Once you’ve caught the game-making bug, then it’s only a matter of time before you’ll want to start designing your own games. There’s nothing more satisfying than realizing your creative ideas and seeing other people enjoy them, and that’s precisely what making games is all about. We don’t want you to feel that you have to finish this book before trying out your own ideas—have a go whenever you feel ready, as you can always come back for more knowledge and ideas when you need them. Nonetheless, there is more to designing a good game than having a cool idea for a character or story, so these design chapters are here to provide some helpful advice for designing your own projects.

What Makes a Good Game?

We all know when we’re playing one; we become completely absorbed by it and the hours fly by in no time at all, but how do you create a game like that? Well, unsurprisingly there’s no formula to guarantee success—otherwise everyone would be doing it! However, there are some general principles that can help you to create better games by thinking more deeply about the way that games work. To become a good game designer, you need to learn to see beyond the surface features of games and consider what makes them fun at a basic level. This is something that takes time and practice, but we’ll try to give you a taste of what we mean. Think about a particularly good section of your favorite game for a moment. Visualize where it is set, what your character is doing, and how it is interacting with the other characters and objects in the game. We’re going to take fighting giant squids in *Zelda: Wind Waker* as an example:

The skies turn black and there is a crack of thunder as a giant squid rises from the surface of the ocean and towers over Link’s tiny boat. A whirlpool forms around the monster’s enormous body and the boat begins to circle helplessly around it in the current. The music reaches fever pitch and Link’s only hope is to destroy all of the squid’s bulging eyes with his boomerang before he is inevitably dragged down to a watery grave!

Now this next part may seem a little strange: imagine that something has gone wrong with your PC or console and all the characters and objects in the game have turned into colored cubes! The music and sound effects have stopped too, but everything else is working just the way it always did. Now try to visualize your scene again:
A giant pink cube appears in front of my brown cube and I begin circling around it, gradually getting closer on each turn. There are eight white cubes attached to the pink cube—all of which must be destroyed before I get too close. To do this I must target white cubes in my line of sight and launch a yellow cube to fly out and destroy them. The yellow cube then returns back to me, and I can target and launch it again.

Now here’s the question: would the cube version of the game still be fun to play? Well, it certainly won’t be as much fun to play, since it has lost most of its original atmosphere and emotional involvement. However, for good games (like Zelda), some of the gameplay that makes it fun to play would still be there. It may look ridiculous—and you definitely wouldn’t buy it like that—but part of the game’s original magic remains.

Not quite convinced? Okay, take a look in the Games/Chapter05 folder and play the example games, evil_squares.gm6, galactic_squares.gm6, and lazarus_squares.gm6. These are the three games you’ve already made but with simple shapes and sound effects, instead of the usual backgrounds, characters, and music. Give each game a chance and you’ll soon see that there is still fun to be had once all the pretty graphics, characters, and stories have been completely stripped away (see Figure 5-1). Once you’re convinced, read on . . .

Figure 5-1. Galactic Squares is not very pretty, but it’s still fun to play.

Game Mechanics
Okay, so good games are still playable even after all the fancy graphics and sound effects have been removed, but what creates this gameplay? Game developers call it the game mechanics: the basic rules and interactions that make a game fun to play. Understanding game mechanics
is probably the most important part of becoming a good game designer. Sure, creating appealing characters and stories is really important too, but they need to be combined with solid game mechanics to create a good game. Think of game mechanics as the engine of a car and the graphics, characters, and storyline as the bodywork and finishing. A rusty old wreck with a Formula One engine may not win the Grand Prix, but it stands more chance than a Formula One car with a rusty old engine!

Of course, the best games combine great game mechanics with superb graphics, believable characters, spectacular music, and compelling storylines. However, these other aspects are not unique to computer games, and there are plenty of books about filmmaking, storytelling, music, and artwork that cover these topics far better than we could. Therefore, these design chapters will focus on the core skill that distinguishes game designers from designers of other forms of entertainment: game mechanics.

