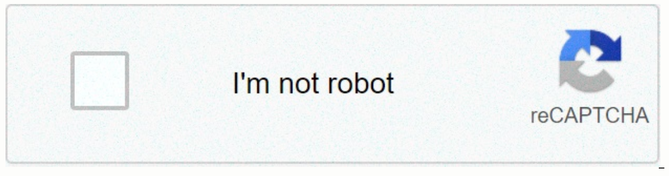

Continue

Build and fight robot games

Robots are pretty exciting machines to watch if you have any interest in science, engineering or artificial intelligence. Many parents buy robots, or robot building kits, if they think their children will be interested in programming or engineering. But as Jenn Choi reports for Quartz, those parents are often disappointed when they realize that most robots are nothing more than remote-controlled droids. This can be good for young children, who are still perfecting the motor skills they need to build a robot and may not yet be ready to learn how to program. But there are plenty of kits that teach some programming for older kids, as long as you know what you're looking for. Choi has tested numerous robots with the help of families with children between the ages of five and eleven and has identified many robots that will demonstrate the benefits of coding by immediately running the programs that children create. But Choi writes that not all robots are right for everyone, something you'd do well to keep in mind, no matter what kind of robot you're looking for or the age of your kids. Robots are available at a variety of prices and for a wide range of ages and capacities for the particular priorities of each family. Among the highest rated robots, which have earned high marks from Choi and other reviewers, there are many models that you can buy and build at home. Check out our favorites in advance. 1. KIBO KIBO is a robot that scans barcodes from color-coded wooden blocks, which connect together with pegs. Each block has an image illustrating a directive and, together, the directives make a programming sequence for the robot to be scanned. Choi notes that unlike most robots, KIBO is simple so as not to intimidate children. This means that this kit is great for teaching kids the basic logic behind programming and mobile robots, even if they don't have any previous experience with coding. 2. LEGO Mindstorms EV3 Source: Shop.lego.com Choi characterizes lego Mindstorms EV3 as the robot par excellence and a good match for what most people think when viewing a robot. Its modular design allows children to change the structure of the robot and its programmable brain allows it to complete tasks without being directed at every step. A typical scheduled task will use an if/then clause. Choi notes that while this robot might seem like a natural next step after your child has built some complex LEGO sets, a child must be interested in the potential to build and code for this toy to succeed. The help of the you need to download software or programs to your computer and understand complex directions. And children facing building challenges will benefit from the help of someone who can ask them the right questions to help them find a solution. 3. VEX IQ Robotics Robotics Kit source: Mastermindtoys.com A similar system is the VEX IQ Robotics Construction Kit, which has been on the education market for years. Consumer-facing versions of VEX IQ and LEGO EV3 are different from training sets, which contain parts trays and rechargeable batteries. Anyway, vex IQ is a great robotics platform for children as young as 8, who can create colorful, personalized and programmable robots without tools. Your children can follow step-by-step instructions to build their first robot and then create their own projects and programs. 4. Remote Control Machines DLX Source: Amazon.com Choi writes that DLX remote control machines can be a good match if you're looking for a robot that doesn't really involve programming or require parents. The set comes with great color instructions that allow your kids to figure out how to build the robot themselves, and kids can choose from 20 different builds. Not all children are as interesting at coding as they are in construction, and that's fine. Choi explains: Although programming isn't on offer here, it's good for kids to learn about the mechanisms they would actually need to program in real life. 5. Makeblock Starter Robot Kit Source: Amazon.com Another blogger with tips for the best robots to build for your kids is Gillian Pemberton, who writes for Fractus Learning that the Starter Robot Kit Makeblock is a great choice because it uses basic Arduino programming. The kit builds a robot tank or three-wheeled robot car and is a great kit for beginners due to its durability and ease of programming. 6. Smart Machines Remote Control Toy Building Kit Source: Amazon.com An easier starting point when it comes to robot kits is the toy building kit for the Smart Machines remote control. The included instructions are for compilation that should take between 40 and 140 minutes to complete and the parts are reusable, so your child can also design and build their own remote-controlled robots. This is a great kit for kids who already like to build and is also a good introduction to remote controls for younger children, who may not yet be ready for a coding-oriented kit. More from Gear & Style Cheat Sheet: This is an edited excerpt from Chapter 5 of Playful Design: Creating Game Experiences in Everyday Interfaces, published by Rosenfeld Media. Okay, you say, I'm ready. You're a regular designer with a long history of creating compelling user experiences. See the benefits that games can offer to the design of the user experience. You like (maybe even love) the idea of building an iPhone game to serve as a tour of a historic park, a game of media for Facebook to organize an independent Flash political campaign or game to teach basic physics on an educational website. You also have an approximate concept of what that game might look like. you you be able to get funding for the project within your company or present the idea to a receptive customer. You want to start putting your ideas on paper, start development and accelerate towards launch. It's a great place to stay. Vision and ambition are written in the opening lines of all success stories. But there are