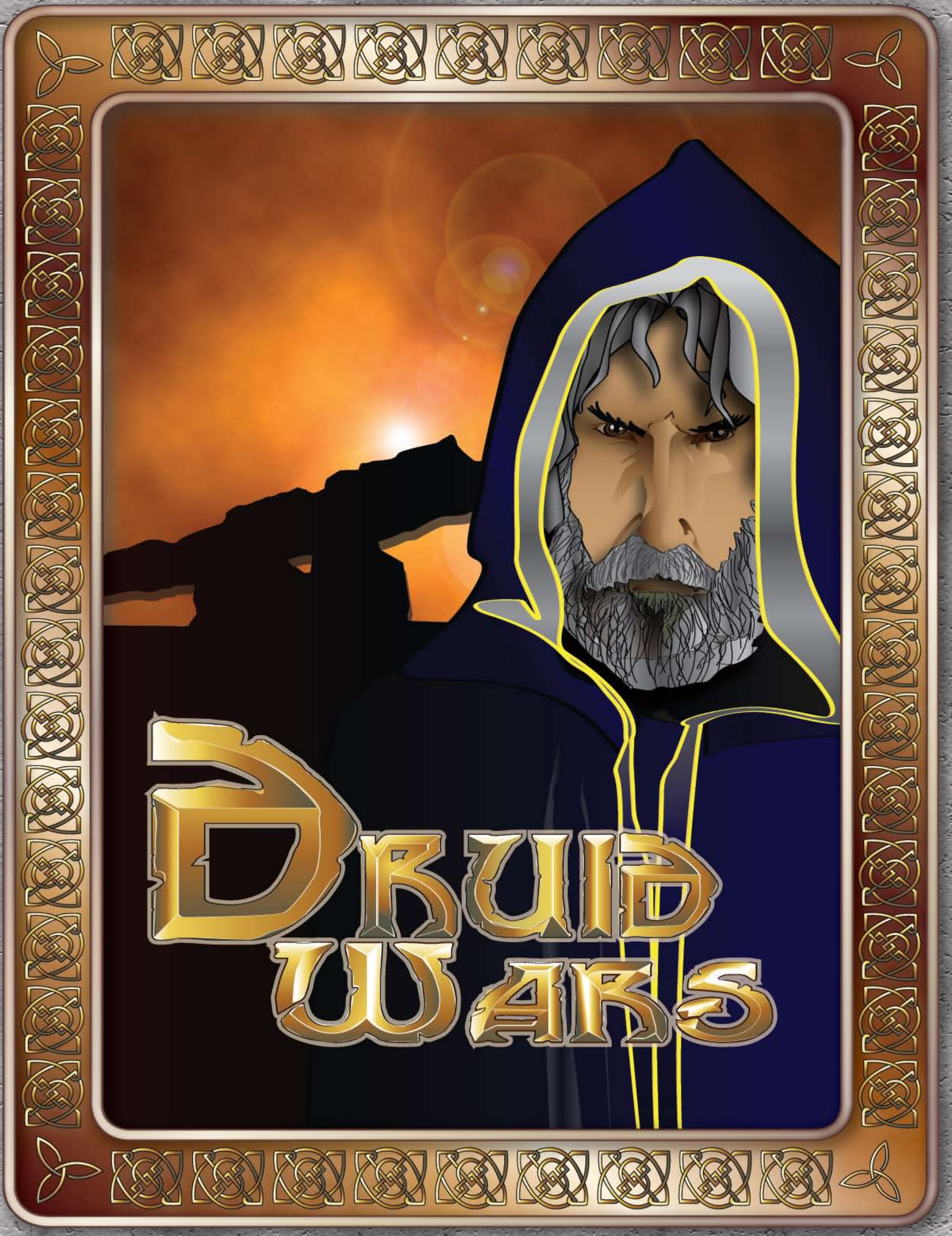

Druid Wars

Druid Wars

Project Overview

Table of Contents

1. Market Analysis
2. Monetization Plan
3. Game Overview
    - Game genre and style
    - Target Audience
    - The single-player game
    - The multi-player game