**Interactive Challenges**

What’s the difference between a film, a toy, and a game? It might sound like the start of a bad joke, but it’s actually a question that highlights the two main features that make games special as a form of entertainment. The most obvious difference between films and games is that games are interactive—players have some control over the outcome of games, but film audiences do not. Toys (like train sets, for example) are also interactive, as players have control over what they want to happen when they play with them. However, toys don’t provide challenges for the player in the same way that games do. A player can create their own challenges using a toy (like deciding to race trains), but those challenges have to be created by the player and are not part of the toy. A game normally comes with its own set of challenges that the player must overcome in order to win the game.

So you can think of games as being “interactive challenges,” therefore it’s easy to deduce that both interaction and challenge are key elements of game mechanics. For the remainder of this chapter, we’ll look at the various ways these two elements improve the game mechanics of your designs and make them more fun to play.

**Game Genres**

We often group games into different genres, and one way of doing this is to look at the types of interactions and challenges that different kinds of games provide. Games are evolving all the time, so there will never be a final list of genres that everyone agrees on. Nonetheless, we have made our own list of the main genres. As you read each one, try to distinguish the role of the game mechanics from the part that the characters, stories, and graphics play in the experience of that genre.

- **Action games** (e.g., sports, combat, platform, racing) usually involve fast and furious interactions with lots of physical challenges that leave little room for mistakes.
- **Simulator games** (e.g., flight sims, racing sims) usually involve realistic interactions and physical challenges with no room for mistakes at all.
- **Strategy games** (e.g., war games, puzzle games, god games) often involve slow or turn-based interactions with long-term intellectual challenges that involve planning and organization.
• **Adventure games** (e.g., point-and-click) usually let players interact at their own pace, providing short-term puzzle-based challenges and long-term story-led challenges. These challenges are often impossible to fail if you keep trying.

• **Role-playing games** (e.g., online RPGs) usually provide slower interactions with long-term story-led challenges. However, these are often less important to the player than the story and challenges that the player creates for themselves while developing their character.

Of course, most games don’t fall neatly into one genre and may combine several kinds of interactions in one game. Nonetheless, a game designer does need to consider players’ expectations of a particular genre; a role-playing game that requires lightning reflexes or a turn-based shoot-em-up might not go down too well! It’s also worth remembering that new genres are only created when rules and conventions are broken, and the great games of the future are unlikely to follow the same conventions as today.

**Challenges**

We hope that you can see from the game genre descriptions that different players want different kinds of challenges from the games they play. Despite this, there are some general guidelines that can help you to provide better challenges in your games. We’re going to apply these guidelines to the Evil Clutches game from Chapter 2 to see if we can turn it into a better game. All the new versions of the game can be found in the Games/Chapter05 folder on the CD. We’ve provided these as .exe files because we just want you to play them and notice how the changes are affecting the gameplay. You really don’t need to know how they are made, but you can find the corresponding Game Maker project files in the Games/Chapter05/Registered directory on the CD. However, because these versions contain effects that are only available in the registered version of Game Maker, you will need to use the executables to play the game if your copy is still unregistered.

**Difficulty**

Challenges are important in games, because beating challenges makes players feel good about themselves. For this to happen, a challenge must be easy enough for a player to achieve but hard enough to be worth bothering with. Players give up on games that are too easy, because there is no satisfaction from beating a challenge that you could do blindfolded. However, players give up on games that are too hard because it makes them feel bad about themselves for failing, and they don’t feel they are making any progress.

At the moment our Evil Clutches game is far too difficult at the start of the game, but in other ways it’s too easy as well. It is too hard because just one touch of a demon will kill the dragon, and the game can be over before the player has worked out the controls! However, it can become too easy later on if players realize that they can always safely hide just offscreen and swoop in to rescue the hatchlings.

Even once these issues have been fixed, the game will still be too difficult for some players and too easy for others. People have different amounts of experience with computer games, but the best games are the ones that players of all levels can get into. We’re going to make sure that our game appeals to as many people as possible by adding a difficulty menu at the start of
the game, allowing the player to play in easy, medium, or hard mode. So in combination with
the other tweaks, these are the changes that we're going to make to the first version of our
game:

- Display a health bar for the dragon starting with 100 points of health.
- Make the dragon lose only 10 health points for each collision with a demon.
- Prevent the dragon from leaving the screen.
- Add a difficulty menu at the start of the game for easy, medium, or hard mode.

