**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»**

**Факультет комп’ютерних наук та технологій**

**Кафедра інженерії програмного забезпечення**

ДОПУСТИТИ ДО ЗАХИСТУ

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ Олена ГРІНЕНКО

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»**

**Тема:** Мобільна гра в жанрі «Hyper Casual» на Unity

**Виконавець:** Новік Олександр Олександрович

**Керівник:** к. т. н., доцент Гріненко Олена Олександрівна

**Нормоконтролер:** к. т. н., доцент Поважний Василь Петрович

**Київ 2025**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»**

Факультеткомп’ютерних наук та технологій

Кафедра інженерії програмного забезпечення

Спеціальність 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_\_\_\_\_\_\_ Олена ГРІНЕНКО

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2025 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Новіка Олександра Олександровича

1. Тема кваліфікаційної роботи: «Мобільна гра в жанрі «Hyper Casual» на Unity»

затверджена наказом в. о. ректора від «\_\_\_\_\_»\_\_\_\_\_\_\_\_ №\_\_\_\_\_\_\_.

2. Термін виконання проекту: з 12.05.2025 р. по 22.06.2025 р.

3. Вихідні дані до роботи: розробити гру за допомогою ігрового рушія Unity.

4. Зміст пояснювальної записки:

1. Концепт гіперказуальних ігор.
2. Характеристики гіперказуальних ігор.
3. Будова гри.
4. Прототип гіперказуальної гри.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1. Структурна схема …..
2. Функціональні можливості програмного ….
3. Інтерфейс системи автоматизованого проектування ….
4. Схема роботи програми.
5. Демонстрація роботи програми.
6. Демонстрація роботи модулів програми.

6. Календарний план-графік

|  |  |  |  |
| --- | --- | --- | --- |
| № пор. | Завдання | Термін виконання | Відмітка про виконання |
| 1. | Розробка та затвердження графіка роботи | 12.05.2025 | відмітка |
| 2. | Ознайомлення з постановкою задачі, вивчення інформаційних джерел та складання плану роботи. | 13.05-14.05.2025 | відмітка |
| 2. | Підготовка 1 розділу та подання його керівнику | 12.05-18.05.2025 | відмітка |
| 3. | Підготовка 2 розділу та подання його керівнику | 19.05-25.05.2025 | відмітка |
| 4. | Підготовка 3 розділу та подання його керівнику | 26.05-01.06.2025 | відмітка |
| 5. | Підготовка 4 розділу і висновків по роботі та подання їх керівнику | 02.06-08.06.2025 | відмітка |
| 6. | Загальне редагування пояснювальної записки, графічного матеріалу.  Представлення роботи для перевірки на академічну доброчесність.  Проходження нормоконтролю. | 02.06-08.06.2025 | відмітка |
| 7. | Отримання відгуку керівника.  Підготовка презентації та тексту доповіді. | 09.06-15.06.2025 | відмітка |
| 8. | Попередній захист (представлення електронної версії пояснювальної записки, презентації, позитивного відгуку керівника). | 09.06-15.06.2025 | відмітка |
| 9. | Рецензування кваліфікаційної роботи | 09.06-15.06.2025 | відмітка |
| 10. | Здача секретарю ЕК пояснювальної записки: електронної версії кваліфікаційної роботи; презентації доповіді; відгуку керівника, рецензії; результату проходження перевірки на плагіат; довідки про успішність | 09.06-13.06.2025 | відмітка |
| 11. | Захист кваліфікаційної роботи в екзаменаційній комісії | 16.06-22.06.2025  Поставити конкретну дату захисту коли буде затверджено графік захисту робіт | відмітка |

Дата видачі завдання 12.05.2025 р.

Керівник кваліфікаційної роботи:

к. т. н., доцент Олена ГРІНЕНКО

Завдання прийняв до виконання: Олександр НОВІК

**INTRODUCTION**

The rise of hypercasual gaming represents one of the biggest and most interesting shifts in the game development industry over the last decade. Although complex, high-budget video games keep dominating gaming markets, a new powerful hypercasual segment has arisen as a massive economic and cultural current, generating billions of dollars in revenue and gaining a large consumer base. This qualification work addresses the challenge of developing both an engaging and a minimalist, simple gameplay experience, as well as researching the peculiarities of this area of digital entertainment. The relevance of this work is supported by such factors as rigorous expansion of mobile gaming worldwide, evolution of advertisement-based digital monetization, and an energetic growth of demand for hypercasual games that can be easily and quickly developed. Practical application of this subject is demonstrated by a self-designed and developed hypercasual mobile game, which also contributes to development methods in this volatile area of business and entertainment.

The goal of this work is to develop a hypercasual game using the Unity engine. The game has to effectively implement features and characteristics specific to the hypercasual genre, while also maintaining efficient development practices. The theoretical part of this work will also include extensive research on various aspects of the hypercasual gaming field, from its unique composition of business models to peculiar demographics and target player bases. The research notes will lead through the general philosophy of this genre and describe how it manages to stay afloat and bring enormous revenues by seemingly defying normal laws and seemingly intentionally dodging all the standards of traditional game development.

Multiple tasks were performed in this work on different levels and can be arranged in the following list:

1. Learn the concept and defining characteristics of hypercasual games.
2. Identify the features and conditions that make a difference between successful hypercasual games and games from other genres. The business sector of this field, along with technical specifics, is explored in the qualification work.
3. Work out a set of technical requirements for the prototype of the game and elaborate on the choices made during the selection of technologies used for development. The requirements are largely affected by the theoretical research performed in previous chapters.
4. Design and develop the actual game using the Unity engine while sticking to all the necessary steps that define a hypercasual game.
5. Evaluate documented progress and results of development and compare them to commercial standards, and draw conclusions.

The object of this work is the development workflow of a hypercasual game. It consists of all decisions regarding the structure and design of the game, as well as technical implementations of separate game mechanics. This also includes the full development cycle from the very beginning of research of the business domain to the final stages of development. Much attention was given to the constraints and requirements set by the genre.

The subject of this work is the application of the Unity engine to create a gaming experience with respect to simplicity and minimalism that defines the hypercasual genre.

Research methods utilized during the making of this work can be enumerated as follows:

1. Literature review. Close examination of publications about concrete games and market reports gave a solid understanding of this genre.
2. Comparative analysis. To learn how different conditions lead games from diagonally opposite genres led all of them to identical great success, data comparison had to be performed.
3. Requirement analysis. Evaluation of requirements and initial conditions was studied and linked to the output value of a given game project.
4. Technology assessment. Justifications for the given architectural and business decisions were researched.
5. Market analysis. A deep dive into market trends was done to understand why and how this genre became popular.

The scientific novelty of this project is mainly described by how this project effectively shows that hypercasual game development is one of the most accessible means of content creation, with a high effort-to-reward ratio. It clearly shows how this field of development combines creativity with technicality in a very ergonomic and minimalistic way, producing a business entity as a result. It also solves a modern-day issue of short-form content that floods the internet and manifests the rules of content creation. The game includes deep storytelling and expands the meaningfulness of the game beyond a single session.

The practical significance of this qualification work extends to multiple domains at the same time:

1. Fully working application. The created game runs as required by current industry standards and demonstrates the practical instance of this genre’s manifesto. It is a piece of content that can be consumed by the target audience of the hypercasual genre.
2. Contribution to development references. This can be used as a reference by those seeking to become a part of game development in the hypercasual genre.
3. Technical blueprint. The source code and files can be used as a direct template by other developers to implement their own games based on this project’s codebase or techniques.

This work equally focused on both practical skill application for game development and large amounts of research. Ultimately, the developed project is a solid contribution to the area of mobile game development, particularly to the hypercasual segment of it.

# **CHAPTER 1**

**THE CONCEPT OF HYPER CASUAL GAMES**

* 1. **Definition of hypercasual gaming**

Hypercasual games emerged in the volatile world of mobile gaming as a clear distinction from traditional games. Not only have they transformed the way games acquire players’ interest and engagement, but also transformed the business models and development workflow. This category of mobile games first saw the light of day in the 2010s, however, the features of the entire concept started surfacing even earlier.