some risks of jumping too fast. The development of the game takes a lot of time and requires resources. It can be difficult to make a significant change in direction if you find out midway through the development that some of your initial ideas aren't translating into the great gaming experience you've had in mind. So, before you start and run with it, you need a primer to steer you in the right direction, to get away from the most common mistakes and maximize your chances of success. This is the chapter for you. While the challenges of building a fun game shouldn't be overly simplified, the 10 general guidelines I present here will at least help you refine your ideas and overcome some of the common barriers that might otherwise hold back your design.1. Matches must be first of allThis point may seem too obvious, but it can be very easy to miss. And it often lacks the cancellation of a well-intentioned design. You can design games to teach and persuade (as discussed below), but if such real-world goals oversee meaningful gameplay, they will undermine your chances of success. First of all, a game must be enjoyed. The schwab game MoneyWise It's Your Life has a noble mission: to convince people to save more money for retirement and other long-term goals. Just like the original Game of Life board game, It's Your Life offers players a number of choices between spending or saving money over a simulated lifetime (Figure 5.1). In the end, players get a letter vote to represent how well they did. Figure 5.1 With each step in Schwab's It's Your Life game, it's pretty obvious what choice will lead to a winning result The problem is that the game's designers were much more interested in hammering their message at home than creating a real gaming experience. If you want to win the game, the right choice at every stage of the route is to save your money and not spend any money on it. Never. On anything. This means you can earn an A+, the highest score possible, if:Skip collegeDo you never leave your parents' houseDo you never get marriedYou never have kidsDo you never travel or take holidaysWork indefinitely over 65Die alone with a lot of money and no one to leave it to Me I'm sure the designers reasoned that the people who through the scenarios they would elect to do valuable things with their lives, but they set the game so that doing nothing with your life vigorously saving is a safe way to win. Separate what people should do from what's rewarded expected message. Although It's Your Life is packaged as a game, it is not committed to being experienced as a game. As far as we all know that the test is absolutely, completely indispensable in the design of the user interface, I must emphasize (the grammar must be damned) that it is even more absolutely, completely indispensable when designing a game. Although the mantra of every UX designer is testing, testing, testing, it's still worth saying that you don't really have to neglect to test your game early and often. The reason why testing is so important in game development is that most video games are highly dynamic experiences. The flow of events changes from moment to moment, and every decision made by the player leads to a multiplicity of results. Most games are also programmed with an element of randomness, so the same player never has the same experience twice. Multiplayer games throw even more unpredictability into the mix. As a result, the designer does not directly control the actual gameplay, but only controls the underlying system in which the game takes place. Without actually seeing the game in action, you can't reliably anticipate how it will work. Mike Ambinder, an experimental psychologist at game developer Valve Software, says this in scientific terms: Every game project is a guess, and every game instance is an experiment. So, you're not lucky to have a background in testing user experiences! Take advantage of it at every opportunity when designing your game. Take your colleagues, your family, your friends - anyone willing - sit down with your game and watch them play it. Don't forget to play it yourself! Be harshly critical. Do you like to play with it? When it's over, do you feel like playing it again? Is it frustrating? Is it boring? Is it too hard to figure out what to do? I'll go into more detail about game-specific testing methods in Chapter 8, but it's important that you're ready to put your game under the microscope over and over and fit the design to make it more enjoyable.3. Games don't have to be for kidsI's young people have a lot more free time than adults, and many of us remember spending long periods of our childhood playing. So it's natural for us to associate games with children. Video games in particular tend to have a youthful image, and this is not without reason: 91% of children under 17 identify as players and often have a lot of influence on which games a family buys. Large segments of games are marketed to kids, and many of these games feature children's mascots like Pikachu or Mario. The link between childhood and video games is very real, so it's no surprise that designers decide to create games specifically intended to appeal to children. But with a large market that caters to them, even children children latitude to be very demanding consumers. Hugely sophisticated cross-media marketing campaigns that push big-budget headlines already flock to each other, so you'll find that getting the attention of a young gaming consumer is a tremendous challenge. Children often select a popular title specifically because they feel it will raise their social status among their friends. Since these games can be very challenging of their time, your idea must offer a pretty compelling value pro proposal for them to sacrifice minutes or hours that might otherwise be spent with their favorite pastime. Kids take games seriously and you can't assume they'll play your game just because it's a game. We also know that children are only the minority of people who play video games. As I said in Chapter 1, 82% of players are over the age of 18 and 29% are 50 years of age or older (Figure 5.2). 