

Introduction to game design

Electronic games are growing in popularity—and not just with kids and teens. With the ubiquity of mobile devices, you'll find that just about everyone from small children to grandparents has a favorite online game.

Gaming trends

With the popularity of mobile devices, several new trends are emerging.

- There is renewed demand for simple and repetitive games.
- Lower-budget games created by individuals and small teams can be financially successful.
- You can find them on game arcade sites such as Newgrounds.com and AddictingGames.com.
- You can choose from dozens of game genres: action, adventure, casino, puzzle, role-playing, and more.
- For many developers, the Adobe Flash platform is the preferred tool for game development.



Figure 1 Mobile game examples

Principles of game design and development

One way to get started developing games is to use Adobe Flash Builder or Adobe Flash Professional (development environments) to develop a single-player game (that is, one that does not need a real-time multi-player server) that is published in the SWF file format and runs in Flash Player (a runtime) in a web browser.

Runtimes

A *runtime* is a software component designed to support the execution of computer programs. The Adobe Flash Platform includes two runtimes: Adobe Flash Player and Adobe AIR.

Adobe Flash Player

Adobe Flash Player is a browser plug-in that provides a multi-platform client runtime. You can build your game once and it will look the same in virtually any web browser and on Windows, Mac OS, or Linux operating systems. Flash Player plays content that has been compiled into an SWF file. Flash Player is licensed free of charge so end users can start playing your game without additional costs.

Adobe AIR

Adobe AIR lets you view SWF content on your desktop without a web browser. You need to have the AIR runtime installed to run AIR files. The AIR runtime is licensed free of charge to end users.

Development environments

Adobe Flash Professional CS6 and Adobe Flash Builder 4.6. are development environments for creating SWF-based games and other online content.

Adobe Flash Professional CS6

In Flash Professional CS6, you can see your layouts and animations as you work with them. For that reason, it's very popular with developers who create visual games. It strikes a good balance between visual design and code design.

Adobe Flash Builder 4.6

Adobe Flash Builder 4.6 is used by developers who create Rich Internet Applications (RIAs) that connect to a data source or web service. For example, an application that regularly updates to show current movie times could be created by using Flash Builder. Flash Builder is a popular tool for creating web applications and mobile apps, including games.

Many developers use a combination of Flash Builder and Flash Professional when developing games. For example, you can use Flash Professional to design the graphic elements of your game (characters, maps, effects, and so on) while your developer works in Flash Builder to build the final game controls and connect the game to a server and other players.

Key components of a game

Whether you're building a simple single-player game or a multi-player online gaming world, there are several components that are common to most games.

Input: Input refers to how the user controls the game. Basic games are typically controlled with the mouse, the keyboard arrow keys, or both. On mobile devices, touch gestures are the most common input method and the Flash Platform has robust touch support.

Sprites: In the gaming world, a sprite is a visual elements such as an alien or fish moving across the screen.

Note: Game sprites should not be confused with the ActionScript 3.0 class "flash.display.Sprite."

Sounds: Sound effects in games accent important events to enhance the game play experience. Sounds provide an important element of feedback to the player and make games interesting and fun.

Collision detection: This is the code that senses when any two sprites touch on screen, which is crucial for most action-oriented games.

Game loop: This is a term used by game developers to describe the programming logic that controls the game experience. ActionScript is the programming language you use to execute the game loop in Flash games.

Game design and development workflow

Whether you're an individual developer creating your first single-player game, or a large corporate team designing a complex multi-player environment, you'll follow a similar workflow for game design and development (**Figure 2**).

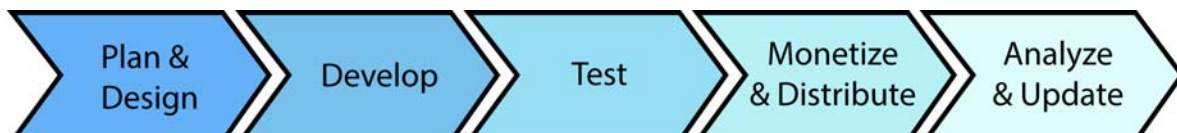


Figure 2 Game design and development workflow

Plan and design

The most successful games are the product of great ideas followed by great execution. This usually involves a combination of sound design principles, creativity, and strong technical development skills.