Hypercasual games can be briefly described as mobile games that sustain characteristically extreme simplicity, wide accessibility, and high playability. Although there is no clear definition of hypercasual gaming in the general gaming industry, players, developers, and analysts have rendered a list of concrete attributes only applicable to games of this specific kind. The very core of hypercasual games is the simplified version of any gameplay mechanic imaginable, stripped to its most basic components. This way, instant playability is emphasized over visual or logical complexity. “Simplicity” in question can be described by an example of gameplay from one of the most popular hypercasual games, “Stack”. The most complex shape in this game has 6 sides and is a mere cuboid (pictured in fig. 1.1).



Figure 1.1

**Technical details**

The term “hypercasual” can give a deep insight into how these games operate. Typically, in scientific terms and definitions prefix “hyper” indicates a high extent of particular value, as it is breaching the regular boundaries of “normality”. These extremes can be observed across the hypercasual field of mobile game development on every possible level: business, development, gameplay, and even budget spending. This is the final point of gaming simplicity, where nothing goes beyond.

From a more technical perspective, hypercasual games are built around a singular atomic action that can be performed on any phone: tap, swipe, hold, or combinations of them. These actions are the glue that holds visual pieces of hypercasual games together and are the center of user interaction with the game. Such a model of interaction is extremely intuitive and requires little to no guidance in the form of game tutorials or instructions. The target user learns to play the game in a matter of just one session. A good unspoken rule of hypercasual gaming is that the game has to be intuitive enough for a new player to understand the controls within just 3 seconds.

Extreme simplicity is noticeable in the visual design of games as well. Most of the hypercasual games resorted to basic geometric shapes like spheres, cubes, lines, or two-dimensional shapes. The color palette is also limited to stay in harmony with the simple aesthetic environment of the game. Not only does such simplicity reduce development costs, but it also makes the game accessible to a wider range of players by removing visual clutter.

Typical gameplay session lasts between a few seconds to few minutes. Usually, there is no logical or storyline progression across the game. In other words, these games are endless, and to compensate for the missing feeling of progress, they utilize scoring systems based on points accumulated during the session, distance reached, maximum combos, etc. Imminent failure is one of the most interesting, although subtle aspects of these games: the player will always eventually lose, and when that happens, another session can be started immediately withing a tap or two. This unavoidable failure generates an endless loop of trial and error, which keeps players engaged.

**Business side**

Hypercasual games can also be defined by their unique monetization strategy. Rather than enforcing a one-time purchase of their product or premium subscriptions, publishers of hypercasual games primarily rely on rigid advertising in their games. In other words, hypercasual games are mostly free, but the user is constantly presented with large numbers of ads in game menus and between different stages of game sessions.

To support a distinct monetization model, the development model has to accept that the quantity of game projects is to be the priority over their quality. This results in a rapid iterative development cycle, in which small teams of 1-3 developers focus on quick development of game prototypes over days or weeks. The final product is tested by actual users instead of game designers (not literal “testing” is meant here. Hypercasual games are still tested for bugs or errors, but the actual concept of a particular game and its code idea is for the market to be evaluated). This brings the initial value of a prototype to the lowest level, unless the game gains high popularity and revenue as a consequence. Such a development model allows for quickly discarding prototypes that couldn’t engage players and focus on a new iteration of development. The ultimate idea behind this way of developing games is to randomly hit the sweet spot for players after randomly producing dozens or hundreds of game ideas and dynamically trying them out on the market. Low development cost allows for blatantly throwing out projects that couldn’t adapt to the industry standards. A generic development process schema is featured in fig. 1.2.

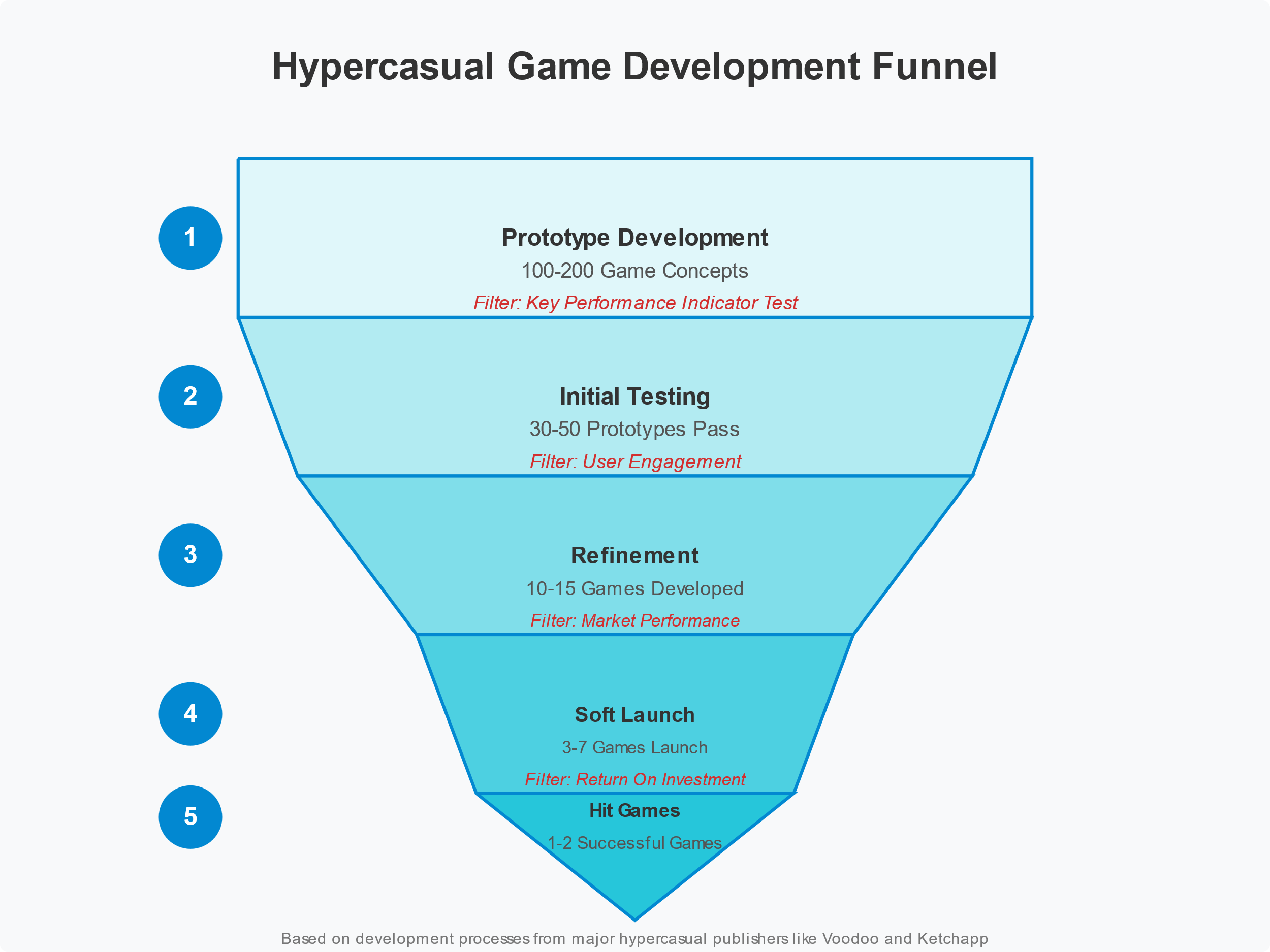


Figure 1.2

Since there is no magic formula for a perfect hypercasual game, developers resort to trying random mechanics or visuals within this genre and hopefully make the user interested enough to download the game.

**History**

It is rumored that the historical background for hypercasual gaming could hide among old arcade games like “Pong” (1972) or “Space Invaders” (1978), however it is not clear whether simplicity of these games had any effect on creativity in 2010s, since at that time simplification was caused by technical limitations of hardware, and not by specifics of target player base or development models.

First instances of hypercasual games laid the foundation of future prototypes. One of such examples would be “Flappy Bird” (2013), pictured in fig. 1.3, which might have influenced many timing-based hypercasual games that were released in the years following.