Project Phases

1. Preliminary Design
2. Proof of Concept
3. In-Depth Design
4. Alpha
5. Beta & Refinement
6. Rollout
7. Upgraded

Design Considerations

- Performance
- Fun
- Monetization
- Balance
- Security

⊕ Market Analysis

Ad revenue has not proven to be a viable revenue stream for individual developers, but rather seems to be profitable only for content aggregators. Since the aggregation market seems to be saturated, there doesn't seem to be much room there to move into.
Flash games typically have drawbacks that prevent them from being viable sources of revenue:

- If all of the logic is in the flash movie, it's very easy for aggregators to "steal" content.

- Security features such as domain checking are typically very easy to defeat with the use of decompilers. which leads to the questions, "how do we prevent theft of content?" and "how do we monetize content?" the best way to prevent theft of content is through the use 0f serverside code. if all or most of the game logic is handled with server-side code, the return on investment for content thieves drops to the point where theft isn't cost-effective, as the code would need to be reverse-engineered and rewritten at a greater cost than the initial development. It would be easier for them to simply copy the game from scratch. Monetizing content has been done successfully by several sites, including AdventureQuest (battleon.com), Travian (travian.com), and Gaia Online (gaiaonline.com).

⊕ AdventureQuest is a Flash-based browser RPG (roleplaying game). It is a single-player game, with no multiplayer content. The graphics and animation are fairly clean and have decent quality. Its main strength is in the quantity of its content; new quests and other content are added to the game on a weekly basis. Its main drawbacks are lack of multiplayer content and server queues (when the servers become "full", players must waitin a queue before they can play). AQ has a membership of xxx players, with an average of 10,000 users playing at any one time.

The game is free to play, but users may pay for enhancements in two ways:

1. Becoming a "Guardian", a one-time subscription fee of $20 or $25 that provides the following benefits:

- content not available to free accounts, including items, quests, locations, and others
- the ability to skip the server queue and always be able to log in and play immediately
- the $25 option provides quicker leveling.

2. By purchasing a specialized form of currency, which can then be used to purchase ingame items. This currency can only be purchased by users with a Guardian account.

AdventureQuest also features banner ads from ad networks.

⊕ **TRAVIAN** is a PHP-based strategy game with no Flash content. It is a multi-player game, with very slim single-player content. The graphics are of good quality. It currently has around 130k active players, with about 5-10% of its player base online at any given time.
The game is free to play, but users may pay for a specialized currency with which a user can purchase the following enhancements:

- Increased rate of resource gathering for a 7-day period
- Instantaneous completion of all production and research tasks
- A 7-day subscription to Travian Plus, which, in addition to several data reporting enhancements, allows the user to queue buildings for production rather than forcing the user to log on to order new production when a building is finished. This is a particularly clever enhancement, as users pay for what amounts to a reduction in server load!

It is interesting to note that Travian changed revenue models. At release, the only option available for purchase was Travian Plus, which was available in increments of several months. This new system allows more flexibility to the user, provides more choices, and allows the user to make impulse one-time purchases in addition to the subscription-style purchases.
Travian does not use banner ads from ad networks.

Gaia Online is an amalgam of gaming and social networking. It boasts the world's largest Internet messageboard, with over 14 million users. Its graphics are top notch, and has an anime style. It has Flash games and other content. Apart from the social aspects, the highly-customizable user avatar is a large draw to the site. Membership is free, but users can purchase in-game currency which can be used to buy clothing and accessories for their character. Some in-game items can only be purchased with real-world currency. Gaia also features banner ads from ad networks.