3 Adults may also be more receptive to playing outside the mainstream and have more disposable income to spend on games (if you plan to sell it to consumers). That doesn't mean kids couldn't put a part of the audience in your game. But if your game is clearly aimed at young children, as announced in the breathless starbursts that read Hey kids! and Super cool! you'll turn off the larger segment of players. Then consider directing your game to an older age group while keeping it accessible to a wide range of ages. Figure 5. 2 Children under the age of 18 represent the smallest minority of game 4 players. The action can be boringCall of Duty: Modern Warfare 3 is a fantastic action game. It takes place for tens of hours, during which you encounter waves of enemies exquisitely balanced with the resources at your disposal, interact with teammates controlled by artificial intelligence (AI) algorithms, and fight through various locations that provide no unfair advantage to either you or your goals. And it's all wrapped in an engaging and complex plot. Call of Duty also took years to make and a team of dozens of designers, artists and engineers at a cost of many millions of dollars. You're probably not doing Call of Duty. It is very difficult to sustain adrenaline excitement for a long time. If you choose to create an action-based game on a small scale, you'll find that you're limited to very simple, short-lived scenarios that look like arcade-era games. Running a car. Throw a basketball. Shoot a spaceship. Taken alone, these types of experiences tend to get annoying quickly. Compared to the extremely sophisticated action games that people have access to today, they are simply boring. Consider what makes a intrinsically interesting. You'll find plenty of creative opportunities in games that make the player think through interesting choices instead of running twitch responses. Paper Hearts, for example, are choices (Figure 5.3). What three cards should I pass to my opponent? Do I need to play a high card or a low card? Should I break hearts or resist to see if anyone else does it first? If I play club again, will someone else inont me to the queen of spades? Should I try shooting at the moon, or will this prove self-destructive? Each choice is reevaluated from makeup to makeup, depending on the changing condition of your hand and new information about what other players have already done. Although Hearts can be a fairly long game, it can maintain player interest without laser blasters or lava levels. You can also invite players to apply their imagination to the game. Mafia Wars, a Facebook game with over 3.5 million monthly active players,4 simply alludes to street crime while showing none of this (Figure 5.4). To get a bank robbery, just select Bank Robbery from a menu of criminal activities. The game responds immediately with a message that you have successfully completed the work. Instead of real-time action and 3D graphics, players are offered choices about what jobs to take, how to invest their earnings, and what personal attributes to develop. There is no limit to what can be achieved in a player's imagination. Figure 5.3 Hearts creates excitement by presenting players with many interesting choices Figure 5.4 Mafia Wars leaves depraved crime to the imagination of player 5. Adapt the game to the player's lifestyleInvece to real-life contexts where people will play. Start the design process by asking:Who are your players? How much time do your players have to give the game and how much time would they actually be willing to give? Will your players have to take a break from the game and continue later? How will your players access the game? Where will your players be when they play? What kind of hardware, software, and internet access will be available to your players? Answers to these questions can help you set requirements for the duration of the game, how you will access the game, and the technical requirements of players' computers and devices. Use game tests to see if your estimates are going up. For example, Unisys has developed a series of online games for the company's sales team to send to customers as holiday greetings. Customers would receive an email link to an online vacation card with a personal message from the agent. The card would then open in the game, marked with the Unisys logo (Figure 5.5). Figure 5.5 The minigolf game Unisys is designed to be a quick and non-intrusive diversion from the Work As players were adults who received these emails at work, games could not require a significant investment of time to reach the end, so all were designed last less than five minutes. And because many players accessed the game while sitting in standard office cubes, where they typically had computer speakers turned off to avoid irritating colleagues in their shared space, limited sounds in games were not essential to the experience. Compare this design to Metal Gear Solid 4: Guns of the Patriots, a home console game containing movies (game movies during which gameplay is suspended) that can run for up to an hour and a half and can arrive at any time during the game (Figure 5.6). Games like this require real commitment from their players and are only appropriate for the audience with plenty of free time. Figure 5.6 Individual footage in Metal Gear Solid 4 could run for an hour and a half FarmVille makes it skillfully adaptable to the player's lifestyle. Players should only spend a few minutes at a time, during which time they can plant seeds for crops that take different amounts of real time for harvesting. Raspberries only have two hours, so they are useful when the player can check in multiple times in a single day. Eight-hour pumpkins fit well just before and after a working day. Crops like artichokes need four days to harvest, better for players who can check in only from time to time. The player is asked for some commitment, as fully grown crops that are left uncultivated for too long are cultivated and cost the player gold coins. But staggered growth rates allow the time commitment to be on the player's terms (Figure 5.7). Figure 5.7 Staggered crop harvest times at FarmVille allow players to decide how much gameplay can fit their lives 6. Create meaningful experiences Players must apply their time, concentration, and troubleshooting skills to the challenges your game throws at them. These efforts