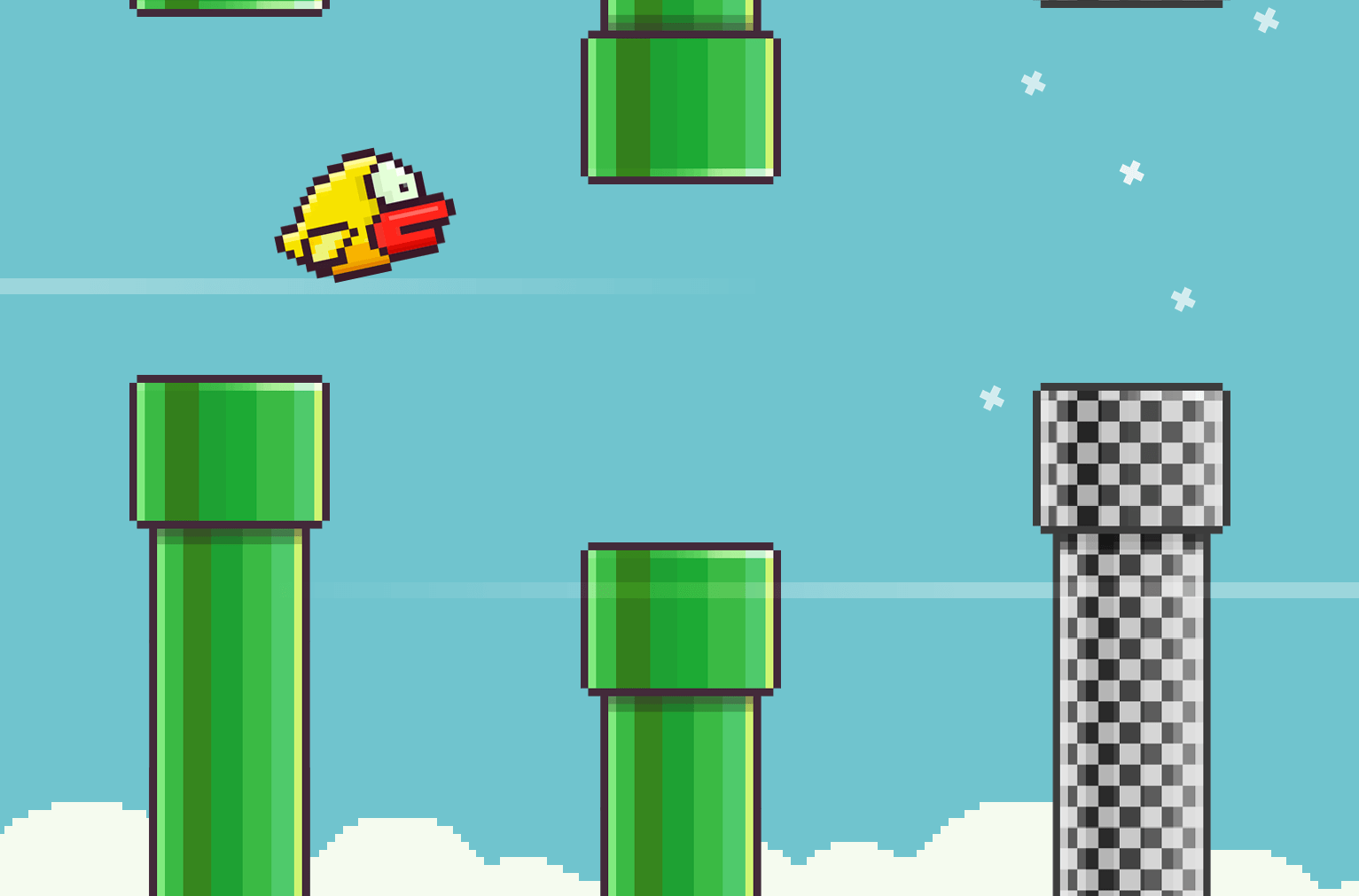


Figure 1.3

Turns out, visual simplicity combined with basic, but excruciatingly difficult gameplay synthesized an actual cultural blast which etched this game forever into gaming history. The idea was quickly picked up by massive game publishers like Ketchapp and Voodoo, who began to mass-produce hypercasual games and by 2018-2019 conquered the top of the Apple App Store and Google Play Market charts.

The COVID-19 pandemic even further accelerated the growth of hypercasual games, as players had to stay home and somehow fill in the big amounts of free time they had during lockdowns: from December 2019 to March 2020, the number of downloads of hypercasual games doubled worldwide.

**Idea**

The philosophy of hypercasual gaming can be described as absolute rejection of the sophistication of any kind: logical, visual, or mechanical. Conventional game genres sooner or later will have to rely on either deep storytelling or complex game mechanics combined with stunning visuals in order to be accepted in the industry and fulfill the player base’s expectations. This is a diagonal opposite to hypercasual gaming, where there is a radical opposition to the idea of “more is always better”. Hypercasual games refuse to embrace elaborate game mechanics systems, user interfaces, interaction models, quests, progressions, as well as stunning visuals. Where they lack in beautiful graphics, they compensate in the ability to properly draw attention of mobile users by offering visually satisfying experiences through high-quality animations, audio, action feedback, color palettes, and timings. In short moments of players’ rest, they offer just what is needed: quick dopamine release through brief gaming sessions, rather than demanding long sessions that require a high attention span and significantly more dedication both in time and in effort. The very basis of the joy of gaming comes from a simple loop of facing challenges of any kind, overcoming them through skill development, and progressing to the next challenge. Hypercasual games are a real proof that this fundamental idea’s existence does not require the presence of marvelous visuals or complicated interaction schemas. By stripping all the optional parts from games, hypercasual games present the players with a very raw, natural form of challenge and skill that can be pulled out of a pocket on a bus stop, during lunch break, or on a couch at home with no particularly high commitment needed. This bare experience of skill development has widened the potential player base beyond any known in the field of high-profile gaming, which is also reflected in relatively equal gender distribution and a lot broader age range.

**Impact and issues**

The cultural perspective of hypercasual gaming shows that hypercasual games practically turned gaming from just a hobby to a mass popular culture movement: everybody plays hypercasual games everywhere. Gaming doesn’t require dedication in the form of separate long-playing sessions, it can now be done anywhere and at any time. Thus, gaming became a routine rather than a hobby.

Everything described so far can be summarized and combined into a single definition: hypercasual games are a separate category of games that feature extreme simplicity in all aspects of gameplay and making, immediate playability, short session lifespan, barebone visual design, high accessibility, and a very wide demographic range as a result. They prioritize raw challenge over complicated visuals and interactivity.

Since hypercasual games promote short sessions over longer, more dedicated loops of gameplay, many negative effects of low-quality short-form content (like Tik-Tok videos) are applicable to hypercasual games: they serve as time fillers rather than meaningful interactions with made-up in-game worlds and lore. Playing a traditional game with significant effort put into its development can be compared to reading a decent-sized book of any non-technical genre, while playing a hypercasual game would, in comparison, be reading a two-sentence summary of the given book. It is still technically reading, but very shallow and meaningless. A very important innovation that can be made to hypercasual gaming is adding a particular kind of storytelling or emotional foundation. Storytelling exists completely independently from game mechanics and can be done in any way imaginable; therefore cannot discard the simplicity of this genre. This project will address the issue and offer a real solution to make hypercasual games more “alive” and less artificial.

* 1. **Prime examples of hypercasual games**

Numerous extraordinarily successful hypercasual titles have been released over the last decade that still rule the market. They are the pioneers of the hypercasual idea of simplicity, extreme accessibility, and addictive gameplay. Despite their minimalistic design on every layer, these games accumulated unimaginable numbers of downloads and decently large retention values. Close examination of these superstars of hypercasual gaming can give a better idea of what defines a successful game in this industry.

**“Flappy Bird” (2013)**

It all began with a cultural phenomenon: “Flappy Bird” mentioned earlier. There is no better example of the potential of hypercasual gaming than “Flappy Bird”. It was released by a single Vietnamese developer in 2013. Very basic pixelated graphics and atomic mechanics did not prevent this game from becoming a global hit. At its peak, the game in which singular taps control a small bird that flies through vertical gaps in pipes generated a whopping $50000 of income daily through in-game advertisements. A coincidental mix of features gave birth to a perfect formula:

* One-tap gameplay: tapping the screen is the only way to control the bird character
* Obvious and immediate challenge: the hardcore and straightforward difficulty of the game was clear in the first couple of seconds
* Overcoming and skill development: the extremely challenging nature of the game created a natural response of “just one more attempt”
* Quick game loop: the majority of sessions last under a minute
* Flat learning curve: the task of the game and means of its completion (despite the fact that the game is practically endless and has no finish line as such) are obvious to any player. No tutorial required

“Flappy Bird” not only became a successful project (in fact, so successful that the owner had to remove the game from application stores due to unmanageable fame and feeling of guilty stemming from parents worldwide complaining about their children abandoning their lives in favor of uncontrollable gaming), but also the most profitable opportunity for those willing to explore the new shiny idea of games that intentionally cut off visual and logical complexity.

