The Philosophy of Computer Games Conference, Copenhagen 2018

From Rogue to lootboxes: two faces of randomness in computer games.

Paweł Grabarczyk

Introduction

In this paper I try to look at two opposite ways that randomness is employed and, more importantly, evaluated in computer games. It seems fascinating that the same mathematical and statistical phenomenon could lay at the roots of some of the most acclaimed and despised design principles of contemporary games. On the face of it, this may not seem to be paradoxical at all as it could be argued to be true of many other objects and processes. After all, the same knife could be used to prepare a meal and to hurt people. What makes the case of randomness interesting however is that this connection between its heralded and criticized sides is never examined and that they are often treated as completely separate phenomena.

One initial clarification that may be needed is that the paper does not cover instances in which randomness has been used only during production of the game. In other words – I am interested only in cases where random values and patterns appear during the game execution. To see this difference, think of any game which contains a map generated in random or partially random manner via some additional development tools, but where the game itself uses the map as a fixed asset. To use a concrete example, compare Just Cause 2 and Minecraft. The former contains a fixed map which has been (at least partially) procedurally generated by the developers (Blomberg 2013), the latter contains a map which is generated by the game itself, whenever it is started anew.

Conceptual clarifications

In order to discuss the two titular "faces" of randomness, it is first important to look at the phenomenon from a very general perspective and describe it in a way which does not preclude anything about its particular implementation. The first distinction to be made is that randomness can be understood from the perspective of epistemology and ontology. The former perspective connects randomness with the notion of "unpredictability" (Futuyma 2005). A string of characters or a chain of events can be said to be random if we are unable to discover a pattern enabling us to anticipate the next character or event. From this perspective, randomness can be thus defined as the "lack of predictable patterns" (Eagle 2005, Frigg 2004). As is typically the case, ontological perspective provides a stronger claim, according to which the unpredictable nature of random patterns is not only a result of our insufficient knowledge or lack of processing power, but rather an objective feature of the string (or chain of events). According to this perspective, random string can be characterized as a string which is unpredictable in principle, regardless of our knowledge or processing power. One interesting example of a definition of randomness which results in ontological consequences and which is, at the same time, especially relevant in the context of video games, because it refers directly to the notion of computation, is Kolmogorov's definition which states, that a given string is random if it is longer then any procedure which can generate it (Li & Vitányi 1997). The main idea behind this definition is that true randomness cannot be compressed in any way, because compression has to utilize patterns which the random sequence simply lacks.

One simple reason why the difference between epistemological and ontological randomness is important is that it is perfectly possible for a given pattern to be epistemologically random without being ontologically random and vice versa. A string or a pattern which is not predictable to us can contain patterns, which we are simply unable to detect. On the other hand, a random string or pattern can be perceived by us as non-random, either due to some cognitive error or due to the fact that we perceive it only in a simplified, idealized way. In this case we focus on patterns existing only in our imperfect perceptual model of the analyzed string or pattern. As we are going to see below, the difference between epistemological and ontological understanding of randomness plays an important role in the comparison between both faces of randomness in games.

Another important general question, which we should answer, before we proceed further is the question as to whether randomness can actually appear in computer games. The reason why this question needs to be raised is directly related to Kolmogorov's definition cited above. Computer games are software objects built from algorithms and data sets. If randomness cannot be generated via procedure, then how could we employ it in an algorithmic way? It can be argued that the lack of real randomness is something that differentiates between computer and traditional games, because the former can easily contain physical random generators which do not contain algorithmic methods. Die rolls or cards shuffling used in the majority of traditional games can be seen as model examples of such physical random generators. These simple mechanisms are so ubiquitous that It is easy to forget how crucial they are for traditional games to work. An example of the simplest possible random generator of this sort can be found in the ancient Egyptian board game Senet, which contains small flat sticks which could be thrown and generate two outcomes, depending on which side they fall (analogous to the flip of a coin). Interestingly, even games which do not contain randomness employed as a central mechanic, utilize random generators. Starting a football match with a flip of a coin can be a good example of that.

