****

**APOLLO Social Plan 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Revisions | By Whom | Date |
| 1.0 | Draft Document Created | Tarl Raney | 09.08.09 |

Contents

[Contents 3](#_Toc240204768)

[Goals 4](#_Toc240204769)

[Approach – Content vs Systems/Mechanics 4](#_Toc240204770)

[Research 4](#_Toc240204771)

[Implementation 5](#_Toc240204772)

[Testing 5](#_Toc240204773)

[Focus Testing 6](#_Toc240204774)

[Maintenance 6](#_Toc240204775)

[DLC 6](#_Toc240204776)

[Social Components 6](#_Toc240204777)

[Custom Challenges 7](#_Toc240204778)

[Hide the Trophy 7](#_Toc240204779)

[Bat Swarm Helpers 8](#_Toc240204780)

[Fight for the Cowl 9](#_Toc240204781)

[In-Game Gamertag Displays 9](#_Toc240204782)

[Community Contribution 10](#_Toc240204783)

[Online Stat Tracking and Display 11](#_Toc240204784)

[Where is your Gotham? 12](#_Toc240204785)

# Goals

The primary goal of the social components of the Apollo project is to add value to the game that extends the life of the product. By adding this extended value, the consumer is less likely to return the game to retailers after completing the campaign portion of the game, is less likely to rent the game rather than buy it and is more likely to tell his/her friends to purchase the game to share in the experience. In addition, the longer the player has the game in their possession (has not returned to retailer or rental store) the more likely they are to purchase additional content through micro-transaction channels such as Microsoft’s Xbox LIVE Marketplace or Sony’s PSN Marketplace.

The secondary goal of the social components is to help the developers establish a process by which they can continue to add valuable content through DLC and community updates in a quick and simple manner. This is a new opportunity for the game development community and it’s important that we establish best practices to capitalize on these opportunities in the most efficient manner possible.

# Approach – Content vs Systems/Mechanics

When discussing the types of DLC/Social components we’d like to implement in Apollo, we were very careful to concentrate on systems and mechanics rather than specific content. The reason being, we don’t know what story elements, characters, plot points, etc we’ll be able to use due to the fact that the filmmakers have not agreed to make a film, let alone what that film would be. As with our story and character planning, we’re doing everything in our power to move forward without locking ourselves into anything specific that could be forced to change.

To that end, this document deals mainly with the types of mechanics and systems we would like to implement. Any content attached to those ideas is simply there to help explain the system in a more understandable manner. As we gain a better understanding of what the film will be and what elements we will be able to use in the game we will begin to design the specific content to be used within the more ‘generic’ functional systems.

# Research

Much of what is discussed below is in the very early research stages. Monolith has never attempted to do the more advanced social components to our games before. Even adding DLC content to our games is a very new process and has only been done with one previous game, FEAR 2. We’ve learned a lot about how DLC should be approached and will be able to use that information to more efficiently build, test and submit DLC related content. The biggest risk comes from the new social components that will require back end servers, web based front end interfaces and (in the more extreme examples) iPhone applications.

On September 15th 2009, Monolith’s development team will be meeting with [Agora Games](http://www.agoragames.com/) to learn about the services they provide and what things will be required from our own development team to take advantage of that support.

# Implementation

Once the meeting with Agora Games happens, we’ll be able to more definitively talk about the implementation of our social components. DLC implementation is a bit more straight-forward as we have experience with this from FEAR 2.

Through our experience with FEAR 2’s DLC development, we found a number of areas where our technology is not as robust as it needs to be to support certain types of DLC content. As we’ve discovered these weak areas, we’ve been working to refactor those areas to better allow updates for DLC content.

One example of this is the String Database that houses all of the text in the game. Currently, these files are separated on a per language basis and do not allow DLC content to update or add individual strings within these files. This means that if we want to release DLC content with new strings, we must update the full string tables for every language we support. For DLC content this is fine, but if the strings have to be part of a Title Update on the Xbox 360, all of these string files must be included in the package which has a 4MB size limit. All of the string files for every language required for the Title Update tend to come very close to that size on their own leaving very little room for the other files required for the Title Update.