**“Helix Jump” (2018)**

Another phenomenally popular game was developed by the Voodoo studio. This game falls into the same “hypercasual” category as “Flappy Bird”, but shares only a few features. In this game, a player has to rotate a tall vertical platform with numerous “floors” with holes in them to guide a ball that is falling from above, avoiding obstacles along the way, as pictured in fig. 1.4.



Figure 1.4

A lot of attention was paid to the sensual, visual, and audible feedback from players’ actions. The factor of addiction was not the game’s difficulty, but rather the feeling of satisfaction from perfectly guiding a ball through a row of narrow holes. Key elements of the game include some aspects that are already known from previous hits:

* Single-action scheme: player uses only one action to control the game by moving the finger left and right across the screen
* Feeling of satisfaction: platforms are broken into pieces whenever a ball successfully falls through them. This creates a feeling of “clearing” particular stages of the game
* Progressive difficulty: rather than hitting with a constantly hardcore difficulty, the game leads through a gradual increase of difficulty
* Short sessions: a parameter that remains a constant in all instances of hypercasual games. Logical ending of a session comes in under one minute
* Clear goal: the game still utilizes the mechanic of counting players’ scores rather than having a conclusive ending of the game. The ultimate goal is to beat own highest record

“Helix Jump” managed to accumulate more than half a billion downloads. This game became a clear example that hypercasual gaming was not just a stutter in the gaming industry, but rather a viable business idea that has a gigantic potential. “Helix Jump” brought its developers $25 million in revenue from in-app advertisements.

**“Subway Surfers” (2012)**

While being an object of unresolved argument as to whether this game can be considered hypercasual or casual, it still has a handful of features that could land it in the hypercasual section of mobile gaming. It was created by SYBO Games and eventually formed a whole new genre of mobile games, so-called “runners”, where a character runs forward and has to dodge all sorts of obstacles and collect power-ups to improve the abilities of characters, as shown in fig. 1.5.



Figure 1.5

This game proposed yet another interesting formula of great success:

* Intuitive controls: slightly more complicated, but still simple enough to get picked up by most players
* Visual appeal: the game keeps the graphics simple, but uses a very versatile and saturated color palette
* Regular updates: early instances of developers making use of frequent updates to their hypercasual projects. New updates brought completely new visual styles and cosmetics while keeping everything else untouched
* Unique progression system: game sessions are endless and increase in difficulty over time, and there is not much to it. However, an unwritten and unconventional goal of the game for many players was to collect all possible cosmetic upgrades to the main character. The most expensive characters required months of grinding for in-game currency
* Competition with other players: the game had a Facebook integration, which allowed players to see players’ friends’ records and try to beat them

“Subway Surfers” achieved a whopping number of downloads of 3 billion installs across all platforms, as well as $155 million in revenue over 10 years. It is still ranked #7 in the Apple App Store under the “Action Games” section. Such results made this game one of the most downloaded games and applications on mobile systems in history. Not only did it become a hit at its time, but it also managed to retain popularity after a decade through developers’ willingness to release frequent updates. “Subway Surfers” is proof that simplicity is never an obstacle to retention and replay value rating of a game.

**“Hole.io” (2018)**

Another hit by Voodoo, which revolves around a simple concept of an infinitely growing hole that consumes objects above it. Like the previously successful “Helix Jump”, the main value of the game was that sensation of satisfaction when small objects collapse and fall through the hole. The user’s interaction that controls the position of the hole is the defining feature that delivers this feeling of satisfaction to the brain (see fig. 1.6).



Figure 1.6

“Hole.io” uses a slightly more complex set of geometrical shapes, but still falls under the category of hypercasual games while also adding some new aspects of gameplay:

* Time pressure: sessions are divided into two-minute sessions. The highest record is defined by how large a hole a player can make during these two minutes
* Unique progression: the larger the hole becomes, the bigger the appetite: the player can consume larger objects with their hole over time
* Environmental destruction: similar to how platforms are destroyed in “Helix Jump”, the hole can eat the whole in-game city piece by piece. Everything on the screen is destructible
* Competition: it is possible to play in the same session with other players and compete with them if own hole is larger in diameter
* Visible progression: the hole becomes larger the more things it consumes. This gives a clear indication of the player’s progress

“Hole.io” was brave enough to bring innovation and new small concepts into hypercasual gaming and was generously rewarded by the player base. As of May 2025, this game was downloaded 1 million times last month and generated $100000 in revenue. It is ranked #1 in the Apple App Store “Sports” games. This time, Voodoo proved that the hypercasual genre, despite its simple concept, still tolerates innovation and variation in ideas.

**“Stack” (2015)**

Minimalist and elegant creation from Ketchapp (another publishing giant, rival of Voodoo) that can be considered the purest and most barebone hypercasual game and an actual masterpiece of game design. The game’s idea is a moving cuboid that needs to be placed perfectly on top of another cuboid by tapping the screen at the correct time (fig. 1.1). If done improperly, the dimensions of the stack get smaller every next move until it is not possible to put any new cuboids on top. The game can be considered a phenomenon due to the amount of attention it managed to obtain over such simple visuals and idea:

* One-touch controls: tapping the screen is the only way to interact with the cuboids
* Pastel visuals: visuals are unexpectedly calming and don’t strain eyes with sudden movements or color variations. The game looks and feels smooth and slow
* Natural dynamic difficulty: unlike in other games, the difficulty curve is not pre-defined. It depends on the player’s actions: the better a cuboid is placed, the easier the next one will be

The game almost feels like a piece of modern art rather than a means of entertainment. Neat visuals and straightforward simplicity brought large masses to the game, generating 50 million downloads so far. Unfortunately, such simplicity also became the main flaw of the game, as it is impossible to bring new updates or cosmetic content to it. The game was eventually abandoned by most of the users and doesn’t produce that many downloads and revenue anymore.

**Comparisons**

We can see a visual comparison of known hypercasual games in table 1.1.

Table 1.1

Comparison of hypercasual games by features and output values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Game** | **Release year** | **Control scheme** | **Session length** | **Key innovation** | **Downloads** | **Revenue** |
| *Subway Surfers* | 2012 | Swipe | 1-3 minutes | Dynamic environment | $3 billion | $165 million |
| *Flappy Bird* | 2013 | Tap | 5-30 seconds | Extreme difficulty | $50 million | $18 million |
| *Stack* | 2015 | Tap | 30-90 seconds | Elegant minimalism | $50 million | $30 million |
| *Helix Jump* | 2018 | Drag | 30-60 seconds | Sensual satisfaction | $500 million | $25 million |
| *Hole.io* | 2018 | Drag | 2 minutes | Sensual satisfaction | $100 million | $40 million |

It would be hypocritical to suggest that there is a clear formula that can be extracted from this comparison, as there is no clear correlation between a particular variable of a game and its popularity or revenue. The only fact that is proven by this table is that hypercasual gaming, like regular gaming, is a field open to changes, new concepts, and ideas. This is the reason why most publishers utilize rapid development workflows of trial and error: it is impossible to know beforehand whether a given prototype will be successful.

However, it is worth noting that there is indeed a list of features that still retain their shapes in all instances of hypercasual gaming: short sessions, monetization through cosmetics or advertisements, short sessions and atomic controls. These features will lay the foundation for the project developed as part of this diploma work.

# **CHAPTER 2**

**CHARACTERISTICS OF HYPER CASUAL GAMES**

* 1. **Simplicity of game mechanics**

At the foundation of hypercasual games lies a strong commitment to mechanical simplicity. For the most part, this particular simplicity defines whether a game will be considered hypercasual, as this is the starting point for the target player. It is not a design preference but a defining factor that separates the hypercasual genre from the others. Deliberately stripped gameplay creates an experience available to anyone within seconds by eliminating all barriers that limit the audience.