Randomness in computer games

How can computer games replicate this effect? To use a simplification (which is sufficient for our purposes), we could discern between two types of randomness: pseudo-randomness and true randomness. Pseudo-random value can be generated in a variety of ways. One popular way of achieving this is to utilize some of the contingent properties of the system running the software. The idea is that even if most of the properties of the system running a given piece of software are stable across different runs – otherwise the software would have been completely unpredictable – the computer contains also values which are irrelevant for the software and which change in the case of every run. Date, time of day and machine running time are typical examples of this. Pseudo-random values are typically computed using these contingent values as input data. There is a number of reasons why these contingent numbers are manipulated by an algorithm and not simply passed to the user. First of all, their format is often incompatible with the needs of the user. For example – if I wish for a computer program to simulate a coin flip, the number representing the

running time of a computer or a date is not helpful, as everything I need is a binary outcome of 1 or 0. Second of all, if the number was not manipulated with, it would have to be easily predictable for all users. Note that the notion of pseudo-randomness remains to be heavily connected with the notion of epistemological randomness – a pseudo-random value is unpredictable to the user, but is, in principle, predictable for anyone who knows both: the algorithm used to manipulate the value and the value. Resorting to pseudo-randomness would have been rather unacceptable for any serious task, but is very often sufficient for computer games, because the only purpose it serves is securing the weaker (epistemological) sense of unpredictability.
Contrary to this, true randomness relates to random sequences or values obtained from a source which is considered to be unpredictable in the stronger, ontological sense. Examples of sources the developers can use are weather data and… popular lava lamps which could be simply photographed or filmed. It is easy to see, that this method escapes the problem we started this section with as the random values are not really generated but rather obtained or utilized in further computation and the fact that they are later handled by algorithms does not negate their random origin. Older computer games typically resorted to pseudo-randomness for purely practical reasons – machines were not able to access random data, because the connection to services providing such data was not possible. At the same time, storing the data locally proved to be problematic because, due to the nature of randomness, compressing it was not possible and storage space was often expensive. Repositories of genuine random patterns are currently widely available for the developers , so the decision as to whether a given computer game should use pseudo or genuine randomness is no longer a technical one.

Note that the difference between pseudo and genuine randomness provides another case in which the difference between ontological and epistemological understanding of randomness comes in handy. The difference between real and pseudo-randomness is completely indistinguishable from the epistemological perspective of the user and the question as to which form of randomness has been used can typically be only obtained only from the developer. As we are going to see in the section below, this fact may sometimes be important to the players.

Whether randomness is considered to be a welcomed addition to a game depends largely on the genre the game belongs to. Using a simplification that is sufficient for our current purposes, we can distinguish between three types of games. In the case of games of skill containing a very rigid structure, any random occurrence during the run of the game may be considered to be an overlook in game design or a glitch. Think of Mega Man series or any other platformer requiring memorization and skill. The way these games are played precludes random elements as the memorization and so called “muscle memory” requires the developers to avoid unpredictable elements. We could say that these games function as meticulously crafted machines the player skillfully operates. Contrary to this, in the case of games of chance, randomness remains to be a key element of design. It can be argued, that in this case the usage of genuine randomness may be very important to the players – think of on-line casinos as of a good example of this. The users of these services may feel safer if the company resorts to genuine randomness as it may give them a perception of a more “fair” game. The third, most varied category contains games in which the addition of randomness is possible, but nor required. This category is interesting the most because it provokes the question as to what exactly do developers achieve by adding random elements. The answer to this question is complex and belongs to a bigger study on randomness and game design, but it may be useful to point at two reasons why randomness may be desirable.

