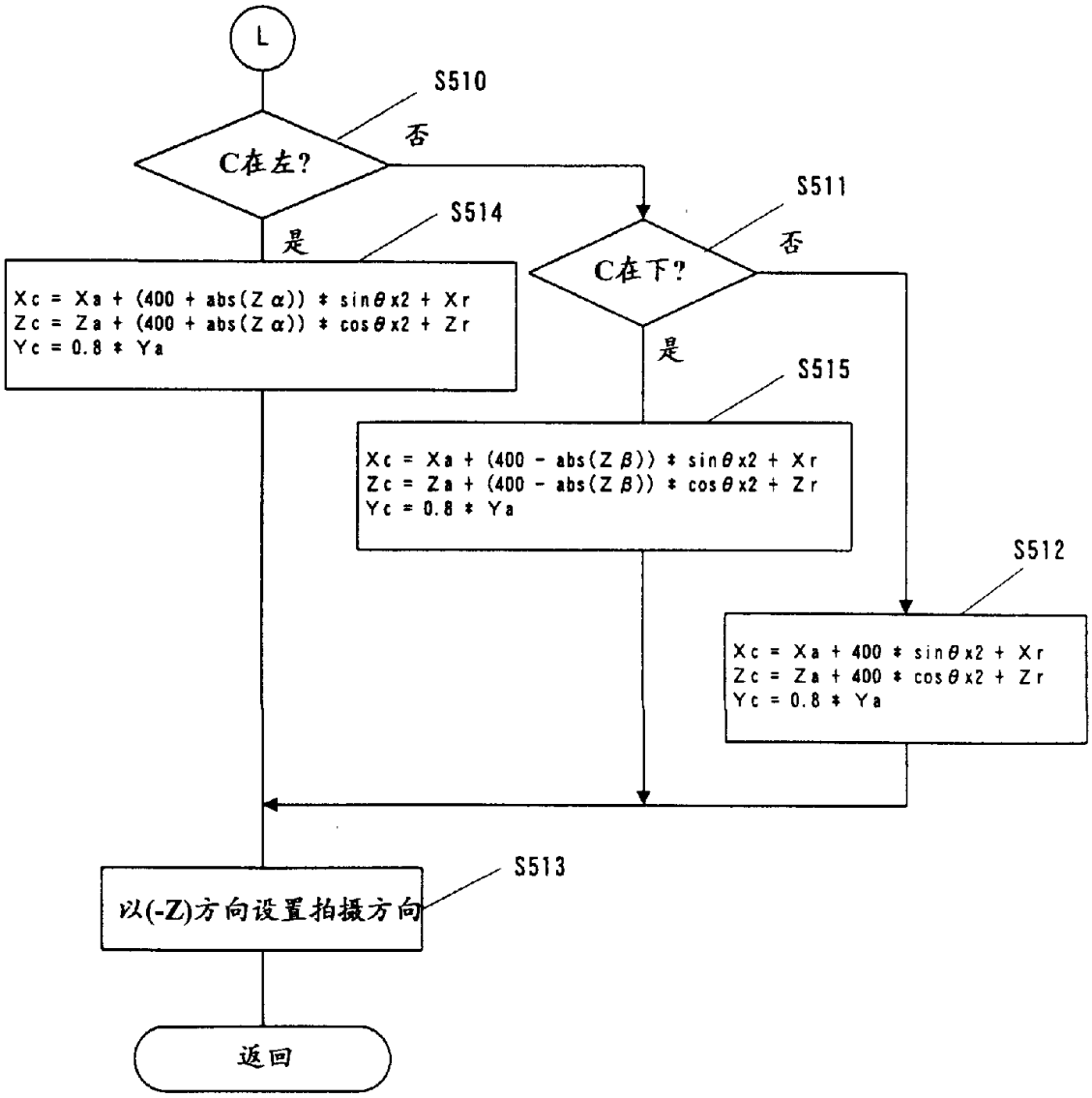



图 45

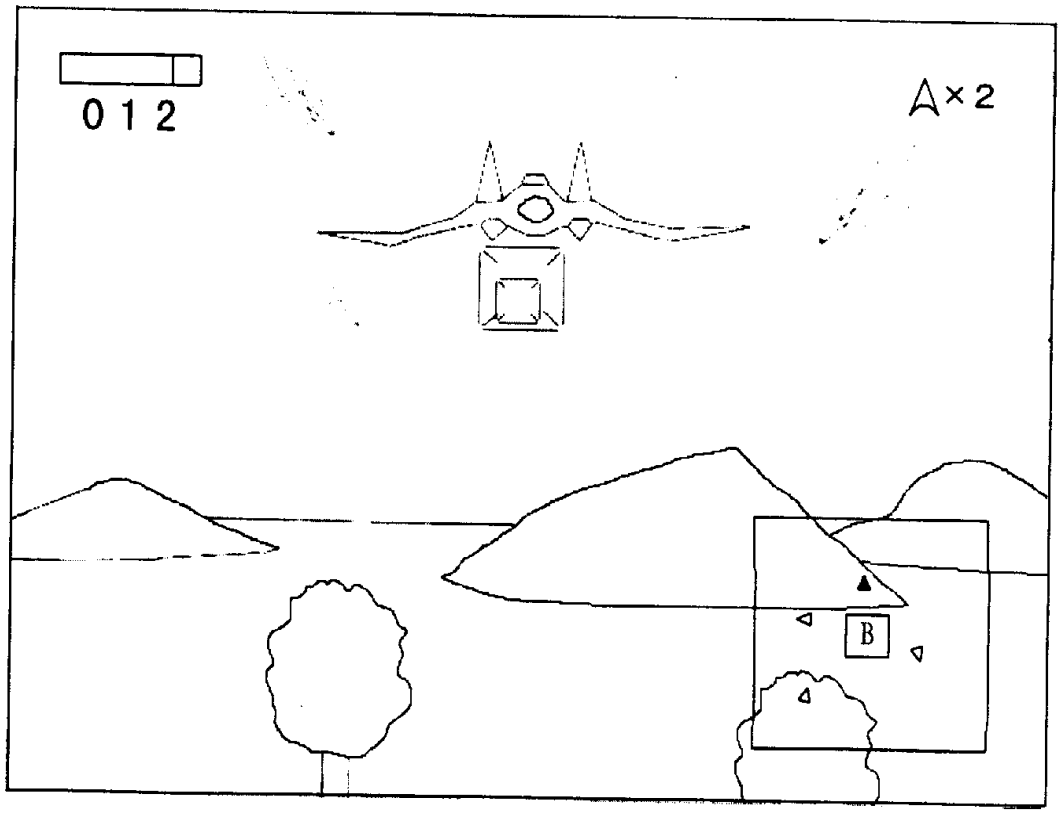



