Bug or Feature?

Cameron Browne, Queensland University of Technology (QUT)

This article explores the relevance of the 'bug or feature' concept found in computer programming to the process of game design. Several examples are presented of successful games with either apparent bugs that proved beneficial on closer analysis, or actual bugs whose solution provided worthwhile emergent benefits. Game design is posed as a bug fixing process.

1 Introduction

THERE is an old joke among computer programmers that when a customer complains about a bug[1] in a piece of software, it is sometimes easiest to just describe the resulting behaviour as a 'feature' of the program. For example, an error that inadvertently deletes a non-critical database every millionth entry might be described as a 'memory saving feature'.

This analogy can be extended to the practice of game design, which often starts with an initial idea for a desired mechanism or behaviour and some preferred set of equipment, followed by an iterative process of identifying bugs in the design and fixing them, hopefully improving the game with each iteration. In this context, a *bug* refers to some undesirable behaviour and a *feature* refers to some desirable behaviour, resulting from the interaction between the rules and the equipment.

This article explores two aspects of the bug/feature dichotomy relative to game design; bugs that are actually features and bugs that can be turned into features. Sometimes it is not clear whether a particular design aspect is a bug or a feature, until the situation is studied in greater depth.

We are not interested in mere problem solving here; the design of *any* game could be described as one long process of bug fixing. Instead, we are interested in bugs that produce some emergent and unexpected benefits, either through side effects or through their solution, that add some significant feature to the game. The best examples play off the detrimental behaviour of the bug to produce some beneficial result.

2 Bug or Feature

This first set of examples includes games that contain apparent design bugs that have turned out to be positive features (or can be viewed as such), without modification.

2.1 Mambo

Mambo[2] is a tile placement game for two players, Red and Blue, who take turns placing one of the Mambo tiles shown in Figure 1 (left) in any orientation to match at least one adjoining tile. The aim is to kill an enemy group by stopping it from further growth. For example, Red has killed the central Blue group to win in Figure 1 (right).

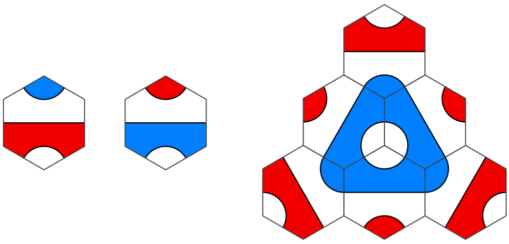

Figure 1. The Mambo tiles and a Red win.

The rules for Mambo were initially simpler and required only that players close an enemy group. However, this initial rule set had a problem in that players could create unplayable *null points* that no tile placement could match, such as the point marked × in Figure 2, and thus protect their groups from closure to avoid defeat *ad infinitum*.

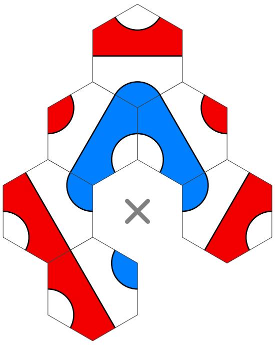

Figure 2. No tile can be played at this null point.

[1] The term *bug* refers here to an error in a program or mechanical system that produces unexpected behaviour.
[2] http://www.cameronius.com/games/mambo/