The first reason comes from the fact that randomness enables subsequent runs of the game to feel different for the players. To use a popular term, it facilitates “replayability”. This function of randomness has been known for years, since it was used in classic games, such as Rogue. It is nonetheless important to mention this advantage of randomness because it became especially relevant in current times due to the resurgence of the rogue-like genre (Garda 2013). The second reason why introducing randomness to games is desirable comes from human psychology and boils down to the simple fact that a skillful introduction of randomness gives the players pleasure. A very simplified explanation of this is that, whenever we encounter a rewarding experience which seems to be completely random, our brains try to find the mechanism responsible for this lucky event. In order to discover this mechanism, the brain needs to collect a relevant sample of similar events, because this is how a statistical pattern may be discovered. The only way of collecting this statistical sample of events is for the brain owner, that is us, to repeat the action which preceded the reward. This makes the brain to reward us internally for repeating the action that led to the reward. Needless to say, the pattern will never be discovered (as there is simply no pattern in the circumstances surrounding the reward), but the action itself starts to be pleasurable in itself. As we are going to see below, both of these aspects of randomness contribute differently to game design techniques, resulting in the titular “two faces of randomness”.

The dark side of randomness

One of the biggest controversies surrounding computer games which relates to randomness concerns so called “loot boxes”. As can be seen in (Dingman 2017, Hood 2017) the problem gathered interest of specialized press and even mainstream media (BBC 2017, Brown 2017, Hern 2017). What is even more important, in the aftermath of the controversy several legislating bodies in the EU countries started to debate the possibility of regulating games containing loot boxes or maybe even banning them altogether (Cross 2017). In order to discuss the problem in a jargon-free manner it is probably best not to use the term “loot box” as it is not precise enough and may lead to confusion. For example, several analogous mechanisms existing in games do not contain any representation of boxes, so the phenomenon is obviously much more general than a particular implementation. For this reason, as proposed in (Nielsen & Grabarczyk 2018) we suggested a notion of Random Reward Mechanism (henceforth RRM) which can be defined as any randomness-based intermediary between player’s action and the reward she gets. The main difference between contemporary RRMs and implementations popular in older games is that the new games containing these mechanisms seem to accentuate and even celebrate randomness. To show it on a particular example it is best to compare the loot system present in the original Diablo and the one present in Overwatch. It is possible for the user of the first game to be unaware of the random nature of the contents of a chest she is opening as the opening itself does not represent the random procedure itself. Contrary to this, Overwatch prolongs the moment of opening the box which suggests to the user that what she is witnessing is the random procedure in action. The hypothesis that the sheer witnessing of the random procedure started to be enjoyable to the users seems to be confirmed by the existence of streams devoted to loot boxes opening. There are several reasons why randomness used in RRMs spawned controversy. Let me focus on three which I believe to be the most prominent.

First and foremost, many users find combining RRMs with real money purchases to be questionable. As presented in (Nielsen & Grabarczyk 2018) RRMs can be divided into four different categories, depending on how they relate to real world currency purchases.

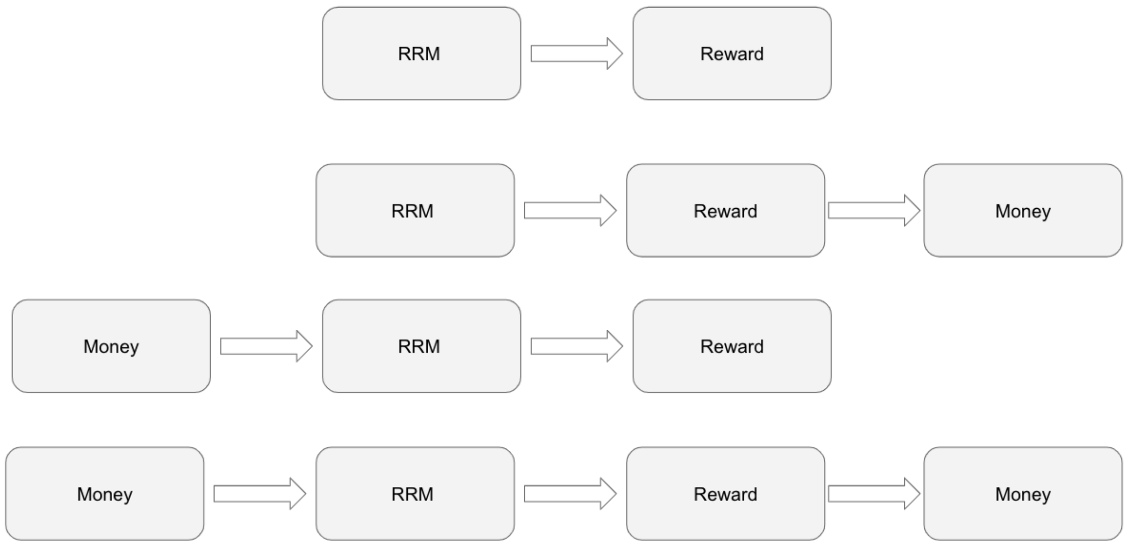

Figure 1. Four types of Random Reward Mechanisms implementation[1]