图 46

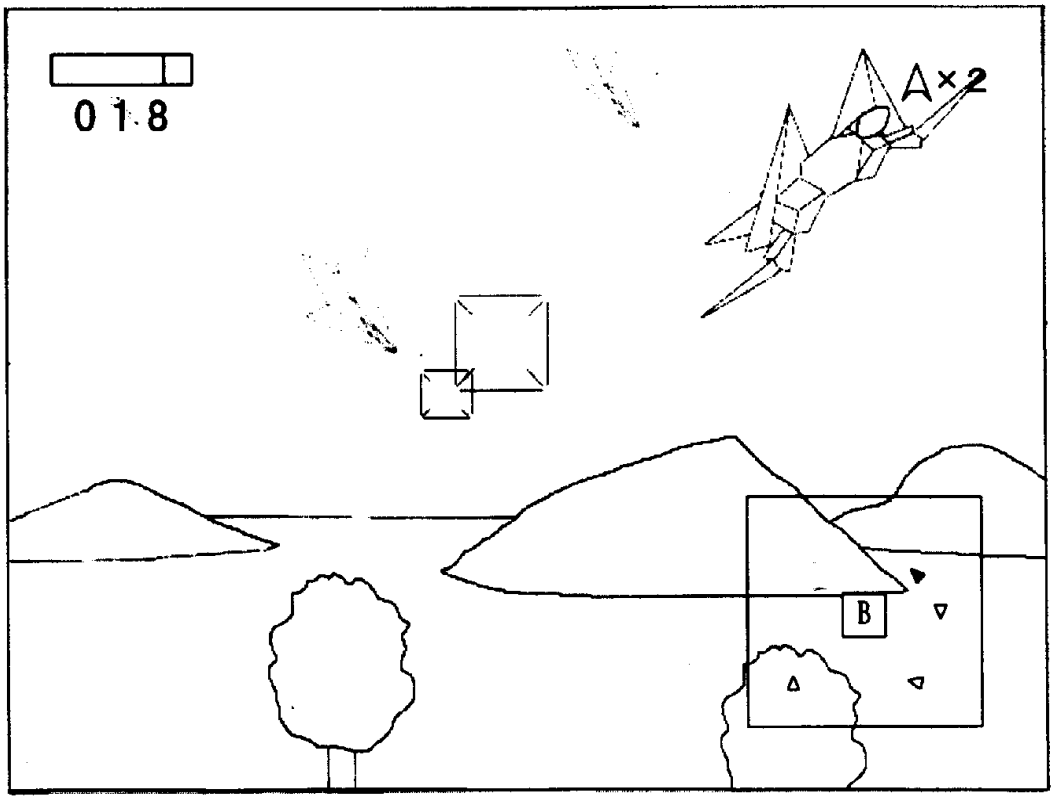



图 47

击中确定(敌人)