Monetization Plan

A common theme among the above-mentioned sites is the ability to sign up and play for free, with the option to purchase in-game currency which can be used to obtain certain advantages. Fantasy Conquest will follow a similar model:

- User accounts will be free to sign up and play.
- Banner ads will supplement site income.
- Users will be able to purchase subscriptions in monthly, three month, and six month increments that will include enhancements such as:
- removal of banner ads
- access to deluxe content
- the ability to attack/build/etc more than once at a time

Users will be able to purchase specialized currency which can be used for immediate purchases including:

- the completion of pending tasks
- reset of a character's name/appearance/skills/etc
- special offer items/equipment

Game Overview

Game Genre and Style

Fantasy Conquest is a real-time multi-player strategy game with a Celtic fantasy look and feel. It uses a combination of PHP and Flash for the user interface, with all of the game logic on the server side. The game world is a 500x500 grid of provinces, divided into four quadrants. Each province has several territories, and in the center of the province is a druid grove. Since the game world is not an infinite size, new servers will be added as game worlds become highly populated.

Target Audience

The target audience for Fantasy Conquest is early-20s to late-30s males with modest amounts of disposable income. Because the game can be played for a few minutes at a time and left for long periods, it is ideal for students and people who work in an office setting.

The game will appeal mainly to highly-competitive players who enjoy micro-management, but should hold some appeal for casual gamers as well.

The Single-Player Game

Each player starts the game with his main character, a level one Druid, who represents the leader of the user's army. The player is placed in the druid grove of a random unoccupied province in the quadrant of his choice. The druid grove is initially empty, and it is up to the player to strategically fill the grove with plants, to nurture and grow them to maturity, and thus build his personal power. From his grove, the druid will be able to recruit certain sylvan creatures based on which plants he chooses to grow. Maintaining a grove will appeal to players who enjoy micro-management. There will be many varieties of plants available, and the bonuses they provide will be determined by which quadrant of the grove they are planted in, what phase of the moon they're planted in, what soil and water is used, and the skill level of the druid. Certain types of plants will interact with surrounding plants, and plots will need to be weeded on a regular basis depending on the hardiness and aggressiveness of the specific plants. The bonuses for getting just the right setup will be good enough to appeal to micro-managers, but modest enough that casual players will be able to safely ignore them.

As the player progresses through the initial tutorials, he will gain the forces necessary to invade the surrounding territories. Territories will initially be non-player (NPC) controlled, and populated by enemy forces that will need to be defeated before the player can take control. Once the player controls a territory, he will be able to gather resources (gold or mana) from that territory at a certain rate per hour. In addition, certain territories will have buildings which can be improved for special bonuses, and some territories will have units which may be recruited into the player's army for mana or gold. Units require an upkeep of mana or gold on a per-hour basis to continue service, and thus players will be forced to balance income with expense.

Winning in combat will award your character with experience points, which will increase your druid's level. As your druid's level increases, you gain skill points which can be invested in several skills, including gardening, magic, and martial. In addition, characters will be able to research plants, spells, potions, soils, and waters. At any given moment in time your druid can be researching, gathering, cultivating, or exploring. Players can choose the path they want their characters to take, to be focused on magic rather than military might, to be a trader, builder, or conqueror.

After a player has conquered all of the territories in his own province, he can then attack other provinces, either NPC or player-owned.

The Multi-Player Game

Once a player has conquered his own province, he opens himself up both to attacking other provinces and being attacked by other players. A single player could ostensibly own several provinces and become very powerful. Players will be able to conquer provinces owned by other players, but a player's home province will be unconquerable, to prevent a player from being ejected from the game. Players will be able to trade resources and items through a trading post system. Players will also be able to form strategic alliances, which will allow for easier trading, social networking, and organized defense.

Project Phases

1. Preliminary Design

Development of an outline of all of the game's major functionality, including a timeline for each section. Will include meetings to discuss strategies for certain implementations. Estimated time of one week.

2. Proof of Concept

Development of a simplified online, real-time, multi-player strategy game, to demonstrate that the major design principles are achievable. Estimated time three weeks.

3. In-Depth Design

Creation of a full game document, including a description of all game mechanics, entity statistics, site design, and graphical assets. Estimated time six weeks.

4. Alpha

Development of all major game functionality tagged as release features. During this phase, testing will be limited to the development team plus a small (six to twelve) group of playtesters. Estimated time eight weeks.