Most successful hypercasual games, as previously proven, do their best to adhere to the “single-action” strategy: a control scheme that only consists of one primitive input type. There is no better type of single-action control, but typically they belong to the following list:

* Tap: “Flappy Bird” and “Stack” work around this input type, and usually it is the timing of the tap that makes the game hard and defines the skill factor. This is the most basic and simplest control, but with the timing condition can become the hardest to master.
* Swipe: “Subway Surfers” utilizes this input method. Swiping technically consists of three sub-actions: hold, drag, release, however, this is one of the most natural movements on touchscreen devices; therefore doesn’t create any significant learning curve for new players. This method doesn’t require any external features to make it difficult and can be performed in multiple directions (top, bottom, left, right, or intermediate).
* Drag: “Helix Jump” is an instance where drag is the only control. It is essentially the same as swiping, with the only difference that it cannot be released. This action might require precision when it comes to finger placement and gives games an opportunity to push their difficulty curve towards making the player perform more precise movements with their finger, as well as time them properly.
* Hold and release: essentially a tap that not only has to be timed on the press stage, but also on the release stage. This can generate some interesting gameplay ideas like in “Over The Bridge” (fig 2.1), where the size of a bridge depends on how precisely the player has released their finger to stop the bridge lengthening.

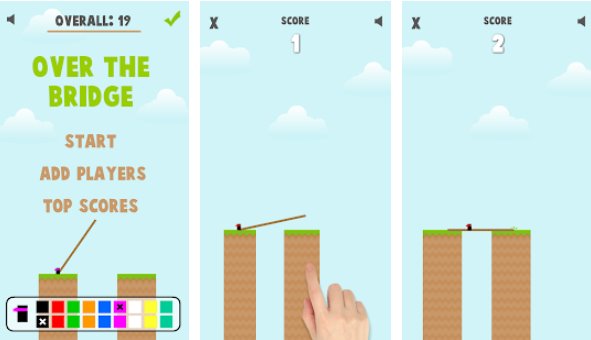


Figure 2.1

The notable difference from traditional games is the complete absence of any complex input schemas with multiple input types combined or separated into stages of any kind. There are no patterns of input that need to be memorized or used to. All of these attributes end up in an unobvious but important feature: hypercasual games can and must be available for playing with a single finger. This exact feature makes hypercasual games extremely available anywhere where it is possible to hold a phone: bus, elevator, staircase, etc.

Instant acquaintance with gameplay stems from the simplicity of input and plays a major role in first session retention rate, which, in turn, defines whether a new player will kick off another session or uninstall the game. Ideally, any hypercasual game must have gameplay simple enough to deduce the controls and main goal of the game just from screenshots or video demonstration of it, but some games occasionally include short animations that explain how to control game objects or characters on the first launch. Controls must be simple enough for players to discover through random attempts and experimentation. If implemented properly, the game will have a completely flat learning curve. In fact, the absence of instructions reinforces the game by letting players explore the gameplay and receive their quick dopamine hit without having to read explanations or manuals. It eliminates the short stage of “figuring out” between the first game launch and the first proper game session. This way, hypercasual games have no alternatives when it comes to quick engagement in moments when significant dedication is not possible.

Visual simplicity is what sets a clear, small goal and doesn’t overcomplicate the scene with unnecessary graphical enhancements or rudimentary detailing. However, it is still important to keep the player engaged within a set of sessions over a couple of minutes, therefore, level design implies diversity where it is possible and acceptable. For example, in stack it is technically not possible to see the same color palette again, as it is procedurally generated before session beginning (fig 2.2).

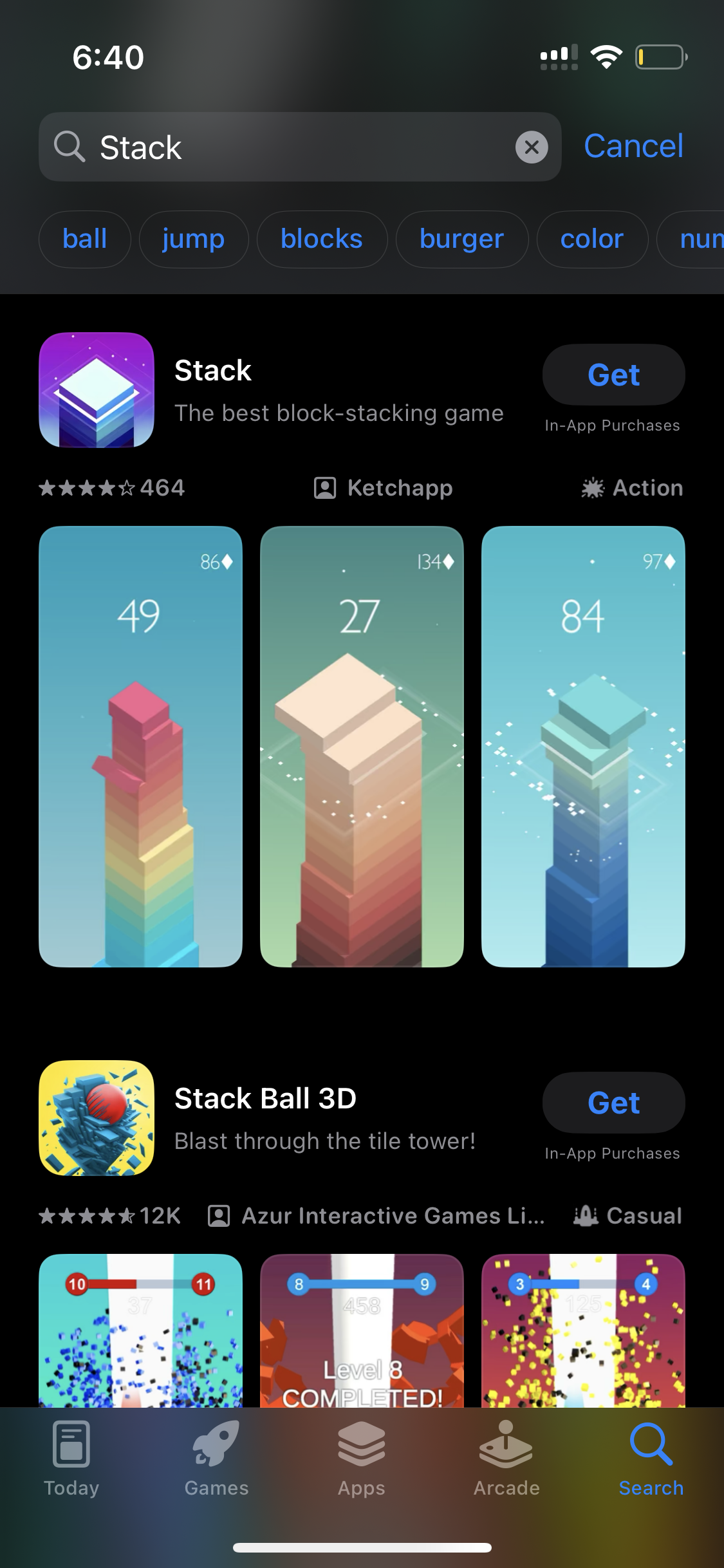


Figure 2.2

Diversification of gameplay is also possible through in-game cosmetics, which also creates a new source of monetization as a bonus. Having new looks every time a new session is started prolongs the lifespan of a single gaming session and sets a goal of trying out different looks or unlocking all possible cosmetic improvements.

Hypercasual games bring joy in different from traditional games ways: either through overcoming hardcore difficulty that lies in the core of the game (like “Flappy Bird”), or by invoking the feeling of satisfaction by breaking objects within the game or letting the player perform series of “perfect” actions: collecting 10 coins at once, destroying dozens of identical objects within a second, or similar (for instance, “Subway Surfers” offers satisfaction through hitting multiple coins in a quick succession that also has a sound effect that increases the pitch after every coin). This sensual satisfaction can be compared with cracking small air bubbles in frozen puddles or throwing a handful of small pebbles into still water and hearing the auditory feedback that almost feels on the skin. Proper combination of the most basic components of handheld devices like visual, sound, and vibration effects can create experiences that players will come for over and over, even after weeks of playing the game.

