

Carefree Games
**Spring
 2007**
Phillip Kerman
phillipkerman.com LLC

OVERVIEW

My presentation has two main points:

- 1, to point out that there's a huge burgeoning market for casual games.
- 2, to highlight some of the terms and technical challenges involved in making games. Infiltrating an industry is always easiest if you speak their language. For me, it's also interesting because it's not as though Flash programming is any different when developing games, but the priorities do differ.

I should also note that because I really don't have *that* much experience building games, this presentation draws examples from the two large scale games I did program: Sudoku and Jigsaw for MSN Games (linked below). You can think of this discussion as "Phillip's whacky adventures in the world of gaming"... but it isn't so much a case study if that's what you expect. I'm going to talk about the opportunities and the technical challenges that I discovered.

DEFINITION

Casual Games are generally defined as downloadable games (less than 50 megabytes) that include some form of try/buy (such as a trial period, fixed number of uses, or premium content for a price). That is, free until it times out or you want the "full" version.

Adver-games are fully sponsored free games serving as a vehicle for promotion and advertising. There are many variations here, but the theme is that users don't "own" the game and they play while something is being promoted. This can include an anonymous game that's driven purely by ad revenue in banners outside the game. Often adver-games are produced specifically for the client. Simple things like Dyson's Telescope game or something advanced such as the Burger King games for Xbox.

The problem with both these definitions is it doesn't talk about the game experience. The game type I'd call "**carefree games**" could be either of the first two (a casual game with try/buy or an adver-game)... but the key is how the user plays the game. With a carefree game users get up and running quickly (without a long story or difficult learning curve) and they don't mind dropping out in the middle of a game as they're usually playing for fun (as a distraction or to kill time).

What I'm talking about is *not* the typical high-budget, so called "triple A" (AAA), game. While AAA games are part of a huge market (that mainly targets young males), it's actually a rather flat market. In addition, it's a mature market in that it's impossible to jump in and make a cheap game quickly. I don't want to discount the hardcore gaming market but the new darling is the burgeoning casual game market (which I'm going to use to mean any game that can be played with very little investment—including adver-games, hence the term "carefree"). Note that this by no means implies casual games are cheap to produce or not sophisticated (as I'll address later when discussing the impending arms race). This is a growth market primarily because it targets a new audience—older people (like my age—not quite that old)... and predominately women.

Just listen to these descriptions of two casual games and see who you think they target:

Dreamday Wedding, where you “make all the wedding arrangements in this romantic story game! It’s a romantic seek-and-find adventure featuring beautiful graphics, unique puzzle games, and extra ‘Wedding Crisis’ levels sure to challenge you!”

Flower Shop: Big City Break, where you “help Meg turn a small flower shop into a ‘blooming’ success!”

The fact this is a hot market was actually news to me. I attended Microsoft **GameFest** last year and figured I’d be totally out of my element. I guess I really was when you consider the Graphics track wasn’t “101 Photoshop tips” but rather “Direct 3D and GPU Performance”. Seriously, I thought “casual games” meant “lite games”. But it was just the opposite. This whole point about the growth opportunities in a new target user was highlighted in the keynote and throughout the conference. Hot Hot Hot... it doesn’t take long to confirm this once you look into it.

OPPORTUNITIES

In the presentation I threw around a bunch of numbers. Here’s an overview:

Casual games usually retail for \$19.95. Through various promotions and subscriptions, most players end up paying \$10-12 per game. The portal site (that hosts your game—such as: oberongames.com, bigfishgames.com, zone.msn.com, realarcade.com, or king.com) will typically take 65-70% of that gross. (See XboxLive Arcade later because developers keep a bigger percentage.) At 30-35% that works out to about \$3 per purchased game for the developer.

The factor by which a game’s success is measured is the conversion rate—or what percentage of those who download the trial version end up really buying it. Currently, if you see a conversion rate greater than 2% your game is a hit, 1%-2% is about average, and less than 1% is poor.

The life of a really successful game (not necessarily a blockbuster, but a very respectful game) might expect this sort of history:

In the first weekend: 60,000 downloads, 2.4% conversion, or \$15,000 gross.