The file evil_new1.exe contains these four changes to the game. Play the new version
and see what you think. We've changed the difficulty of the game by adjusting the chance
of demons and hatchlings appearing in the different modes. There are now extra demons and
fewer hatchlings in the hard mode and fewer demons and extra hatchlings in the easy mode.
When you're setting the difficulty of your games, remember that game developers always find
their own games easier than anyone else because they play them so much. If you make your
own games harder and harder as you get better and better at them, then they will end up too
difficult for other players. Always get someone else to test your game to make sure you've got
the difficulty levels about right, and if you can't complete the game yourself, then don't expect
anyone else to be able to!

Goals

Challenges are created by setting clear goals for players to achieve. If a goal is unclear or for-
gotten, then it no longer creates a challenge and it loses its power. In the last version of the
game, we sneaked in some extra actions that made saving the lives of 50 hatchlings the ulti-
mate goal of the game. However, you won't have felt any more challenged, since you didn't
know about this new goal! In fact, even once you know about it, it's difficult to keep track of
how many hatchlings you've saved, so any interest in the challenge doesn't last very long. It
may sound obvious, but to keep a player challenged you need to make sure that they know
what their goals are, and how they are progressing with them. Our game currently has two
main goals for the player: saving a set number of hatchlings and beating the top score on the
high-score table. We can make sure that these goals challenge the player by clearly displaying
information about the player's progression toward these goals on the screen.

When players know what their goals are and how close they are to completing them, it
also begins to create the what-if effect when the player loses. The closer a player gets to their
goal, the more likely they are to think, "What if I had just been a little bit quicker, or hadn't
made that one, stupid mistake?" Of course, this only happens if players get close to their
goals—if they don't make it past the first obstacle, then they are more likely to think "As if!"
than "What if?" This means that a player who rescues 40 of 50 hatchlings is more likely to have
another go than a player who only rescues 10 of 50 hatchlings. We could make our game so
easy that everyone can rescue 40 hatchlings, but then the game would become too easy to
complete and lose its challenge. Instead, we can be more devious and reduce the chance of
hatchlings appearing as more hatchlings are rescued. So if the chance of the first hatchling
appearing is 1 in 50 (a 50-sided die), then the second might be 1 in 52, the third 1 in 54, and so
forth. That way, by the time the last hatchling appears, the chance of the hatchling appearing
has changed to only 1 in 150 (a 150-sided die!). In practice, this just means that hatchlings are
released more quickly at the start of the game than at the end. This will make it easier for all players to rescue a good number of hatchlings—and trigger the what-if effect—without making the game too easy overall.

So to incorporate all these improvements to our game's goals, we'll make the following changes to the second version of the game:

- Clearly display how many hatchlings have been rescued and how many need to be rescued in total.
- Clearly display the player's score and the top high score to beat from the high-score table.
- Reduce the chance of hatchlings appearing, based on the number that have already been rescued.

Play the file evil_new2.exe containing these three changes. We hope you'll agree that we're already starting to make progress toward a game that is much more fun to play than the original (see Figure 5-2).

Figure 5-2. Finally we can see our goals and how far away we are from achieving them.

Rewards

Rewards are extremely important for maintaining a player's interest in a game's challenges. It can take a lot of time and effort to complete a challenge, so a reward makes players feel much better about it. It also makes it much more likely that they'll want to complete other challenges offered by the game. The high-score table already provides a reward system for scores, but we could do with something extra special for reaching the end of the game. So, once all the hatchlings have been rescued, we will:
• Display a congratulatory message.
• Award the player with a large bonus score.
• Show the player an amusing conclusion to the story of the game.

Although it is most important to reward players for completing the game's goals, it also helps to occasionally give them small rewards for no reason at all. Games often do this in the form of health bonuses and other kinds of pickups, which appear at random intervals. The fact that they appear randomly is significant, as it gives players hope that a pickup may come along at any point. This means they are more likely to stick with the game when they're in a desperate situation where they might otherwise give up—and if a bonus does arrive just in time, then the feeling of relief is enormous. It also adds to the power of the what-if effect, as players can now think, “What if I'd had just one more health bonus—maybe I'll be luckier next time?” We're going to add our own random rewards to Evil Clutches by making the following changes:

• Make the boss demon randomly drop health and shield bonuses.
• Randomly add between 5 and 25 percent to the dragon's health when a health bonus is collected.
• Make the dragon immune to taking damage for 15 seconds when a shield bonus is collected.

