

Growing Maps in Strategic Multiplayer Browser Games

Expandir Mapas em *Multiplayer Browser Games* de Estratégia

Gonçalo Pereira, Pedro A. Santos & Rui Prada

Instituto Superior Técnico, INESC-ID

gdgp13@gmail.com, pasantos@math.ist.utl.pt, rui.prada@gaips.inesc-id.pt

Resumo

Neste artigo descrevemos um método para criar mapas estratégica e visualmente ricos para browser games. O nosso método cria mapas dinâmicos que se expandem de acordo com o padrão de inscrições dos jogadores e que se adaptam às escolhas dos mesmos. Este método elimina limitações do game designer e promove o desenvolvimento de jogos escaláveis focados na estratégia.

Palavras-chave: browser game de estratégia, mapa dinâmico, mapas que crescem

Abstract

In this paper we describe a method to create strategically and visually rich game map environments for browser games. Our method creates dynamic maps which expand according to the players' subscriptions pattern and which adapt to the players' choices. This method eliminates current game designer limitations promoting the development of scalable strategy focused games.

Keywords: strategy browser game, dynamic maps, growing maps

1. INTRODUCTION

Internet-based gaming has grown tremendously in the last few years and it is possible to find many games from different genres (Gamasutra, 2009). Among these are the browser games which are defined by only requiring internet users to have a browser installed in their computer in order to play. Some are multiplayer and take advantage of the internet's widespread nature to link many players into a group game experience. Communities in these games are large and can vary from several hundreds to thousands of players gathered for a collective game experience (Massively Multiplayer Online Game, 2009).

With a few exceptions multiplayer browser games fall into two main categories, namely Role-Playing Games (RPG) and strategy games (List of multiplayer browser games, 2009). In strategy games, the player usually can choose some special traits ("race", "tribe" or "faction") that give him specific advantages and disadvantages. These influence economic and military development and are sometimes influenced by the environment. In this game genre, player location, terrain type and richness in resources influence the strategic balance and player challenge. These and the map's visual detail then affect the player's experience. Currently

most strategy multiplayer online games are using a model of map generation which cannot provide both map complexity (visual and strategic) and scalability for the multiplayer context.

Our goal is to create a game experience where players are placed in a detailed and evolving map environment. This is accomplished by expanding maps dynamically as players enter. The idea is that maps should not limit either game complexity or scalability, they should simply enable the best strategic and diverse environment to be created and experienced.

2. EXISTENT MAP CREATION

There are two main ways of creating maps. Manually or through procedural generation. The manual creation method is characterized by requiring the creator's input for every detail of the map, whether it's a terrain type or a resource distribution. This allows the game designer to be creatively unbounded. He can directly create specific strategic challenges, like narrow passages or mountain ranges. However, the large amount of input required makes it very time consuming. The bigger and more complex the map is the more time it is required for its creation. So due to the time consuming nature of the method, multiplayer maps created in this way become limited in terms of size and number of players they can accommodate.

The procedural generation approach has the advantage of consuming much less time to create maps than the handmade method. The approach is composed of several stages which vary depending on the game (Prachyabrued et al, 2007). Some common stages are heightfield generation (Olsen, 2004), resources distribution and terrain feature creation. These maps can then be adapted to accommodate as many players as the game needs, but due to the unpredictability of player's choices (such as a faction) and player entry times and amount, they must be generic enough to be able to satisfy every player's option. This generalization usually results in a strategically poorer environment. This kind of maps can be seen in games such as Travian¹ where generic game maps are created at a given server's start and joining players are placed in the game expanding from the center of the map towards its limits.

None of the approaches described above can simultaneously accommodate a rich strategic environment and a large and dynamic community (of at least hundreds) of players. The first is impractical for large communities of players and the second produces strategically poor maps.

3. GROWING MAPS

Our approach intends to give players an improved multiplayer online experience by creating a rich strategic environment where the map's characteristics are both influenced by player's faction and time of subscription. This is done by expanding maps with each player subscription and generating territory specifically to accommodate the players' faction.