The same principle applies to the user interface: it must be crystal clear what each of the buttons does, and menus cannot have multiple layers of logic or be a part of actual gameplay. Menus in hypercasual games only serve as places of quick configuration that only allow for adjusting sound effects, music, or some information about the game itself (terms of service, link to publisher’s site, etc.).

The logical picture of a typical hypercasual game loop can be represented by a diagram shown in fig. 2.3.

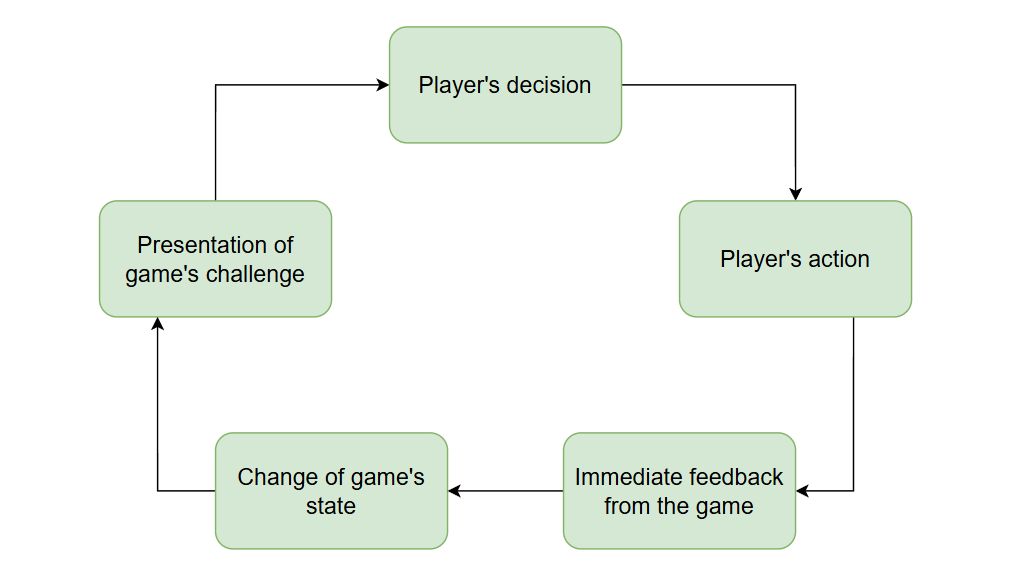


Figure 2.3

The loop is as simple as it can possibly get, and in contrast with traditional games with more layers of complexity, doesn’t have any other loops and produces feedback from the player’s action right away. To understand the difference, one can imagine that in most story-based games, players’ actions might not immediately produce a result but instead show it in later stages of progression. The hypercasual game loop is plain, bland, and highly repeatable.

It is important to understand that the simplicity of hypercasual games is not a limitation in software or hardware. It is a fully conscious choice to focus on raw gameplay and not depth. Such an approach creates a direct challenge to conventional gaming. Although it offers a perfect time-killing opportunity, it still might not always be healthy for a player to dedicate much time to games that don’t boast much depth or philosophical meaning that is available in games of higher rank. This flaw will be addressed by the diploma project without ruining the core principles described previously and in the following sections.

* 1. **Availability across a wide range of devices**

Hypercasual games flipped mobile gaming on its head, not only through simplicity in mechanics but also through exceptional hardware and software compatibility. It is one of the defining characteristics of the genre and a key factor that explains extremely high market occupation. Prioritization of technical availability lets the developers create a gaming experience that can be enjoyed on essentially any modern mobile device worldwide. As a consequence, hypercasual games expanded their audience way further beyond demographics attributed to conservative games.

The technical design of hypercasual games is an embodiment of hypercasual “philosophy”. While traditional games actively push hardware boundaries with photorealistic graphics, realistic physics simulations, and systematic complexity, hypercasual games have an opposite approach: they strip away anything that doesn’t define the game as a piece of entertainment or creativity in favor of maximum compatibility.

Technical minimalism is enforced by several principles:

1. Resource efficiency: hypercasual games must not require significant disk space for installation. Most of the time, application size does not exceed 100 megabytes and is sometimes as low as 20-50 megabytes (smaller than a decently short high-resolution video)
2. Optimized rendering: models, textures, and effects are rendered in the most efficient ways that engines can offer. Simplified graphics are used to ensure low hardware specifications for running the game
3. Low memory consumption: as a result of optimized rendering, less RAM is used to store information within the game. As mobile phones typically have very little RAM, this is one of the most important variables to consider during game development. In-game objects are constantly reused to ensure that no new objects need to be stored in memory
4. Battery consumption: players want to play games anywhere, at any time, and under any conditions, including low battery; therefore, hypercasual games utilize the CPU cores in a way that will minimize power draw
5. Minimal or non-existent dependency on network: an extremely important feature of any modern game is the ability to deliver full gaming experience without connection to Internet. This is especially critical for less developed regions of the world, where Wi-Fi and cellular network is not available everywhere.

Combination of these principles is a foundation for games that have are fully functional on an extraordinarily large spectrum of devices, from the most expensive flagships with modern hardware specifications to the most basic budget phones with hardware and software that was popular years ago. In turn, owners of these devices, who would be completely excluded from gaming otherwise, can enjoy at least some kind of games. This inclusion translates into market accessibility, which guarantees a consumer base anywhere in the world, therefore generating more popularity and revenue.

With the recent boost in technical development of India and surrounding regions, the question of availability is especially ringing, as this part of the world has the highest concentration of player base, but one of the lowest performance capabilities when it comes to smartphones: while around 30% of users can run midcore or hardcore games, hypercasual games boast respectable 80-90%. This way, hypercasual games are one of the very few game genres that have the ability to penetrate the South Asian market deeply enough to make development at least financially sustainable.

Recently, following the progress of web-browsers and social media applications, hypercasual games also expanded to these areas. A very interesting tendency can be observed on Telegram, where developers offered a solid API for app development, which now allows people to create simple games inside the messenger. This way, they fully skip any issues with installation, as the app is stored on Telegram’s server and is streamed seamlessly to users (fig. 2.4).

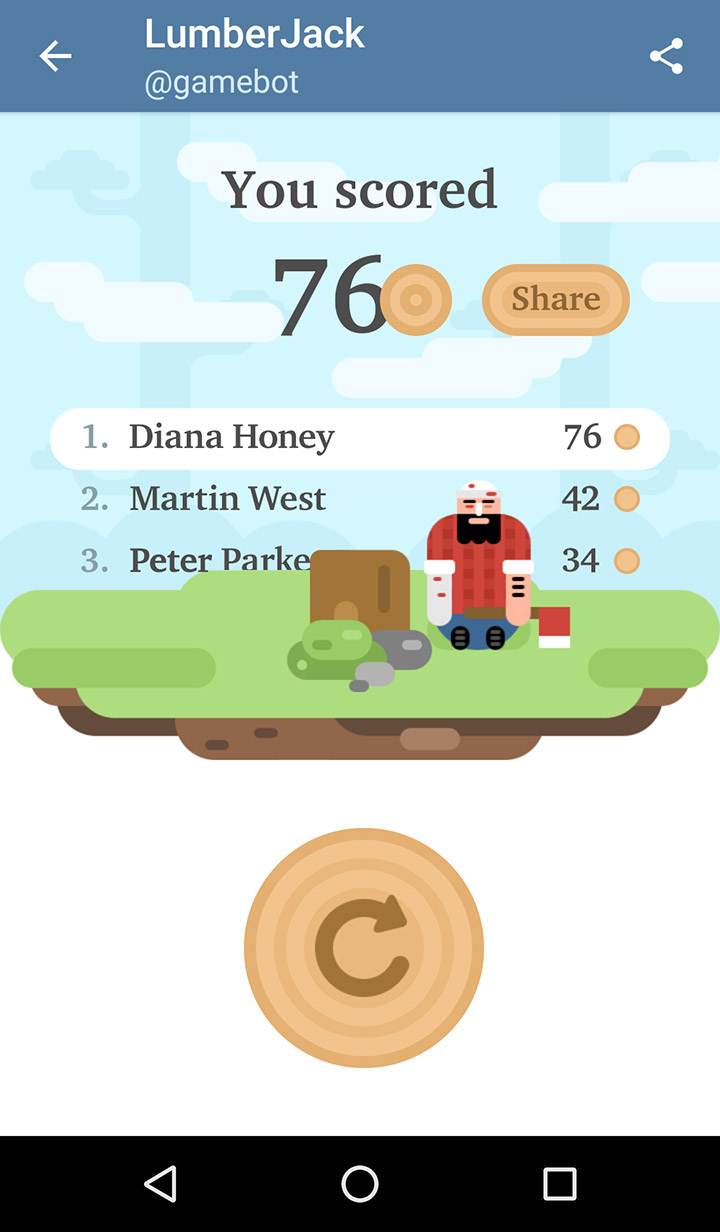


Figure 2.4

These games also have even less impact on hardware and require less computational power to run. As a bonus, this creates another social layer of interaction, where people can share their game records within a single chat and try and beat each other’s results.