First full month: \$100,000 gross. Next few months see the gross gradually drop to about \$50,000 month to finally settle into about \$10,000 per month. Realize, too, that while the game is hot the portal will continue to promote it. At the above numbers, the developer will net 75,000 - \$100,000 in the first few months and then (hopefully slowly) slide down to about \$3000/month.

While the “arms race” has already started in the casual games market, a typical game costs between \$100,000 and \$200,000 to produce. Be warned that as the audience becomes sophisticated their expectations grow. I’m not sure if this means we’ll have 3D jigsaw with smoke coming off the pieces, but you get the idea.

Money facts about Adver-games:

--usually work-for-hire

--simply skinning / branding an existing game isn't popular anymore

--there are still lots of ways to make money without building it custom for one client:

--show an ad while the games loads

--show banner ads while the game plays

--incorporate game breaks where an ad appears.

--sell sponsorships (like a movie still as the jigsaw-of-the-day)

Facts about Xbox Live Arcade (XBLA):

- 6 million xbox users
- try buy... very “frictionless”
- they say \$75,000 - \$300,000 to produce (so you know that must go up)
- certification (can be a bitch—I’ve heard)
- games sell for \$10
- conversion rates around 30-35%!!!
- better revenue share: 50/50 or better (compared to the typical 30-35% going to the developer)

Facts about MSN (from “Developing Casual Games for MSN Messenger” presentation at GameFest 2006)

- 250 million users
- 15 million unique game users per month
- Potential for 30-50 million players
- 30% average yearly growth

Ultimately, I’m just a developer... I just make games. I’m sure if you’re ready to fund such a project you’ll want to get to know more about the nuts and bolts and dollars and cents.

NOMENCLATURE

I already threw out a few terms like “conversion rates” and “frictionless buy process” but before digging into more technical specifics there are a few more concepts I want to mention.

BRANDING

There are all kinds of crazy opportunities to leverage how consumers identify with certain brands. For example, MSN has a Bicycle brand playing cards Hearts game... there’s a “Pat Sejack wheel of fortune game”. To me, it sometimes makes these games look like graphics on Nascar race cars—but it apparently really helps. It’s more than simply adver-gaming. For example, the Burger King Xbox game is \$4 plus a “value” meal. (Cheesy?-- they sold over 2 million of these! Double cheese.)

FRICTIONLESS

If you’re going to sell your game, making it easy to buy is important. Xbox Live Arcade has a slick (slippery) way to purchase games. Basically, your parents stock up your debit account and then with one click you can buy games.

UPSELL

This isn’t super tough, but—sort of as an extension of “frictionless”—if you’re trying to get users to upgrade to a full version there are several ways to sell them on that upgrade (that is, “upsell” them). You can expose all the off-limits features but, when selected, inform the user they’ll have to upgrade to do so (and give them an easy way to do so). You can also call attention to the missing features or various benefits to buying the full version. Also, think about a phased deployment where you may not disclose to the early users that, later, they’ll be offered to upgrade.

PORTALS

Sort of like phone providers, you really have to go through them. The thing is, that I’m pretty confident installing apps that would otherwise be suspect malware from certain sites because I believe they’re reputable. I don’t mean to defend or criticize portals—they’re a fact and, unless you’ve got your own idea to start one up, you’ll probably want to go through one. It’s sort of like book publishers. I could publish my own book but it wouldn’t be in the stores without the publisher’s sales force. Portals also offer important social network aspects such as chat, presence (where you can see you see your buddies online), and the ability to invite buddies to a game. The consoles are going nuts in this regard.

BADGES

For whatever reason, a lot of gamers feel compelled to measure their success—and show it off in the form of badges. Badges drive replay. Achievements are really a more accurate term for how gamers measure their success as badges are more “souvenirs” where achievements are skill based. To deliver this... and to establish “currency”, it sure helps to have an infrastructure (read: portal).

TECHNICAL CHALLENGES (AND TERMS)

Programming is programming and building a game really shouldn't be any different. The difference, however, is in priorities.

USER EXPERIENCE

The whole “user experience” thing is priority #2—second only to making a fun game. (Let me state now, your game won't be a success if it isn't fun.) Compare this to how “user experience” has become a “feel good” business concept on typical websites. Sure, a bank wants your banking experience to be easy and fun, but I hope their priority is security. It's just with games—and especially with casual games where users can easily walk away—you can't overdo it when improving the user experience.