The first category describes games which contain RRMs but which are isolated from real world economy – the players can neither: buy the random containers, nor sell the objects obtained from these containers. Seeing how ubiquitous this form of RRMs have been in computer games, we could say that this form of harnessing randomness does not lead to much controversy. At this point it may always be put under scrutiny and reexamined, but there is no doubt that earlier games containing RRMs did not inspire the same discussion, that the newer games have.[2] The second category adds the possibility of selling items obtained from randomized drops. The difference between the first and the second category can be best illustrated on the difference between Diablo 3 with and without the auction house. The third category can be understood as the opposite of the second one. In this case, the player can trigger the RRM with a real money purchase but is not able to sell the item obtained via the random procedure.[3]It seems, that judging by the number of contemporary games which implement this model, it is at this point, the most popular way of incorporating RRMs in games (popular games such as Fortnite, Overwatch and Heartstone belong here). The last, fourth category of implementation of RRMs gives the player both: the possibility of triggering RRM with money and the possibility of selling the item won. Interestingly, this category of games is often overlooked in popular discussions as the possibility of selling items is

[1] Different visualization of this categorization has been presented in (Nielsen & Grabarczyk 2018).
[2] What is interesting, some of these discussions have been foreshadowed by discussions found in (Fisher and Griffiths, 1995). It can be nonetheless argued that many of these papers focused on superficial similarity between arcade and gambling machines.
[3] At least legally as the possibility of selling a whole account always exists.

typically tied to an external overlay, such as Steam Market. Examples of games belonging to the fourth category are Playerunknown's Battlegrounds and Counterstrike GO.
As argued in (Nielsen & Grabarczyk 2018) two last categories inspired the resurgence of discussions comparing games to gambling practices (see Griffiths & King 2015 for an example of this). The biggest question that needs to be answered is: to what extent games which mix RRMs and money trigger typical cognitive fallacies associated with gambling, such as gambler's fallacy. There are no doubts that this question demands further empirical studies. At this point it is only possible to indicate that functional similarity of gambling and games which mix RRMs with money creates a suspicion that they are designed specifically to exploit cognitive errors in order to maximize financial gains.
Whether such a practice should be considered unethical is a more complex question. It could be argued that the exploitation of cognitive errors and specific aspects of human psychology is a fairly standard artistic practice and that creators often uses "smoke and mirrors" to create desired effects in the audience. Should we consider a classic example of a "jump scare" (think of the famous scene introducing zombie dogs in Resident Evil) to be unethical, just because it exploits our cognitive error - the fact that our knowledge is not capable of penetrating our more primitive reaction of fear? It seems, that ethics enters the picture only because of the involvement of real money transactions and not because of how the game plays with our reaction to randomness.

The second reason why randomness started to be connected to problematic design principles can be best illustrated on the case of controversy associated with Star Wars Battlefront II. As can be seen in the model above, EA's game belongs to the third category which is, arguably, less similar to gambling than the fourth one. And yet, it is this specific game that created the most controversy. As can be seen from popular debates[4], many players do not accept RRMs associated with objects which affect game mechanics (as opposed to game aesthetics). This argumentation pertains to a broader category of additional purchasable content, regardless of whether the content is locked behind RRMs, but the problem becomes much more pressing when purchases of this type become linked with RRMs. The reason for it is that once the additional, purchasable objects become functionally meaningful, the players may feel that the game requires them to buy them. But if the purchases are locked behind RRM wall, the players are now forced to partake in an activity which they may find ethically dubious or psychologically dangerous (because of its gambling-like structure). In addition to that, RRMs make the evaluation of the cost of items practically impossible, because the epistemological uncertainty of randomness makes it economically impenetrable.
The popular misconception related to RRMs is that revealing the odds makes them transparent. And yet, it is important to remember that the only thing this move secures is that it brings digital randomness on par with its non-digital equivalent. As I pointed earlier, one of the characteristic aspects of digital randomness is that the methods used to generate it are hidden from the player (which is especially important in the case of pseudo-randomness). Contrary to this, traditional random generators, such as dice, keep their odds fully visible. The reason why revealing the odds in the case of digital randomness won't solve the problems associated with it is that even if the user knows the odds, she is not able to predict the outcome, just as knowing that the odds of getting heads in the coin flip does not make us any wiser as to what exactly will we get once we throw the coin.

