Automated Playtesting with RECYCLEd CARDSTOCK

Connor Bell, Hendrix College

Mark Goadrich, Hendrix College

Game designers benefit greatly from playtests of their prototypes in development. However, finding experienced, independent playtesters for repeated plays is often difficult. This paper describes RECYCLE, *a card game description language, and* CARDSTOCK, *an implementation of* RECYCLE *which automatically playtests card games with both random and intelligent players. Our system can assist game designers by providing insight into the average player decision branching factor, turn order advantage, game length, and the potential for strategy. We demonstrate its use by playtesting variants of the games Agram, Pairs and War.*

1 Introduction

WHEN game designers seek to create fun and engaging games, they strive to avoid common design pitfalls. First, players can struggle from 'analysis paralysis' in a game with too many viable choices [1, 2]. Alternatively, a game with too few options leaves little room for player agency and control over their fate [3, 4]. A game must find a balance between the opportunity for strategy and the size of the branching factor of options available to the players.

Second, it is easy to inadvertently create a game that is unbalanced, giving certain players an inherent advantage in the game simply based on their turn order. A game should strive to provide a fair experience for all players [5]. Finally, an end condition of a game could be created that is either unachievable or only winnable through extended play. Modern board and card games tend to limit playing time to a reasonable length of less than an hour [1].

However, the rules of a game combined with the choices and motivations of the players can be categorised as a complex adaptive system [6]. This means that emergent properties such as fairness and game length cannot be readily understood from examining the rules in isolation, but are only discoverable in the moment of play.

In pursuit of these virtues of meaningful choice, fairness, and appropriate length, game prototypes typically undergo multiple iterations and modifications while being developed. Depending on feedback from early playtests, a game designer could seek to improve the play experience by changing some elements of the game, for example, by tweaking the point values, altering the mechanics, restructuring the winning conditions, or modifying the components [7]. Each of these changes constitutes a new game variant that must be subsequently playtested. This creates a cycle that can quickly exhaust the time and patience of the available playtesting volunteers.

We introduce RECYCLE, a card game description language, and CARDSTOCK, an implementation of RECYCLE with which designers can automatically playtest card games. After encoding the rules of a game in RECYCLE, designers can perform multiple Monte Carlo simulations of a game to explore the emergent properties and learn the shape of its play space. By separating the game description from the game engine, designers can focus exclusively on modifying the rules and mechanics, and then quickly test for unexpected side-effects using both random and semi-intelligent automated players.

We focus our work on card games because they provide a small yet interesting space of games for automated playtesting tools. Many card games incorporate randomness, via the ability to shuffle and deal from a fixed set of cards without replacement, and hidden information, either through playing cards face down or through player ownership of sets of cards [8].

We first examine related work in the development of card game description languages, followed by a detailed explanation of the elements of RECYCLE and a sample encoded game. We then briefly discuss the CARDSTOCK implementation and the options for automated players. We demonstrate the utility of RECYCLEd CARDSTOCK by playtesting variants of three card games to discover their strengths, weaknesses and potential for strategy. Finally, we discuss RECYCLE's limitations and conclude with thoughts on future work.

2 Related Work

Thielscher [9] states that a general game description language 'includes: knowledge of the players and the initial position; the legal moves, and how

Bell, C. and Goadrich, M., 'Automated Playtesting with RECYCLEd CARDSTOCK', *Game & Puzzle Design*, vol. 2, no. 1, 2016, pp. 71–83.