This approach was developed for the game Almansur Battlegrounds². *It is a strategy game of politics, economy and war set in the early middle ages or, in some scenarios, fantasy world.* The game is turn-based and the map has a deep influence in the game, ranging from different troop movement speeds for each race to the difficulty and consequent cost and productivity that the same building can have in different terrains. In this environment there are several

¹ <http://www.travian.com>

² <http://www.almansur.net>

racas for fantasy games: Barbarian, Dwarf, Elf, Human and Orc. Each one has different needs in terms of terrain characteristics, for example Orcs need swamps for better development but Elves need forests. The player starts as a lord from a race of his choice with a small army and territory which he can then develop economically, make alliances or expand by conquering neighbouring territories. Until now this game has relied on manually created maps for a fixed number of players. Our goal is to provide growing game maps while maintaining the game balance and keeping the existent strategically rich environment.

3.1. HOW TO CREATE THE MAPS

At the core of any map there is a model of how it should be represented. Almansur Battlegrounds uses a vertically aligned hexagonal grid where the terrain unit is the hexagon.

To create the maps we use an iterative procedural generation method which has several input parameters. The most important parameters are the terrain prototypes for each race that can be played in the game and the number of turns a player placeholder takes to expire. A *terrain prototype* contains base values for each of the terrain's characteristics which the game uses. Examples of terrain characteristics are altitude, fertility and the abundance of arborization.

These prototype terrains are carefully crafted by the game designer, since they are the basis for an adequate and balanced map generation. Besides the base values of each prototype the game designer also groups prototypes by categories and evaluates the quality of each within their category based on heuristics. The categories will enable the use of the adequate prototypes for the different terrain purposes. Examples of categories are "race dwarf", "feature lake" and "neutral terrains". The evaluation of individual prototypes in a group also enables their differentiated use, based on their quality regarding that group.

A *player placeholder*(PP) is a specific position in the map that can be a starting point for any entering player. As we will see below, when there are no players joining for a period of time, the generator must avoid upcoming players be placed adjacent to others much more developed. The parameter with the number of turns a player placeholder takes to expire specifies how many turns can elapse before an unused PP position becomes unavailable.

In order to be able to support an ever growing map, it is first necessary to find how it will grow. The idea is that the map should expand in an interleaved way between neutral zones and player zones which balance conflict and expansion. A zone is a set of terrain units that are generated with a specific purpose based on one or several categories of terrain prototypes. Based on game tests, with manually created maps previous to this method, a player/neutral zone schema was chosen and is illustrated in figure 1.

The schema shown represents player 1 which is surrounded by three player zones and three neutral zones. These player zones represent other players which cause direct points of conflict with the player

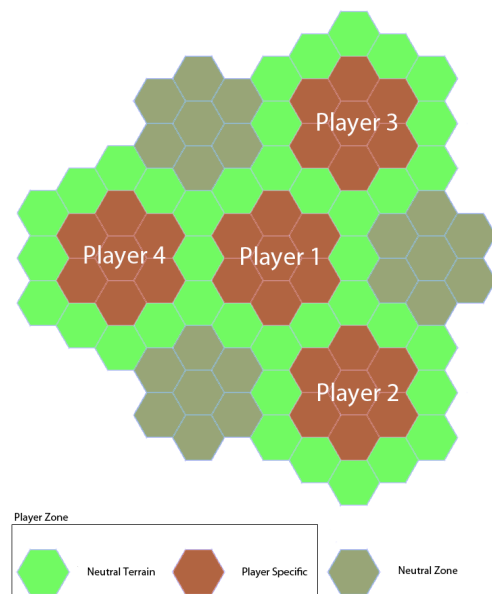


Figure 1 - Player/Neutral zones schema

in the center. Also notice that these zones are composed of two kinds of terrain, the player specific and the neutral territories.

Based on this player/neutral zone distribution we can then scale our generation model to grow as needed. The creation of new zones then depends only on player entries, where we impose the constraint that one player must always be placed near an already existent one. Once a new player is created the remaining possible neutral and player zones surrounding it are made available for the generation process and then enable more players to join. The growth of the map is then done in a spiral order for both player and neutral zones. This ensures that a new joining player will be placed in the oldest unused player placeholder available so the elapsed game time between new players and the ones already playing is globally the smallest possible. It's important to note that even though the neutral zone placement follows the spiral scheme, the actual placement only happens when a player zone that should be placed immediately after is created. This spiral schema is illustrated in figure 2.