[4] The famous Reddit most downvoted comment related to Star Wars Battlefront II (https://www.reddit.com/r/StarWarsBattlefront/) can be a good example of this.

The light side of randomness in games.

In a stark contrast to the perception of RRMs, randomness used in order to enhance procedural generation is typically met with praise. This can be seen in the reaction to the initial trailer of *No Man's Sky* – a game which promised to generate almost every of its aspects procedurally. What is more, even a cursory glance at the most successful and critically praised independent games shows, that most of them contain procedurally generated elements. From *Minecraft*, *Spelunky*, *Rogue Legacy* and *The Binding of Isaac* and Don't Starve. Moreover, procedural generation and randomness started to be introduced to genres which were rarely earlier combined with it, such as first-person shooters (*Tower Of Guns*, *Strafe*) and even adventure games (*Unavowed*).[5]

In order to avoid confusion, it is important to point out that notions of "procedural generation" and "randomness" should not be conflated. It is perfectly possible for a game to contain procedurally generated elements which are not randomized. Being procedurally generated means simply that a given element is generated by a procedure (instead of being premade by a human designer). Many early computer games generated their graphics procedurally in order to save disk space – Sierra adventure games from the 1980s can serve as a good example of this. Still, the connection between random generation and procedural generation is fairly common and is typically achieved through the generation of the initial value set used for further generation (so called "seed"). What is the reason of the recent popularity of procedural generation (and the randomness used to generate its values)? It seems to me that there are two important groups of factors which are responsible for the "procedural explosion" we are witnessing just now.

The first group of reasons concerns the pragmatic side of game production. It can be argued that the popularity of procedural generation came from the synergy of several factors which, taken in isolation, may seem to be fairly independent. First of all, the production methods used in contemporary independent games favor some of the design practices and solutions which are similar to practices used in early times of game development. Independent games are often created by very small teams (sometimes by a single developer). What it means in practice is that the developers have to make compromises when it comes to asset creation. The ability to create some of the assets (be it levels, textures or even sound) procedurally helps to alleviate this problem to a certain extent. This enables the small development teams to deliver games which rival big budget production in their scope (*No Man's Sky* or *Minecraft* can be seen as model examples of this). Additionally, the increase in computing power of contemporary machines eliminated some of the downsides of procedural generation visible in early computer games using these techniques. One obvious example of this change is the increased speed of graphics generation – a problem which plagued most of the early games where the users have to wait for the graphics to be generated and drawn on screen (look at *Elite 2: Frontier* as an example of this). The second reason is that procedural generation proved to be compliant with many of the most popular contemporary genres, such as survival games, rogue-likes and so called "metroidvanias" (to a lesser extent, but see *Chasm* as a good example of this). Third of all, it can be argued that high randomization of game elements remains to be one of the key aspects responsible for the game success on streaming services, such as Twich, because it gives the streamer the possibility of playing the game as long as his audience desires. Fourth of all, procedural generation is also very well suited for the culture of game photography (which gained popularity over the years) (Poremba 2007). Last but not least, procedural generation and randomness seem to be very well suited for the iterative model of game

[5] Admittedly, a number of early adventure games, such as *Zack McKracken* contained randomized mazes.

production and distribution such as Steam's Early Access model. One of the reasons for this is that it is much easier to reconstruct virtual worlds created via procedures than its manually designed counterparts.