All of these new rewards are included in the file evil_new3.exe. The animation at the end of the game is an example of the kind of animated rewards you can quickly create in Game Maker with a little bit of imagination. It's not exactly a beautifully rendered cut-scene (see Figure 5-3), but it should make players smile.

Figure 5-3. If you mess with dragons, then you're bound to get your fingers burned!
Subgoals

Subgoals can enhance your games by providing short-term or optional challenges for your players to take up. Most games include a long-term goal that must to be met in order to complete the game, but these can often seem very distant and hard to achieve at the start of the game. Subgoals give the player something to aim for in the short term, and good games tend to provide a series of both short- and long-term goals to draw the player through the game. Our game is not very long, but there is certainly room for an additional short-term challenge. We’ll challenge the player to shoot demons without taking damage, and reward them by powering up their fireballs as their demon tally increases. To achieve this we will:

- Count and display the number of demons shot in a row and reset the count back to 0 when the dragon takes damage.
- Limit the number of fireballs in the air at once, based on the current demon count. Begin with a limit of 2 and add 1 to this for each 10 demons on the tally.
- Scale the size of the fireball and add smoke effects to make the fireballs look more impressive as the demon tally increases.

Optional subgoals are a good way of providing extra challenges to advanced players, which other players can choose to ignore. These often include collecting particular items to unlock extra options, or hidden levels that less adventurous players are unlikely to find. These are really just a different way of balancing the difficulty of your game, so that players naturally find the right level of challenge for their own abilities. For our game we’re going to turn the collection idea on its head and add a subgoal of trying not to accidentally shoot hatchlings! To make this work, we’ll include the following changes:

- Each time a hatchling is accidentally shot, subtract one from the total number of hatchlings that have to be rescued (already displayed).
- At the end of the game, award the player bonus points based on the total number of hatchlings they’ve saved.

Ideally this should have its own special reward at the end of the game, but to keep it simple we’ve just rewarded the player handsomely in points for each hatchling saved. You can play a version of the game with these new subgoals in the file evil_new4.exe.

Interactivity

As well as challenges, the other main feature of games is their interactivity. Interactivity is about putting the player in control. Good games leave the player feeling in control of the game, while bad games make them feel powerless. As with challenges, players of different game genres often prefer different levels of control, but there are some common ways of helping to maintain a feeling of control in your games.
Choices and Control

To give players a feeling of control, we need to provide them with choices that seem to have a real effect on the outcome of the game. Action games constantly require players to make choices about the physical actions of the game (jumping, shooting, flying, etc.) and so provide an immediate feeling of control. However, games of all genres should ensure that enough choices are available to create this feeling too. Adventure games without enough choices can seem very linear—as if you are being forced through a path that has already been decided for you. Whenever you add choices to your games, think carefully about the difference they really make: is it worth having ten different weapons that all work in the same way? What's the point in allowing the player to choose what to say to a character if it always has the same outcome? Adding these kinds of features won't generally make your game any worse, but changing them so that they make a real difference will give more control to your players and make them more involved in the game.

We're going to add a choice of characters to our game, so that the player can choose to play the mother dragon or the father dragon. To make sure that this is a choice worth making, we're going to give each character a different special ability, which affects the way they play. The mother dragon will be able to call hatchlings to her—causing them to speed up and get out of harm's way more quickly—and the father dragon will be able to blast demons at close quarters with a cloud of steam—sending them crashing back into other demons. To add the new character and include different separate abilities, we will:

- Add the ability to select between playing the mother and father dragons at the start of the game.
- Add the ability for dragons to activate their special abilities using the Ctrl key.
- Make hatchlings at the same vertical level as the mother dragon speed up when she calls them to her.
- Make demons close to the father dragon fly back into other demons when he blasts them with a cloud of steam.

Figure 5-4 shows the character selection screen. You can see the effect of all these new changes by playing the file evil_new5.exe.

Control Overload!