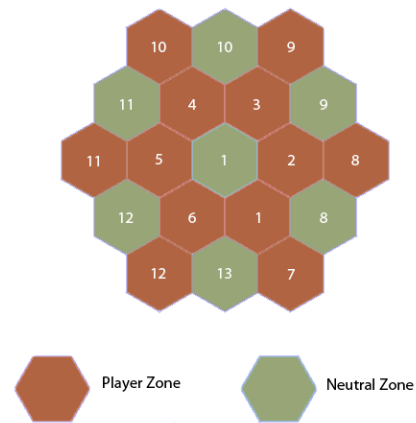


Figure 1 – Spiral Expansion

The figure illustrates a normal evolution, where there is always players entering. Notice that zones are now represented as just one hexagon for image simplicity, but the result is the same. In this example each different number represents one expansion iteration. When the first player joins one neutral and player zone are created, when the second joins a player zone alone is created, and so on.

However, player entries are not predictable, so we can have different evolutions. If there are no player entries for a given amount of game time (in our case turns) the generator marks as *expired zones* the neutral and player zones for which the initially set parameter number of turns to expire has been reached. If players could enter in these places there would be a very high probability that the older players were much more developed and could easily overrun the new ones. The expired zones are generated as neutral impassable zones. These are however different from the ones created for the normal evolution of the interleaved pattern of player and neutral zones since they are zones where expansion stops. In order to accomplish this, they are created as natural barriers, like a lake or a very high mountain impossible for game units to cross. Notice that this does not imply the game map growth will end as soon as a zone expires, it just means that the map will not grow in a pure spiral fashion but will only continue to expand in a few places. This system enables the map to expand or contract the places for expansion depending on the flow of players. An example of such a situation is illustrated in figure 3.

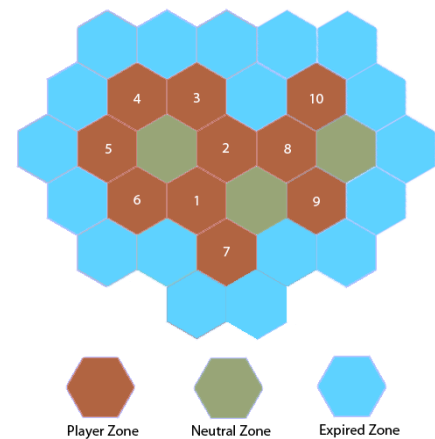


Figure 1 - Spiral expansion with expirations

In this figure we have a situation where the first eight players entered without having any expiration. Then all the neighbour places from the first seven players expired and two more players joined. Finally no more players entered and the map closed itself.

4. PRACTICAL RESULTS

In this section we show some examples of different maps created by our method and explain the game tests currently in progress. The generation of these examples followed the same zone sizes described in section 3. The fact that players start with a single hexagon is a way to reduce the impact of player view updates.

The example illustrated by figure 4 is a map with twelve players that is partly closed because of expired zones given that there are few players entering but still has some expansion zones. In figure 5 we can see the evolution of the map views from the second player to join the game in the previous example. In (a) we have a snapshot of it right after the player joined and in (b) a snapshot of it after all his neighbour player placeholders have been filled.



Figure 4 - Example of a semi-closed map

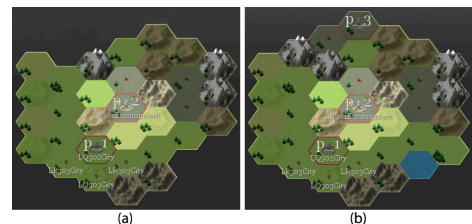


Figure 5 – Player view changes

To evaluate our solution compared to other methods and perceive its effect on the player experience, the maps are being evaluated with several zone configurations and beta tester players (most of them experienced) of Almansur. The parameters being measured are the game aggressivity, alliance formation and player survival. The player experience is analyzed based on a questionnaire to each player at the end of each game. It addresses the subjects of player challenge, map shape, resources distribution and balance.