Each team may work differently to develop new ideas, but the basic planning process is as follows:

1. Brainstorm to develop ideas for the game.
2. Create a storyboard or map of the game world.
3. Design your game in a game design document.

Brainstorm

Your inspiration for great games can come from anywhere—dreams, movies, books, nature, work, cities, friends, family. Sometimes the best ideas are made even better by collaborative brainstorm sessions. Write down all ideas, including the outrageous or silly ones.

Storyboards and maps

During the storyboard and mapping process you begin to sketch out characters, backgrounds, screens, *game mechanics*, and more. Game mechanics are the core set of rules that define what makes a game unique. A game features a variety of game mechanics, such as keyboard input, turns, decisions, risks and rewards. During this stage, the artwork visually communicates your ideas. This is not your final artwork. As your game concept takes shape, add details to the game world. Describe the challenges to the game player. Describe the rewards. Does your game have characters? What is the plot or story? What motivates the character to do what he does? Does the game player view the action from above or from the side? Does gravity exist in the game? Can the characters be injured and die? If so, how?

Design document

The design document describes the details of the game, including the technical requirements for development. In a team development environment, the design document is especially important. It includes final decisions regarding the game's visuals, game mechanics, and menus. It helps ensure all team members are on the same page during production.

- Who is your target audience? Are they experienced gamers with patience for difficult game play or casual gamers who want something easier?
- What devices will you target? A big browser window, a small phone screen, or both? Consider the unique characteristics of each delivery screen.
- What spoken languages will be included? How will the translation of the game be organized? What logos, art, menus, and other text will be affected? If the game will be localized into more than one language, what is your plan to accommodate them?

Develop

The following are the basic steps to developing a game with Flash technology:

1. Follow your design document.
2. Build the game in stages, creating standalone demo projects. This will help to identify and trouble-shoot aspects of the game that are difficult to implement or create usability issues during play.
3. Create an assets library of artwork in Flash or other Creative Suite products. Locate royalty-free sounds online.
4. Construct the menus. Games often feature screens outside of the game itself such as the introduction, instructions, and Game Over screen.
5. Test the game repeatedly throughout the development process to be sure it's enjoyable and challenging.

Test

Testing can take many forms. If you are an individual developer, the testing process can involve you and a few friends playing the game. Try to have open eyes and an open mind, so you can see new problems and new solutions.

Larger game development teams often have a dedicated quality assurance (QA) team of testers. This group plays the game and builds a list of changes and suggestions.

Regardless of the team size, a good testing process will answer these fundamental questions:

- Is it fun?
- Does it work (is it free of bugs)?
- Does it perform as expected on the desired platforms?

You can use Flash Professional CS6 to target many devices, but designing, developing and testing for multiple screens presents a unique set of challenges.

For web games, test the game on a variety of computers and web browsers. Ensure it works well in all cases. Even when the game works great on your personal machine, it is important to keep in mind the computers and devices of your audience.

For mobile phones and tablets, testing can be more difficult. Differences between devices (for example, screen size, screen density, and input schemes) and the sheer number of devices available can complicate the process. Flash Professional includes mobile device support to simulate, test, and publish applications to a wide array of devices. For example, you can use the AIR mobile simulator to test mobile application interactions. Flash Builder and Flash Professional ship with a debug version of Flash Player that provides additional information (such as output tracing and redraw regions) as you develop and test. Flash Professional also ships with Device Central, a package of emulators that simulate popular mobile devices right on your computer.

Test on each physical device if possible.

Monetize

There are two main approaches to turning games into revenue: offer games that are free to play or offer premium games that cost money to play.

There are several ways to monetize free games:

- *Advertisements:* Place banner advertising on the HTML page surrounding the game or within the game itself. Advertising deals are typically paid to game developers based on the number of ads shown (impressions) or the quantity of traffic generated (clicks).
- *Sponsorship:* Place advertising, links, logos, and marketing messages within your game for a paying sponsor.
- *Licensing:* Place the advertising, links, logos, and marketing messages within your game for a paying sponsor and restrict the game's availability to a single site.