should make sense, a payoff for their investment. At the end of the game, players should come away feeling that the experience was significant. For the game to be a meaningful experience, players need to have a sense of control over the result. If players win or lose, does this prove anything about their skill, knowledge, or intelligence? Or did everything just come down to a coin toss? Many games involve some elements of randomness, putting parts of the experience out of the player's control. A random element adds interest to the game by questioning the result. But a significant game at least gives players a hand in reversing the odds in their favor. A great example is the card game Killer Bunnies, in success is ultimately determined by a card randomly chosen from a deck (Figure 5.8). The player who holds the game for that card (the magic carrot) is declared the winner. No player has any control over which card is chosen; it's a completely random selection. But gameplay gives players players control over which corresponding cards they hold. Players compete for carrot cards throughout the game, and cunning players will work to hold as many of them before the end of the game. Even for players who don't win, the game says a lot about their mastery of strategy, risk tolerance, and ability to read other people. Players move away from the game knowing they have control over their chances of success, which makes the experience meaningful. Figure 5.8 Players exercise control over the outcome of Killer Bunnies by acquiring carrot cards, increasing the likelihood that they will capture the randomly selected magic carrot 7. Don't cheatWhy video game rules are applied inside the black box of computer circuits, there is a particular temptation for designers to take shortcuts by letting the game cheat. Giving the system more information or control than the player,for example, can be an easy way to build a challenge in a game. Power in a video game is unbalanced between the computer and the player, and the player has no way of challenging the computer or taking it into account. Don't be tempted to cheat. It's a bad design choice because, first, people will be able to say what's going on (oh yes, they will); secondly, cheating is a serious crime in games, and players have an instinctive revulsion at it. Let's say you're designing a blackjack game that matches a player against a computer decker. As a designer, you need to write a script to control the retailer's actions. You want the drug dealer to be a little hard to beat but not impossible. An easy way to create a challenge would be to let the script choose which deck card is drawn later. Then program the decker to choose a card that will win or lose and you will put in a randomization function so that twice out of three it raises a trump card. This strategy also creates an easy way for players to change the difficulty, so that on a more difficult setting the decker will choose a trump card four times every five times, while in an easier environment only one in three wins. Since the deck of cards is displayed face down on the screen, how do you know you're cheating? After playing a couple of times, you'll see how (Figure 5.9). The decker will do seemingly irrational things, like hitting on 20 and magically drawing an ace. The deck won't look random, because some cards will tend to show up soon and others will only be shown after those favorite cards have been drawn. After several playthroughs, these schemes will become painfully Although the player cannot capture the computer in the act of cheating, these revealing artifacts are difficult to cover. When players realize that a game is cheating, they will make the ultimate winning move by turning it off. Figure 5.9 If your computer wins Look this way, players will come to recognize a cheating model even if they have no way of demonstrating it A better approach is to build a simple, rules-based AI. Don't be too intimidated by the idea of building an IA; in the end it's just a computer program like everyone else. In this case, all you need is a line of code that tell the retailer to hit the 16th and stay on the 17th. The important thing is that the computer is subject to the same rules as the player. Make things work as they seem to work. If you show that a deck is being mixed, randomly choose the entire sequence of cards and put it in an array that cannot be changed. Don't let the IA know which card is coming or which cards are in the player's hands. Don't abuse the inherent advantage you have as a game designer.8. Skip the manualThe best way to convince people it's worth playing is to let them jump and try it yourself. You can make people's decision to open a game as the clearest possible signal that they are in the mood to play, not sit back and read about how to play. Relying on written instructions presented at the beginning of each new game only creates a barrier to entry at the very moment you want to be more accommodating among players. The instructions can also become a crutch, used to justify unconventional and non-intuitive choices in the interface. Finally, the game instructions can be very difficult to follow. Each game interface introduces a new vocabulary and a new set of controls. These things can be hard to imagine abstractly outside of gameplay dynamics. So the best place to teach people how to play is right there, within the game itself. Tutorials have become one of the most familiar models