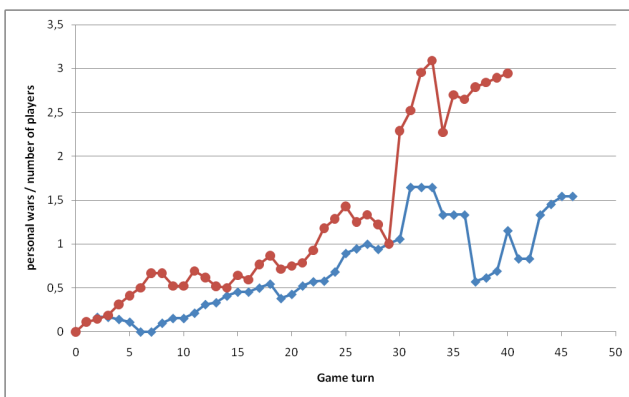


Figure 6 – Personal wars ratio for game-1 and game-2

Although the method is still under testing we already have some results from two games, one configured with player zones containing neutral terrains (game-1) and another without (game-2). To measure a game’s aggressivity we observed the ratio between the number of personal wars (between two players) and the number of players subscribed in each turn. In figure 6 we can observe this ratio during both games, and verify that as expected game-2

(in red) is more aggressive than game-1 (in blue), due to an increased player proximity. For the player survival parameter we applied Fischer’s Exact Test to verify if there is independence between the survival/death of players and the turn they subscribe. We confirmed this independence with a p-value of 0,63 for game-1 and 0,92 for game-2.

Regarding the scalability we tested the amount of players that we could add with an “acceptable” generation/waiting time for the subscribing player. After several tests we found

that with the current implementation we can add between 1000 and 1100 players with waiting times under 4 seconds and running the game on a laptop computer (Intel T7700 - 2.4GHz).

5. CONCLUSIONS

Our approach enables us to create the desired strategically rich environments on maps for strategy multiplayer online browser games and contribute to the solution of some previously existent problems. The main problem faced is the delayed player entries. Our method provides a way to combine both the desired maps and mitigate the consequences of this problem. Nonetheless the problem is not completely solved and is a subject of future work.

Our approach to the integration of multiplayer environments with strategically rich games brings several benefits for both the players and the game creators. The game creator is freed from the limitations of having to create the game around the assumption of a generic or size limited map. Beyond that this procedural method enables the emergence of strategically complex situations, such as choke points or even feature neutral zones like lakes or mountain ranges. However, this means more responsibility on the method parameterization for the game creator and intensive game testing is needed to achieve a balanced game map generation.

From the player's perspective he can now play such games repetitively without tiring due to map repetition. The amount of shapes generated and the strategic challenges created are dependent on the player entry pattern and are thus inherently random and dynamic. A map where more players enter at a given game time will be different from a map where fewer players are entering at the same game time. With our method players can now experience a visually and strategically rich game experience.

Using the techniques described in this paper, the massive multiplayer browser strategy game can attain a depth and strategic challenge unparalleled until now.

Referências bibliográficas

Gamasutra. (June 8, 2009). Analyst: Online Games Now \$11B of \$44B Worldwide Game Market. Retrieved June 20, 2009, from Gamasutra: The Art & Business of Making Games: http://www.gamasutra.com/php-bin/news_index.php?story=23954

Massively multiplayer online game. (July 16, 2009). In Wikipedia, the free encyclopedia. Retrieved July 23, 2009, from http://en.wikipedia.org/wiki/Massively_multiplayer_online_game

List of multiplayer browser games. (July 22, 2009). In Wikipedia, the free encyclopedia. Retrieved July 23, 2009, from http://en.wikipedia.org/wiki/List_of_multiplayer_browser_games

Olsen, J. (2004). Realtime Procedural Terrain Generation: Realtime Synthesis of Eroded Fractal Terrain for Use in Computer Games. In Oddlabs. Retrieved June 18, 2009, from http://oddlabs.com/download/terrain_generation.pdf

Prachyabrued, M., Roden, T. E., & Benton, R. G. (2007). Procedural generation of stylized 2D maps. In Proceedings of the international Conference on Advances in Computer Entertainment Technology (Salzburg, Austria, June 13 - 15, 2007), ACE '07, vol. 203, ACM, 147-150. doi:10.1145/1255047.1255077