This is one of the systems that we have plans to refactor to allow individual strings to be updated and added to Title Updates. This will allow for much smaller file sizes as only the required strings will be included, not the whole string table. As we move further into production of Apollo, we’ll detail the changes being made to various systems for the purposes of DLC and Social Components. The most up to date technical details will be included in the TDD as they are made available.

# Testing

There isn’t much detail that can be added to this section until we have more specific details on the systems that are being updated and how they will be changed. However, we can say that we will be designating new elements of the game (characters, game systems, levels, etc) as ‘DLC Test Beds’ as they are put into the game during development. As those areas of the game are developed and things change, we will treat them as DLC content to allow QA to test the DLC content delivery methods/systems during development. As each DLC Test Bed is brought further along it’s iteration path, we’ll designate new Test Bed content to allow us to fully test the DLC systems with ever-changing content to mirror the way live content would be dealt with.

# Focus Testing

Focus Testing will follow the standard Playtest processes used by WBG publishing departments. For a full list of what will be tested and the success criteria, please see the Playtest Plan. (Will be updated as we define what DLC systems will be)

# Maintenance

As with Implementation, Testing and Focus Testing, the Maintenance requirements are not known currently. This section will be updated as we determine what systems and content are planned and how they will need to be maintained.

### DLC

As previously mentioned, the DLC plans are only ideas for the type of content, not the specific content itself. Below is a list of types of content we plan to be able to support.

Planned DLC content types to support

* Additional levels (Hubs or districts in Apollo)
* Additional characters (either in new levels or added to shipping levels)
* Additional gameplay elements (new quests, storylines or challenges; both in new levels and added to shipping levels)
* Additional player controls (new moves, control schemes, gadgets, etc chosen through tech tree upgrades)

### Social Components

During our brainstorming sessions for social mechanics, we were careful to keep the discussion to the mechanics and systems rather than the specific gameplay implementations. As with the DLC ideas, these are possibilities for the content that could be used within the proposed mechanic. As we move forward and have a better idea of the exact content that will be used within the mechanics we will update this doc as well as the design and technical docs associated with the social components.

The ideas that follow are from these brainstorming sessions. The list of ideas has been culled to only the most compelling ideas, that meet the goal of extending value and (at first glance) appear to be technically feasible within the timeframe of the Apollo project. Before any of these ideas can begin to be worked on, there will need to be more research and design done.

Social Component Brainstorm List

## Custom Challenges

In addition to the planned challenge missions that will be designed by the game’s creators, we are exploring the idea of allowing players to set up challenges for their friends and the Apollo community. Using a set of pre-configured tools, the player would be able to lay down waypoints for races or time challenges, place spawn points for enemy encounters and perhaps even set up ‘crime scenes’ by laying down evidence that must be found and investigated.

The challenge would be set up in the game hub of the player’s choice using as much of the hub as is possible (TBD). Once the player has created and tested the challenge to his liking, he can save it and share it with either friends or the Apollo community through the WB back end server system. When someone either receives a challenge from a friend or downloads one from the community site, they will be able to play the challenge and post their score or time to the leaderboard for that challenge.

This idea could also be extended to even allow users to create their own quests in a limited fashion.

*Required Tech and Content*

* Back end server to store created challenge files, stats and custom leaderboards
* In-game challenge creation system
  + Ability to place items in-game
  + Ability to save created challenge file
* Front end interface for in-game content creation system
* Ability to change or add to these systems in a Title Update or Patch
* Ability to load ‘blank’ level without campaign gameplay elements
* Ability to send challenges to friends
* Ability to find, download and run challenges found on the Apollo community site

*Risks*

* Lots of new tech needed
* Will require a lot of QA testing
* User created content carries it’s own list of risks (will limit this with the parameters of the system we create, but it is still ‘user created’ content)

## Hide the Trophy

Along similar lines to creating challenges, is the idea of allowing players to hide objects in the world for their friends to find. There are different ways to approach this, but the general idea is that players will find an object in the world (perhaps a trophy) and then be tasked with finding a hiding spot (in the same Hub the object was found in) in a set amount of time. Once the object is hidden, they will then be able to challenge their friends to find the item.