When it comes to software limitations, hypercasual games undoubtedly have an upper hand over regular games, since in most cases availability ranges up to Android 4.4, released back in 2013, or iOS 9, released in 2015. Even despite the fact that most people don’t have phones with systems that old, these games still have the technical capability to run on them. Newer releases, though, might not be that compatible, as they are developed with modern technologies and engines that simply don’t allow older systems due to security reasons or internal business decisions. Indian market, despite having a relatively low development index, boasts a pretty low percentage of old Android usage in 2024-2025, as expressed in fig. 2.5.

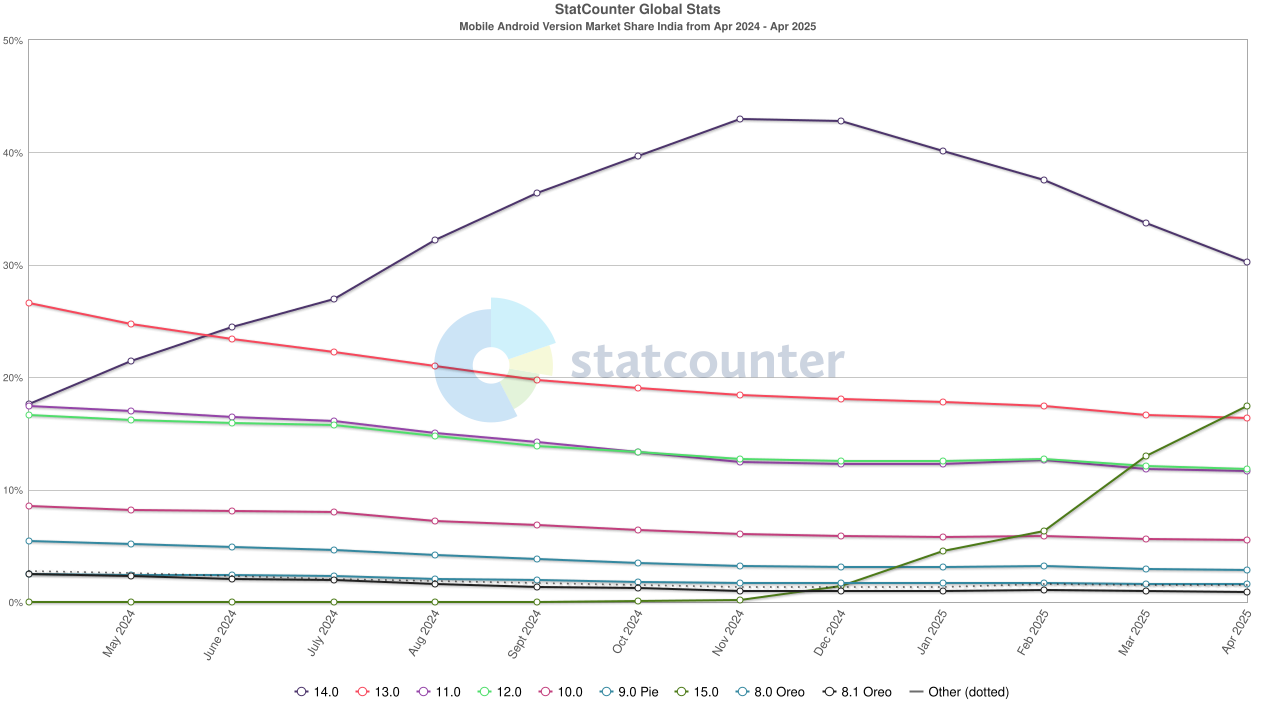


Figure 2.5

In conclusion, exceptional optimization of technical aspects of hypercasual games has a massive impact on demographics and market penetration. The genre managed to find free space in regions unoccupied before. Not only this generates more downloads and revenue, but also boosts market development in these evolving regions.

* 1. **Accessibility to different player groups**

One of the most notable features of hypercasual games is their capability to expand beyond traditional gaming demographics and hit player groups that previously were unable to enjoy the privileges of gaming. Such wide inclusion in terms of demographics demonstrates how just a thorough design of a game can end up being a very universal piece of entertainment. Accessibility of hypercasual games goes beyond technical compatibility and does its best to include a wide range of players with different cognitive, cultural, and social backgrounds.

Hypercasual games modified typical traditional gaming demographics: younger players aged between 18 and 35, with a dominant male side, as well as individuals who mostly have extensive gaming experience. These demographics were “normalized” in a way that, for hypercasual games, they now include a much more balanced distribution among player groups. For instance, hypercasual games are much more popular among female players than regular genres, with some reports stating that women play hypercasual games more than men. Since hypercasual games are free and available anywhere, they are more tolerant of player categories with lower incomes. And while traditional high-budget gaming is only concentrated in technical development hubs like North America, Europe, and Asia, hypercasual games also managed to conquer players in less developed regions, thus having a more global distribution.

Age specifically is the most evident indicator of how diverse player groups are in the hypercasual genre. Low complexity is particularly welcoming to ages that don’t yet have any experience with serious gaming and might not know the typical gaming mechanics or patterns to be able to efficiently enjoy high-tier games. Hypercasual games don’t require investing significant time into climbing the learning curve. Thorough design led gaming to a genre that can be played by the youngest individuals.

Ages 6-17 might find hypercasual games instantly engaging due to intuitive controls and the absence of any obstacles on the path to actual gameplay. They don’t require any reading or complex strategic thinking. They provide clear visual or sensual feedback to players’ actions and immediately reward in-game accomplishments.

Ages 18-65 use hypercasual games to fill in the free time in their dynamic, busy lives. Hypercasual games offer quick stress relief and a dopamine hit in places or times where traditional gaming is not an option at all.

Ages 65+ are one of the most abandoned and underappreciated player groups. It can be compared to the youngest age group, as simplicity replaces gaming experience for older people who might not be able to comprehend the context of more serious games, but still want to enjoy digital entertainment. They make up a decent segment of the hypercasual genre’s player base. Several key factors contribute to such wide accessibility of hypercasual games:

* Simple and clear motor skill development: players only have one thing to worry about; no complex systems, no unpredictable outcomes, just immediate input and reaction to it
* Progression pace control: In many hypercasual games, players choose how quickly they progress or increase difficulty. No competition from other players
* Contrast in visual design: objects are clearly visible in the environment and have a very intuitive purpose, if designed correctly

Hypercasual games boast an interestingly high gender inclusion: thanks to the hypercasual genre, most mobile gamers are now female, although the numbers are close to a perfect balance between the two genders. This is likely due to hypercasual games having stripped off all visual features that would otherwise put them into categories “for boys” and “for girls”. Instead, they offer gameplay with abstract shapes that have abstract goals and exist in abstract environments; therefore, they don’t form repelling gameplay, gendered assumptions, and exclusionary elements. They don’t use characters with tempers that would typically be preferred by players of one gender over another.

While hypercasual games still offer competitiveness through leaderboards and high records, they are not as aggressively competitive as most of the online games in the traditional category: there is no feeling that everyone is significantly better, since there is generally not much room for improvement due to games being so simple in nature. Hypercasual games also don’t enforce direct social interaction to have competition, and rather use scoreboards to determine who is better.

Neutral, unflavored look of hypercasual games removes any concerns about violence or other explicit gameplay features that could harm certain younger demographic groups or in some way restrict them from enjoying games.