5. Beta & Refinement

Development of all secondary game functionality tagged as release features. During this phase, several hundred players will be invited to a private beta test server. Game mechanics will be refined, bugfixed, and balanced based on playtester feedback and internal meetings. Estimated time four weeks.

6. Rollout

During this phase, at least two regular servers and one premium server will be opened for public registration. Beta testers from the previous phase will be given free subscriptions and in-game currency to encourage an initial player base. Aggressive free marketing will be pursued with the major general-interest Internet aggregators (reddit, delicio.us, digg, etc) as well as popular Internet messageboards. Low-cost ad network marketing will be considered. Estimated time one week.

7. Upgrades

Development of all major and secondary game functionality tagged as post-release. New functionality to be rolled out on a monthly basis.

⊕ Design Considerations

The following design considerations are in order of priority, from most to least important.

Performance

Performance is the single most important design consideration. Performance can be broken down into three parts: responsiveness, uptime, and stability. In terms of responsiveness, when a user performs a given game function, the server must respond quickly and report the results. In terms of uptime, there must be few (if any) periods of offline maintenance.

In terms of stability, unexpected downtime must have a minimum impact on actions already taken by players; in other words, players shouldn't "lose time" when servers crash or otherwise need to be offlined.

Players of free online games actually have less patience than purely for-pay games. Slowness, offline frequency, and instability will have a much larger impact on the game population than any other factor, including fun! After all, it doesn't matter how much fun a game is if you cannot play it or feel that any actions you take will be wiped out in an unexpected server crash. Thus, all design decisions must be made with performance in mind.

Fun

This consideration may seem obvious, but shouldn’t be underestimated. In addition to being appealing to hardcore strategists, it should also be fun for casual gamers as well. Thus, when a mechanic that appeals to one group is introduced, the question “can this be safely ignored by the other group?” must be analyzed and answered accordingly.

Monetization

Monetization is tightly linked to the other design considerations. For example, in the interests of good performance, a player may only execute one attack at a time, and must wait for his troops to return before attacking again. We can monetize this mechanic by allowing users who have purchased a subscription to make two attacks concurrently, ensuring that there is a balance between performance and income. When monetizing content, balance considerations should also be kept in mind. For example, it should not be possible to purchase levels or skill points with real-world currency, as it tips the game balance too far in favor of players with disposable income. Cash should not allow players to enter the endgame immediately, only accelerate advancement. For example, it would be acceptable to add experience bonuses to a subscription, which would allow players to advance through levels quicker. Likewise, purchasable items should favor cosmetics over functionality, and only provide small bonuses when compared to items that can be earned through normal gameplay.

Balance

Balance is an elusive goal in multiplayer games, but must be given due consideration. Game balance is the cause of the most heated discussions among players of online games. Game imbalance can lead to two significant problems:

- **Flavor Of The Month (FOTM):** as the development team works on balancing the game, the community discovers which game strategy gives the greatest advantage to the players. As a result, a significant percentage of players switch their game strategy to the “winning” strategy, and players lose any sense of uniqueness to their characters. When the development team rolls out changes that aren’t carefully balanced, the pendulum swings, and everybody switches to match the new mechanics. This leads to boredom, and players leave the game.

- **Frustration:** If I’ve chosen to focus on magic, and I discover that the military approach is significantly more powerful, I may find that I have no chance of competing against military players (the FOTM). If I’ve spent a lot of time investing in my character, I may not wish to start over again as a military player, become frustrated, and leave the game.

Security

Security can be separated into server security and security against exploits. Server security is a given, but exploits are a trickier problem. An exploit is when a player discovers he can manipulate the game in such a way that he gets something he normally wouldn’t be able to get. For example, if a user begins his name with a certain character (like a period or semicolon), he receives free gold. Exploits of this type could be a result of coding bugs, insecurity of the client, or testing features that were not properly removed. To prevent these kinds of problems, it is important that testing features are properly planned to be removed at beta/release, that any logic on the client side is also mirrored on the server side, and that bugs are properly tested for and fixed in a timely fashion. In addition, when planning or designing features, consideration should be given if at all possible to keep exploitation to a minimum if the mechanic is somehow broken.