*Required Tech and Content*

* Back end server to store item location save files to send to friends
* In-game challenge creation system
  + Ability to place items in-game
  + Ability to save created location file
* Front end interface for in-game item placement system
* Ability to change or add to these systems in a Title Update or Patch
* Ability to send challenges to friends
* Ability to find, download and run challenges found on the Apollo community site

*Risks*

* Lots of new tech needed
* Will require a lot of QA testing

## Bat Swarm Helpers

Similar to the co-op orbs in Fable 2, one idea for Apollo is to be able to see where Gotham your friends are. Rather than showing a glowing orb moving through the world, we thought it would be more appropriate to see a small swarm of bats indicating that someone on your friends list is playing the game at the same time and is in a similar location to you.

These swarms of bats, wouldn’t just be cosmetic either. When more swarms of bats are nearby, there might be a buff to Batman’s abilities or a debuff to enemies around him. One idea was to allow the bats to increase the fear level of enemies nearby so that the more players you have nearby, the easier it is for Batman to manipulate them.

*Required Tech and Content*

* Ability to track location of players on friends list
* Ability to send and receive location data
* Ability to limit how many people can be shown as bat swarms in one user’s game
* Ability to use external location data to manipulate in-game settings (other people nearby contribute to current player’s enemy AI fear level)
* Ability to choose which friends can and can’t see users location
* Ability for user to choose which friends they want to see in their game

*Risks*

* Lots of new tech needed
* Will require a lot of QA testing

## Fight for the Cowl

The fight for the cowl is an idea to allow players to compete against each other in the single player campaign and gain bonuses based on their performance. Players would be rated in the leaderboards on various stats or events. The players that do the best would gain a reward for a limited amount of time. After a set amount of time (1 week maybe) the leaderboards would reset and a new winner would be crowned after the specified time.

Using combat as an example, a player may be good enough to be the best at chaining combos together while fighting through the single player campaign. After a week, he is still at the top of the leaderboards. The leaderboard is reset, he is crowned the champ for that week and is given a boost to his XP gained during the next week. This would be repeated every week allowing everyone to have a chance to gain the XP boost. There would also be more than one set of criteria for players to try and achieve.

An additional idea to this is to tie it into our community site and advertise new challenges each week. We could possibly cycle through 6 different challenges over a 6 week period so that every week is a different challenge and new groups of people will be given a chance to compete. Rewards for these resetting challenges could also be things like gamer pictures, themes and avatar accessories, not only in-game rewards.

*Required Tech and Content*

* Ability to reset leaderboards on a set schedule
* Ability to reward players for their specific placement on a given leaderboard at a specific time (Reaching first on the leaderboard on Monday does not give a reward if the leaderboard resets on Friday)
  + In-game rewards
  + External rewards (Avatar accessories, themes, etc)
* Ability to set challenges and tie them into the community site (WoW weekly battleground weekends are a good example)
* Ability to update and add challenges through DLC and Title Updates

*Risks*

* Lots of new tech needed
* How do we keep players engaged once they’ve unlocked the reward for all challenges?
* Will require a lot of QA testing

## In-Game Gamertag Displays

A more general idea that could be used in many different ways, is to add the ability to dynamically show Friend’s gamertag info in-game. When a friend gains an achievement, it could be shown on a billboard in the city or on a TV screen in a shop window. Shadow Complex for Xbox 360 does a great job of showing how a player is doing towards some of their achievements compared to their friends. We’d like to take that a bit further and allow players to check their status vs their friends’ at any time on the Bat Computer.

*Required Tech and Content*

* Ability to display gamertag info in-game
* Ability to track achievement progress of other players and send that info to friends

*Risks*

* New tech needed
* Will require a lot of QA testing

## Community Contribution

Another idea that’s been used to encourage community involvement is the common goal of unlocking additional content that everyone can participate in. Battlefield 1943 encouraged players to keep playing until they collectively achieved 43 Million kills in the game. When that happened, a new game mode and map were unlocked for everyone.