There are also several ways that you can make money by selling a Flash game directly to the video game community:

- *Straight sale:* Sell the complete game for one up-front cost.
- *Freemium:* Offer a free trial and then let users pay for upgrading to a complete version.
- *Micro-transactions:* Encourage users to convert real-world money into virtual game currency to unlock characters, abilities, levels, or other content. Micro-transactions and freemium games are sometimes used together.
- *Subscriptions:* Players can continue playing if they make regular payments (usually on a monthly basis).

Distribute

Once you've developed your game, select a strategy for getting the game in the hands of players. The Flash Platform offers several deployment options to reach your intended audience.

Flash Platform deployment options via Flash Player:

- Web browser (PC, Mac, Linux, mobile)
- Mobile browser (Android, BlackBerry)

Note: The Apple iOS Safari web browser does not support any version of Flash Player. To deploy your game on an iOS device, package the game as a standalone application using Adobe AIR.

Flash Platform deployment options via Adobe AIR technology:

- Desktop (PC, Mac, Linux)
- Android (Phones and tablets)
- Apple iOS (Phones and tablets)
- BlackBerry Tablet OS (tablets)
- Internet-enabled televisions and appliances

The distribution and monetization strategy for your game is best defined during the brainstorming stage. The distribution strategy affects your game because the target device determines screen sizes and input styles. The monetization strategy you choose also affects game content. You may need to implement advertisement integration, links or buttons to sponsors, virtual item sales, and more.

Analyze and update

Consider the following steps to analyze and update your application before you publish it:

- Conduct usability studies and focus groups to gather user feedback.
- Update your game often after its initial launch to ensure it has the longest life possible.
- If you're using a subscription-based monetization strategy, release episodic content to keep subscribers happy.
- Make improvements to the game based on analytics and user feedback.
- Add new content, such as levels and characters.
- Fix bugs.

Choosing Input methods and features

Games designed to run in a web browser or as a standalone application on your computer rely primarily on mouse and keyboard input to control the game. Other devices, such as phones, tablets, or your television rely on a range of other inputs and may include features not typically found in computer games. One example is the use of touch-screen input found on most phones and tablets. Some devices feature a dedicated Back button and volume controls, but some do not. While there are certainly commonalities between the input styles on many popular devices, you should consider the best input solution for each device and customize your game as needed.

The Flash Platform currently supports multi-touch and gestures in Flash Player 10.1 or higher in the browser, SWF content published for the iPhone or iPod touch, and in AIR 2 or higher.

Many devices have features that can be used for gaming. Here are a few examples and some considerations for each.

- *Accelerometer:* Many smartphones, tablets, and game consoles sense their angle of rotation by using an accelerometer. You can use this for aligning and moving objects on the screen.
- *Application multitasking:* How will your game behave when minimized and maximized? If it's a mobile game, does it automatically pause when the phone rings?
- *Casual connectivity:* Users are not always connected to the Internet. Does your game require constant connectivity? Will it have an offline mode and an online mode?
- *GPS/Geolocation:* Devices with geolocation, including most smartphones and tablets, know where they are in the world. You can use this to change the game experience, unlock content, or introduce game players who are close to each other.
- *Microphone:* Use voice or sounds as an input—most current smartphones and tablets include voice input options as an alternative to typing text. Consider creating a character that interacts with players by having a conversation.
- *Persistence:* Many mobile users play games in short bursts, starting and stopping the game when they have time for fun. If your game saves its progress, users can jump back into the action quickly.
- *Sleep-mode:* Some devices sleep when not in use. How will your game react? If the user becomes distracted and sets down the device, will the game pause or will it continue to consume battery life?
- *Still camera:* You can use a device's built-in camera to let users add their own images to a game.

Conclusion

Flash games are everywhere. Based on their growing popularity, this is a great time to get started developing some games of your own. The following are examples of popular games:

- Drag-and-drop games that let players solve a puzzle or build something by selecting, dragging, and placing objects.
- Character movement and adventure games that require players to guide a character through a course of obstacles to reach a goal.

For examples of easy-to-develop Flash games, visit: www.flashclassroom.com

The user-friendly design, coding, and multiscreen testing features of Adobe Flash Professional CS6 make it the perfect tool for new game designers.

Using Adobe Flash Professional CS6, you can turn some of your most creative ideas into fully functioning, marketable, and fun-to-play games.