The second group of reasons which explain the popularity of procedural generation and randomness in contemporary gaming landscape is purely philosophical. I believe that some of the aspects and, especially, promises of procedural generation and randomness create a very specific horizon of expectations which captures the imagination of both: the developers and the players. The first of these reasons is that procedural generation combined with the idea of a random seed suggests the possibility of creating autonomous, non-intentional and non-referential art. This idea can once more be best illustrated by the example of *No Man's Sky*. As pointed out by its main developer, Sean Murray, any live presentation of the game resulted in uncertainty for the creators, because it was not possible for them to predict how exactly will the planet they end up on during the presentation look like and how will the presentation relate to the players experience, once the game is shipped. Even though the game in question was hardly devoid of authors' intent and of referentiality, it is possible to imagine such a case, at least conceptually. It can be even argued (although this claim is obviously rather contentious) that procedural generation contains a promise of game development without any responsibility from the developer (ethical, aesthetical or ludic). To explain this idea better, consider a situation in which a procedurally generated world contains a disturbing symbol, such as a swastika created in a completely accidental manner. Are the developers morally responsible for not preventing this situation? How many disturbing symbols or images do they have to predict and block in their procedural generation mechanism? The idea of the lack of ludic responsibility is probably the most controversial, but it is still possible to imagine situations in which it would have been treated as an advantage. The way I understand it is that procedural generation combined with randomness should be always limited in such a case so it makes the game "playable" or "winnable". Limitations of this type are very likely to be found in most of the procedurally generated computer games, but they do not have to be present in some of the more experimental works.
Once you look at the advantages of procedural generation from the players' perspective it is easy to see that it delivers a number of unique promises as well. First of all, it gives the players a real sense of discovery. If you find an interesting structure in *Minecraft* or *No Man's Sky*, it is quite possible, that you are the only person in the world which saw it. As pointed out above, this aspect is very obviously connected with the culture of game photography, as the possibility of sharing a screenshot of a structure which other players cannot see by themselves (or are extremely unlikely to see) gives additional incentive for the in-game photographers. Last but not least, procedural generation and randomness adds to the sense of gravitas as situations encountered in a given game run may not repeat themselves. This can be seen in any rogue-like which gives the players randomized objects or enhancements. A given run may be extremely lucky and enable the player to proceed further or finally beat a very tough enemy.

Conclusion

As I argued above, randomness and procedural generation started to play a major role in contemporary games. It seems fascinating that the same statistical and mathematical phenomenon could be at the same time employed in practices which are either very highly regarded or heavily criticized. Philosophical reasons of this double nature of randomness (which I tried to identify), demand further study, but there are no doubts that they connect randomness with profound questions about ethics, responsibility of the creator and new forms of expression. Can we safely harness our evolutionary heritage and play with the way our brains react to randomness? Should we prevent adding money to the mix? And what about the promises of procedural generation – is the current rise of popularity of procedurality and randomness only a fad, or is it a beginning of new artistic paradigm? The main point of this paper is that whatever the answers for these questions will be, they should not be asked in separation as the phenomenon which inspires them is the same in both cases.

Games

CHASM. Bit Kid/Bit Kid, PC, 2018.
DIABLO. Blizzard Entertainment/Blizzard Entertainment, PC, 1996.
DIABLO III. Blizzard Entertainment/Blizzard Entertainment, PC, 1996.
DON'T STARVE. Klei Entertainment/Klei Entertainment, PC, 2013.
FRONTIER: ELITE 2. Gametek/Gametek, PC, 1993.
JUST CAUSE 2. Avalanche Studios/Eidos Interactive, PC, 2010.
MEGA MAN. Capcom/Capcom, NES, 1987.
MINECRAFT. Mojang/Microsoft, PC, 2009.
NO MAN'S SKY. Hello Games/Hello Games, PC, 2016.
OVERWATCH. Blizzard Entertainment/Activision, PC, 2016.
PLAYERUNKNOWN'S BATTLEGROUNDS. PUBG Corporation/PUBG Corporation, PC, 2017.
RESIDENT EVIL. Capcom/Capcom, PlayStation, 1996.
ROGUE LEGACY. Cellar Door Games/ Cellar Door Games, PC, 2013.
SPELUNKY. Mossmouth, LLC/Microsoft, Xbox 360, 2012.
STAR WARS: BATTLEFRONTS II. EA Dice/Electronic Arts, PC, 2017.
STRAFE. Pixel Titans/Devolver Digital, PC, 2017.
THE BINDING OF ISAAC. Edmund McMillen/ Edmund McMillen, PC, 2011.
TOWER OF GUNS. Terrible Posture Games/ Terrible Posture Games, PC, 2014.
UNAVOWED. Wadjet Eye Games/ Wadjet Eye Games, PC, 2017.
ZAK MCKRACKEN AND THE ALIEN MINDBENDERS. LucasArts/ LucasArts, Commodore 64, 1988.