Again, taking that idea a little further, we’d like to be able to hint at the first DLC content by dropping in-game clues that players would be able to somehow participate in. At the same time, giving players a chance to affect the game world of other players with their actions. Trying to stay away from getting too tied to a specific set of content, one idea that has really resonated on the development team is having the Joker involved in the first DLC. As players complete the campaign that ships on the disc, they may find optional evidence that is tracked on the backend servers. The more people that find this optional content, the more hints to it appear around the in-game world for everyone. As more and more people find these hints, a mystery begins to unravel.

For instance, after defeating the Penguin in his hideout, the observant player may notice an optional piece of evidence such as a joker playing card. If enough people playing the game and connected online find the same evidence, additional content will be turned on in-game. Perhaps the tell-tale Joker face paint graffiti begins to pop up around the city. More and more people find additional evidence that opens up more in-game content that eventually reveal a web-site that has the information about our DLC plans.

This is an idea that obviously needs a lot more time devoted to research and design, but could be a very effective way to create buzz through in-game and real world viral advertising.

Aside from this specific idea, the systems that are desired would allow players to collectively work towards a common goal and player’s actions in their own version of Gotham may affect the world that other people see in their own Gotham.

*Required Tech and Content*

* Ability to collect conglomerate stats from all players online
* Ability to change the world based on the level of participation in the conglomerate stats collection

*Risks*

* Lots of new tech needed
* Will require a lot of QA testing

## Online Stat Tracking and Display

Similar to the Rockstar Social Club, we want to have the ability to allow players to view (and possibly change) their in-game stats outside of the game. In the most extreme example of this idea, we would love to have the ability to have an iPhone app, web site and even Facebook app to allow players to see the stats of their Batman, compare to their friends’ Batman and even change their talent distributions or gadget loadouts for use next time they log into the game.

Also similar to Rockstar Social Club, we want to be able to show conglomerate stats for all players playing Apollo. Being able to show how many times Batman has locked up the Penguin, how many crimes have been thwarted, etc. Again, these are things that tie the community together and can be used to help draw them further into the game and want to bring their friends along.

*Required Tech and Content*

* Web site to display stats
* Ability to tie external apps into backend database (iPhone, Facebook, etc)
* Ability to design and create iPhone, etc apps
* Ability to collect and display conglomerate stats from all players online

*Risks*

* Lots of new tech needed
* How do we keep players engaged once they’ve unlocked the reward for all challenges?
* Will require a lot of QA testing

## Where is your Gotham?

Tying into the community aspect of the conglomerate stats, another idea is to display the location of the players currently playing in the real world. Similar to the [Ocarina](http://ocarina.smule.com/) iPhone app, which allows users to see dots on a map that indicate where, other people playing the game are located in real life.

*Required Tech and Content*

* Ability to view player’s real world location

*Risks*

* New tech needed
* Will require a lot of QA testing

## Timeline Events

This would provide the ability for developers to set up in-game events based on the real world calendar to tie into real world events. This can range from holidays to timed events that reward in-game advantages. For example players that play during a pre-designated weekend would get double damage between Friday and Monday. Anyone playing on Christmas day would see special in-game art showing Gotham decorated for the holidays or jack-o-lanterns on Halloween.

*Required Tech and Content*

* Ability to change content based on real world date and time
* Ability to turn on/off content or temporary stat changes based on real world date and time

*Risks*

* New tech needed
* Will require a lot of QA testing

## Riddler Mail

This is a generic system that would allow the game to send email (both in-game Bat-computer mail and real world email if opted into by players) from the characters in our game to help keep the player up to date on the status of Gotham while they are offline.

*Required Tech and Content*

* Ability to send emails from the backend server
* Ability to receive notice that the player should get in-game mail when logging in if not opted into real world email delivery

*Risks*

* New tech needed
* Will require a lot of QA testing