USABILITY

Usability testing needs to be approached scientifically. They did the whole 2-way mirror thing with my MSN games. And they did it early. I was shocked at how rough the working prototype they showed the people with whom they tested. These were uglier than I'd show most clients! They even did paper walkthroughs for variations on selection screens... the user would touch the button on paper and the tester would replace the sheet. The point is that you can't just use your brother in law. (Though, I do believe you can and should supplement your testing with kids—if you have any handy.)

LIVING IN A FRAMEWORK

Because quite often your game will need to live inside someone else's framework, I made a point that you don't want to get bogged down trying to improve the framework. Naturally if you're hired to improve the framework, that would be fine. Often there are reasons for why the framework is the way it is, and ultimately, you can't change it. So, basically, just deal.

LIVING IN REALITY

Unfortunately, people have a wide range of hardware and internet connections. Here are just a few limits that I faced in the MSN two-player games:

I could send no more than 120 messages per minute.

Each packet was limited to 1500 bytes.

When I sent a message to the other player, there was no callback to confirm the message was received.

The saved game datastore was limited to 2000 bytes (but that was effectively 1500 bytes because I wanted to let players share their saved games).

The only promise was that messages would arrive in same order they were sent.

MULTI PLAYER

There are a few things to think about when building mutli-player games:

Grieving situations where one player can effectively ruin the game for the other player (say there's a confirmation dialog that one player simply ignores—it's probably best to not make the game hang)

You need to handle race conditions (where two players effectively reach for the same move).

Turn-based games make programming *way* easier.

Don't underestimate how much this involves (similar to getting a game certified)

Turn based makes it **WAY** easier

PORTING

- Plan from the beginning (like most things)
- Think decoupling (factory pattern)
- Include game modes

LOCALIZATION

- Only 20% of MSN gamers are in USA
- Don't worry, be crappy (for example, it was vastly easier to leave dynamic text aliased so we did)
- I had to come up for a name-token replacement strategy: "Phillip is waiting" might effectively read "The waiting person is Phillip" when translated. So, I can't just always put the user name in front of the translated version of "is waiting".
- Another case of not trying to save the world was when I tried to improve the XML format MSN uses for localized text. The bottom line: it wasn't going to change so I just had to deal.

CHEATS

- For example, the Sudoku game had a key to reveal all the correct answers...and another key to go ahead and fill in all the correct answers. The jigsaw game had a cheat that labeled each piece with its row and column., another cheat snapped pieces together, and still another created a board with only 16 pieces total.
- I really miss those cheats! Though, when I play my mom, she sometimes claims they're still in place.

LINKS:

- Jigsaw (one player web): <http://zone.msn.com/en/jigsaw/>
- Jigsaw Too (two player in Messenger): <http://zone.msn.com/en/general/article/mjs2messengerlaunch.htm>
- Sudoku (one player web): <http://zone.msn.com/en/sudoku/>
- Sudoku Too (two player in Messenger): <http://zone.msn.com/en/general/article/msdkmessengerlaunch.htm>
- (For a personal tour of the messenger games, IM me at sudoku9000@hotmail.com)

- Dyson Telescope Game: <http://www.dyson.com/game/play.asp>
- Dyson Ball Game: <http://www.dyson.com/games/ball/default.asp>

Burger King Xbox games: <http://www.xbox.com/en-US/promotions/burgerking/default.htm>

Game Developer Organization: www.igda.org

Info: http://en.wikipedia.org/wiki/Casual_game

GameFest 2006 downloads: <http://www.tinyurl.com/yjj92m>

Game Frameworks:

- <http://garagegames.com/developer/torque/>
- <http://developer.popcap.com/>
- <http://xna.com/>

Papervision 3D: <http://www.osflash.org/papervision3d>

Conferences

- GameFest: <http://microsoftgamefest.com/>
- Casual Connect: <http://seattle.casualconnect.org/>
- Game Developers Conference: <http://www.gdconf.com/>

Contact to make deals developing for MSN: mcg@microsoft.com