References

BBC News, (2017), Call to regulate video game loot boxes over gambling concerns, BBC News, http://www.bbc.com/news/technology-42110066
Blomberg, L., (2013), Sponsored: The World of Just Cause 2 - Using Creative Technology to
Brown, R., (2017), "Loot boxes and skin betting in video games: What are they, are they gambling, and should they be banned in the UK?", Mirror, https://www.mirror.co.uk/tech/loot-boxes-controversy-what-they-11566776
Build Huge Open Landscapes, Gamasutra, http://www.gamasutra.com/view/feature/192007/sponsored_the_world_of_just_cause_.php

Cross, K., (2017), “How the legal battle around loot boxes will change video games forever”, The Verge, https://www.theverge.com/2017/12/19/16783136/loot- boxes-video-games-gambling-legal
Dingman, H., (2017), “How loot boxes are turning full-priced PC games into pay-to- win games of chance”, PCWorld, https://www.pcworld.com/article/3231668/gaming/loot-boxes-ruining-gaming.html
Eagle, A. (2005), ‘Randomness is Unpredictability’, British Journal for the Philosophy of Science, 56: 749–90.
Fisher, S., & Griffiths, M. (1995). Current trends in slot machine gambling: Research and policy issues. Journal of Gambling Studies, 11(3), 239-247.
Frigg, R., 2004, ‘In What Sense is the Kolmogorov-Sinai Entropy a Measure for Chaotic Behaviour?—Bridging the Gap Between Dynamical Systems Theory and Communication Theory’, British Journal for the Philosophy of Science, 55: 411–34.
Futuyma, D. J., 2005, Evolution. Cumberland, MA: Sinauer.
Mi, L., Vitányi, P.M.B. (1997), An introduction to Kolmogorov complexity and its applications, Springer
Garda, Maria B. "Neo-rogue and the essence of roguelikeness." Homo Ludens 1.5 (2013): 59-72.
Griffiths, M. D., & King, R. (2015). Are mini-games within RuneScape gambling or gaming?. Gaming Law Review and Economics, 19(9), 640-643.
Hern, A., (2017), “How much?! – Star Wars Battlefront II and the problem with paid- for video game rewards”, The Guardian, https://www.theguardian.com/games/shortcuts/2017/nov/26/how-much-star- wars-battlefront-ii-and-the-problem-with-paid-for-video-game-rewards
Hood, V., (2017), Are loot boxes gambling?, Eurogamer, http://www.eurogamer.net/articles/2017-10-11-are-loot-boxes-gambling\
McNihol, T., (2003), Totally Random. How two math geeks with a lava lamp and a webcam are about to unleash chaos on the Internet. Wired, https://www.wired.com/2003/08/random/
Nielsen, R., Grabarczyk, P., (2018), Are loot boxes gambling? Random reward mechanisms in video games, DiGRA '18 - Proceedings of the 2018 DiGRA International Conference: The Game is the Message, DiGRA, July, 2018, http://www.digra.org/digital-library/publications/are-loot-boxes-gambling-random-reward-mechanisms-in-video-games/
Poremba, Cindy. (2007), "Point and shoot: Remediating photography in gamespace." Games and Culture 2.1: 49-58.