Of course, it is possible to have too many choices in a game—particularly if extra choices mean extra controls. Most people can remember between five and nine things at once. If you have more than five controls in your game, some players will have forgotten what the first key does by the time they read what the last key does. In general, it's probably not a good idea to have more than two controls plus the arrow keys to move around. Try to make controls automatically perform different functions depending on the situation: pressing the spacebar might pick up items, open doors, or attack creatures, depending on whether the player was near to an item, a door, or a creature. That way, you can include lots of interesting features without needing extra controls for the player to remember.

Fortunately, this is one area where our Evil Clutches game is okay. Some players may find the special move control a bit too much to cope with at first, but because these moves are optional, we don't need to worry about them too much.
Unfair Punishment

With the right level of control in your games, players will feel that they are the makers of their own fortunes. However, you can still quickly convince them otherwise by punishing them for something that isn’t under their control. Such punishments are usually not included intentionally, but friendly characters with suicidal habits and enemies that blow up your objectives are both examples that have accidentally made it into commercial games. Avoiding unfair punishment is usually about making sure your game still works correctly, even when a player isn’t playing the game in exactly the way you intended. The best way to find these problems is to get your friends to test your game thoroughly. They’ll soon tell you if they think that something is unfair about your game.

Our game occasionally punishes the player unfairly. Fireballs go straight through demons, so it’s easy to accidentally kill a hatchling that is flying behind one. The hatchling may not have even appeared until after the player pressed fire, but it is too late for them to do anything about it. When this happens the player may feel frustrated at being punished for something that was out of their control. We’ll solve this problem by making the fireballs disappear when they kill a demon. This also has the effect of making the game a bit more fast and frantic, which is not a bad thing for a shoot-em-up.
Audio Feedback

In the final version of Evil Clutches, we’re also going to add more sound effects to improve the game mechanics. At first glance, this may seem to go against the idea that mechanics are about rules and mechanisms—not niceties like sound. However, sounds are not just included in games just because they “sound nice,” but also because they provide useful feedback to the player about what they are doing. If you go back to the Galactic Squares example from the start of the chapter, you’ll notice that it still includes very basic sound effects. These sounds are designed to quickly inform the player about whether their interactions with the game are good or bad. Audio designed in this way can play an important part in helping to naturally steer the player in the right direction, whereas otherwise they might end up confused. Confused players do not feel in control, so audio has a role to play in this too.

Designing sound effects that both inspire the senses and inform the player in this way is not easy, and commercial games have their sound effects designed by professional sound engineers. See Chapter 15 for more information on the kinds of tools that you can use to try to do this for yourself. You’ll find our sound effects in the Resources/Chapter05 folder and can hear them in action by playing evil_new6.exe from the Games/Chapter05 folder. This final version of the game includes the following changes to the unfair punishment and audio:

- Make fireballs disappear when they collide with a demon.
- Add audio feedback for shooting fireballs.
- Add audio feedback for when the dragon takes damage.
- Add audio feedback when a hatchling is saved.
- Add audio feedback for pickups.
- Add audio feedback for menus.
Summary

Now that you’ve played the final version of the game, we hope you found it much more fun than the original from the end of Chapter 2. In this chapter we’ve learned that challenges and interactivity are a central part of the game mechanics that make games fun to play. We’ve looked at a number of general principles that can help you to create better interactive challenges and followed them through with the Evil Clutches example. These are certainly not the only principles of good game design, and you are unlikely to design a good game simply by following a set of rules. Nonetheless, here is a summary of the main issues as a starting point for your own Game Maker projects:

• Challenge the player by
  • Providing clear, achievable goals and giving feedback on the player’s progress.
  • Including both long and short-term goals.
  • Adding difficulty levels and optional subgoals for players of different abilities.
• Reward the player
  • For achieving goals and subgoals.
  • Randomly.
• Make the player feel in control by
  • Giving them choices that seem to make a real difference to the game.
  • Not confusing them with too many controls.
  • Not punishing them for things out of their control.
  • Giving the player audio feedback about their interactions with the game.

If you apply these principles with a bit of thought and care, then you should find that they can help you to make your own games more fun to play too. That concludes this chapter and the second part of the book—we’ll look at some more design principles at the end of the next part, but for now we’re joining the creatures of a Japanese coral reef to learn something about parenting...