in games. Minimalist and just-in-time instructions are even better (Figure 5.10). Ask yourself: What is the minimum amount of information the player needs to make the first move? Then provide nothing but this; you can get to the second move when the time comes. To play is to learn. If people are interested in the game, they will be motivated to fill in the blanks themselves by playing it. Figure 5.10 In the Bri lance Kanyu game, step-by-step instructions on how to play are cleverly embedded directly into the game's history And keep in mind that if your game needs robust instructions for people to play it, this could be a warning sign and in itself. Your game may be too complex and some simplification may be in order.9. Make sure the game makes sense Players need to understand why things happen in the game to feel that they have the Your skills as a UX designer will be very valuable here, because this point is basically about the intuitiveness of the gameplay. In the design of the game, building a sensible experience is based on some key information between the designer and the player. When players lose, it should be because they lost. If not, players will not be able to improve the game by avoiding the same error in the future. If this happens repeatedly, players will start to feel that they are being punished unfairly. When players win, it should be clear why they won. If not, it will be difficult to replicate the victory. A win that doesn't make sense can also be cheap of the experience, leaving players feeling that the standards of the game weren't that strict in the first place. Every effect should have a clear cause. When something happens, players should be able to understand why it happened. Foldit, discussed in Chapter 1, is a wonderful example of game mechanics applied to a real-world problem. The relationship between cause and effect, however, is often unclear in the game. Twisting a protein's side chain can create a conflict, but twisting a similar one in a similar way can earn points. Trying to understand why these actions have different results can be tremendously frustrating. The object of the game should be clear. Players need to know what they're working on. A clear goal gives structure and meaning to the experience. It allows players to formulate strategies and gives them a reason to interact with the game. From the beginning and at every moment of the game, players should be aware of their ultimate goal. Players should always know what actions are available. At all times, visible or ural signals should be provided to let players know what they can do. Adventure games, a popular genre in the 1980s, were plagued by failures of basic intuitiveness, because they often forced players to guess what arcane actions might be available. Using a blue key to open a blue door makes sense to most people; use your athletic supporter as a slingshot to knock out a guard (as required in Space Quest II) actually not.10. Make it easier to try againWhen you're down in the pits building the mechanics of your game, it's easy to focus on the ideal case where players play directly from start to finish. It makes sense to create a game as a continuous narrative, with a beginning, a medium and an end. But thinking about your game in these terms also risks losing sight of how it will actually be experienced in the real world. Remember to take a step back and think of the game as a discontinuous and iterative experience. When a player loses, it should be easy to get back in the game and try again, instantly and effortlessly. Even large commercial games with multimillion-dollar development budgets make the common mistake of forcing a long loading screen into that anxious space between a loss and a second Extending the time space to the second, third, or twentieth rounds inevitably attempts the player's patience. Games like Braid and Prince of Persia: The Sands of Time have taken a smart road this issue, allowing players to rewind time to a safe point before the lost moment. Also think about the amount of work the game asks players to invest in it and whether players would be frustrated if they lost and had to start all over again. This alone might be enough for some players to decide that it's not worth returning to the game. Consider giving players a chance to save their progress. Think about giving players incentives to play again after they've completed it. Some common ways to do this include:Simple performance parameters, such as ratings on a carnival strength testCollectables and results obtained during the game, and a count of how many players managed to get tracking Online scores and leaderboards Interactive productions of new contentNew features and privileges that become available only on later playthroughsWhen people play a game , will signal a personal appreciation for its design. Tracking the number of times people replay is one of the best general measures of your game's success. Play with your strengths10 guidelines will help you get started, but there are plenty of challenges ahead of us as you start designing and developing your game and you'll need to learn how to manage them as they arrive. One last tip is to play with your strengths. If you have a background in the design of conventional user interfaces, use the skills and techniques that come with it by all means. Wireframing, user testing, rapid prototyping, storyboarding, flowcharts, and other major skills all translate well into game design and can help you overcome the inevitable rough patches. When a game design problem confused you, trust your instincts and ask how I would handle such a problem if I wasn't designing a game. Most of the time, you'll find that you can point in the right direction. This is an edited excerpt from Chapter 5 of Playful Design: Creating Game Experiences in Everyday Interfaces, published by Rosenfeld Media. Please use the discount code NETMAG12, voucher for a 20% discount on all products purchased by Rosenfeld Media. Average.