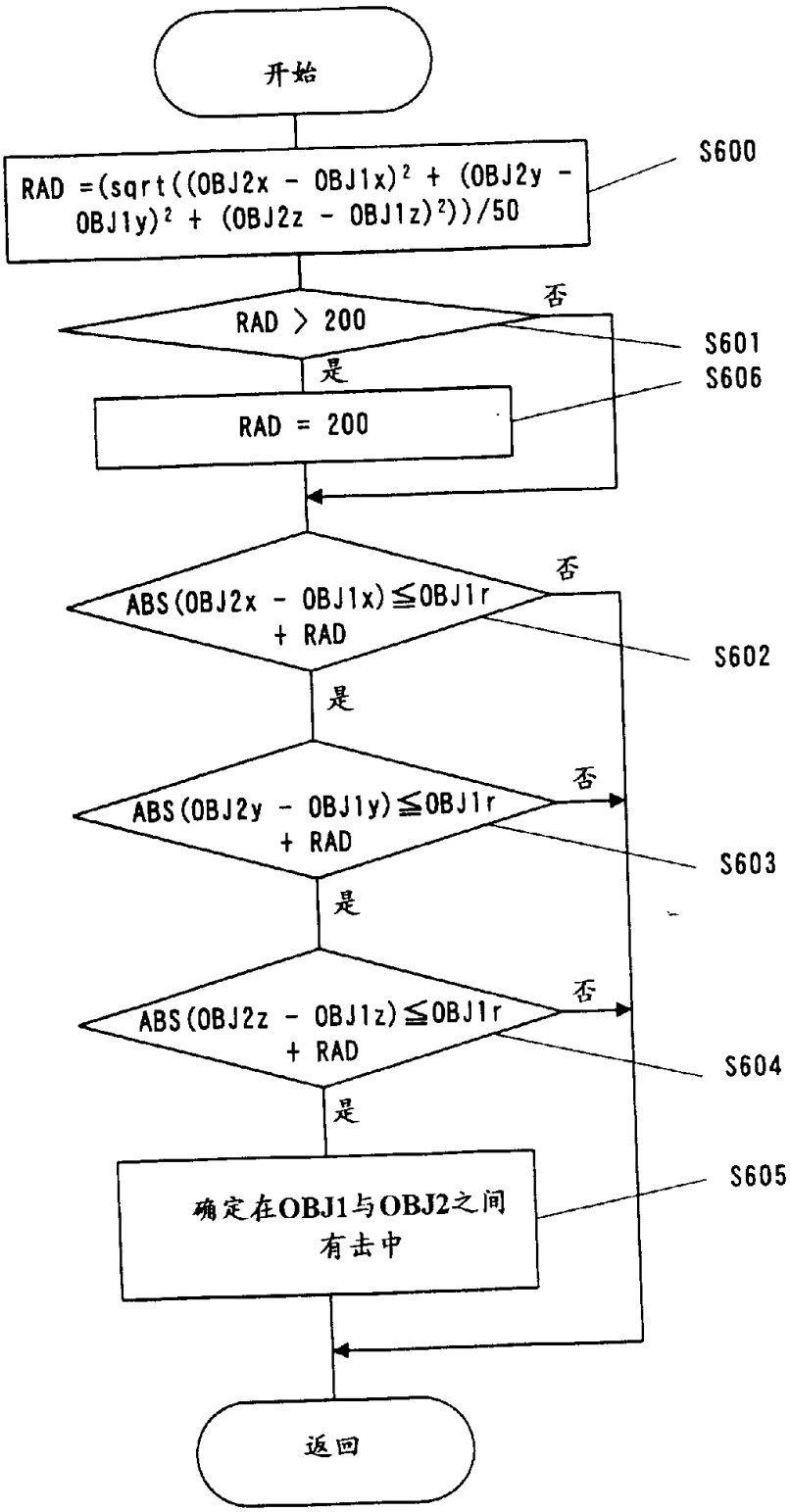

开始
RAD =(sqrt((OBJ2x - OBJ1x)² + (OBJ2y - OBJ1y)² + (OBJ2z - OBJ1z)²))/50
S600
RAD > 200
否
S601
是
RAD = 200
S606
ABS(OBJ2x - OBJ1x)≦OBJ1r + RAD
否
S602
是
ABS(OBJ2y - OBJ1y)≦OBJ1r + RAD
否
S603
是
ABS(OBJ2z - OBJ1z)≦OBJ1r + RAD
否
S604
是
确定在OBJ1与OBJ2之间有击中
S605
返回

图 48