The same neutrality can be attributed to the cultural and linguistic aspects of hypercasual games: they don’t represent or show any specific cultures, and don’t require knowledge of languages to be able to play them. Even if some markets require some cultural adjustments to the gameplay features, it is easy and inexpensive to release slightly modified versions of the same game in different regions. Minimal localization requirements allow publishers to globally release their projects simultaneously on different ends of the worldwide market.

Hypercasual games, as a result of their technical simplicity, strongly tolerate markets with players who have particularly low incomes and can’t afford to pay for games upfront in a single purchase. Not only do people have to purchase expensive gaming hardware, but they also pay for games. This is a non-existent issue in the hypercasual sector: these games are available to practically anyone.

Hypercasual games typically have accessibility features for people with all sorts of disabilities. Naturally, simple controls are also easy to master through accessibility features like voice input instead of sensor tap. Also, it is always a good practice to add color schemes for colorblind people and put a feature toggle in settings. Hypercasual games that don’t revolve around auditory features (which is quite a rare, but interesting idea) don’t require sound at all, and can still be played on muted mode. It is usually the visual feedback that provides proper indications and responses to players’ actions.

Cross-generational gaming is a rare occurrence that is possible through hypercasual games. Children can play the same games with parents or grandparents without having to worry about learning curves or difficulties in understanding of gaming concepts. This creates a shared experience across different age categories. As a bonus, such games can serve as a bonding opportunity for family members.

There is also behavioral diversity in hypercasual games: both hardcore gamers and casual gamers can enjoy the same experience, since player progression is defined by their own actions. While highly competitive and dedicated players can focus on beating the highest record on the table, more relaxed people can just enjoy the sensual satisfaction over a 5-minute game session without worrying about records.

Significant accessibility of hypercasual games to various player groups is likely the most significant achievement of the genre beyond commercial success. Hypercasual games embraced the idea of having a more neutral and generic gaming experience, rather than features only suitable for a specific player base. These games welcome any player regardless of age, gender, income, or gaming preferences. This genre is a sample of universal entertainment culture that has no barriers that previously excluded it from certain markets. This “democratization” of gaming is a great tendency that shifts the view of gaming and entertainment in general. It turns out that the path to a more global player base lies not in the creation of complex systems and fabulous visuals but in the exact opposite: simplicity.

* 1. **Intentionally low level of replay value**

A unique feature of hypercasual games that might look paradoxical at first glance is their intentionally short game cycle and very limited replay value. This order contradicts the rules of traditional game design, but is, in fact, a beneficial side of the genre. Regular games do their best to deliver long-lasting gaming sessions that would make players keep coming back after numerous playthroughs of the same chunks of the game. Long-term engagement defines whether a game will be “fun”. This requirement is different for hypercasual games, as they are never supposed to be played for long periods of time (longer than 10-15 minutes) at a time. Usually, publishers abstain from spending time and budget on increasing replay value. Instead, they focus on packing those short-lasting sessions with bursts of content and engagement. Such a strategy displays a very intricate understanding of the psychology of the target player base.

The counterintuitive philosophy of intentionally harming the “replayability” of games is a general narrative in hypercasual publishers. “Burn bright and fast” is an expression that can best describe how they treat their projects. While traditional development schemas imply investing in complex progression systems, storylines, deep emotional meanings, or social features to make people play the game as much as possible, hypercasual developers embrace a much faster pace of work.

There are several reasons why this approach works in the hypercasual field, while it fails to live in traditional game development:

* Quick content consumption: players can see the full kaleidoscope of experiences at once. This engagement overload completely eliminates the need for any depth in meaning that would make the replay value higher
* Easy development: no need for variation in content means no work. Projects can be developed faster
* Portfolio-oriented business model: rather than releasing a major hit once every few years and polishing the same stone in an attempt to turn it into a diamond, hypercasual publishers spit out dozens of games per month. Players will get tired of one and switch to another with no financial consequences for the studios
* Volatility of the market: quick development means quick response to changes in trends. If a particular game type gets a significantly high amount of attention from players, a new similar one can be released right away in no time to keep the trend going.

Low replay value might look like an unreasonably destructive characteristic of a game, but in the hypercasual world, it is the only thing that makes the business side of the industry so unique and efficient.

Limited replay value results in different patterns of psychological satisfaction, both positive and negative. Understanding of these patterns will give an explanation for why a massive group of people still prefers to play hypercasual games and what kind of issues there are to resolve:

* Instant reward: the highest level of satisfaction delivered to hypercasual gamers occurs in the very first couple of sessions. Players experience rapid skill growth in a game they are playing first. This exponential curve eventually becomes flat, and players move to another game
* Accumulated frustration: in case if skill level stops growing, failure can become more frequent and more annoying. Eventually, players will drop the game since there are no other alternative ways to play it
* Oversaturation: the most important flaw of the entire genre, which will be addressed in the project of this diploma work, is cognitive overload caused by short game sessions packed with demands for focus and attention. Rather than watering down saturated parts of gameplay with more relaxed ones, hypercasual games fully focus on delivering a massive sensory blow. This creates an illusion of satisfaction that brings more mental damage than joy

# **CHAPTER 3**

**DESIGN OF THE GAME**

* 1. **Overall requirements for the game**

To develop a hypercasual game, a thorough understanding of technical limitations and principles of healthy and efficient software design has to be established. Based on features of hypercasual games and their unique market environment, general requirements for an endless runner featuring puzzle elements will be outlined. Requirements will consist of both technical and functional considerations, but will also tackle questions of accessibility and business perspectives.

In general, this project can be categorized as an endless runner-like hypercasual game. The idea of the game revolves around a concept of a player who indefinitely runs from a deadly hazard that is following them. The player uses the environment to escape the danger, but will imminently fail due to obstacles that either make escape more difficult or directly end the session. Gameplay and ultimate goal of the game are very similar to those presented in “Crossy Road” (2014), where the player is in control of a character that is trying to cross multiple lanes of the road, which has both static and moving obstacles, as shown in fig 3.1.



Figure 3.1

It is a single-player game, and game sessions are aimed to last under a minute, or at least not longer than 3 minutes for a typical new player, in order to respect typical hypercasual game timings.

The control scheme must consist of atomic actions that are intuitive and can be learned in a few seconds and mastered in a couple of game sessions. Since the game’s target software platforms are Android and iOS systems, it is mandatory for the input system to operate on touchscreens, but alternative methods of input are welcome.

Mechanics-wise, it is a rule that the game’s complexity must stem from environmental conditions (constant pressure from it, occasional hazards or enemies, etc.) and not the complexity of controls. The game’s main limitation on players’ actions can be the unavoidable danger that makes the player perform actions and not remain idle. If the game enforces new mechanics, progressions, or power-ups, then they must be intuitive and not require significant experimentation or tutorials. Overall, players’ actions and game’s feedback loop should be applicable to the flow explained in fig. 2.3.

The immediate challenge is key. From the first second of the session, players should be aware of the game’s main constant hazard. It has to be intuitive and simple enough not to require any warm-ups or tutorial phases. The game algorithms must be in charge of driving the engagement flow and directing players’ actions, but at the same time, not be responsible for them: it is up to the player to decide where to go and how to solve the immediate challenge. However, time limit and reaction time have to be enforced, as in other runners.

The puzzle element is what can potentially solve the issue of cognitive overload and meaninglessness of the genre: by extending control skills with actual problem-solving, the game’s replay value is increased without any violations of other key principles of the hypercasual genre. The game has to arrange an environment that will let the players think 3-5 moves ahead and figure out the best path of escape from danger.

Endless sessions are the defining features of endless runners, therefore, environment generation has to maintain consistency and become more hazardous the longer the session lasts to make sure that the difficulty curve doesn’t flatten and become boring. The endlessness must be diversified by environmental changes that would make consequent sessions distinct, at least by their visual appearance. In-game algorithms must sustain and support the thrill of exploration if players’ sessions become too lengthy and repetitive. Additionally, progression must be recorded and displayed clearly to the player to represent their achievements and skill development over time.

The game must have generally responsive actions. They should never feel slow, sluggish, or hit players with loading screens. The time between session ending and new session start has to be as small as possible to guarantee seamless connection between game loops. Player movement must feel snappy and have no artificial lag, and if sensitivity is a thing, it must have a slider to control its strength. Everything about the gameplay should feel lightning fast and rigid.

Procedural generation has to be treated as a key feature that can bring total play time up. For this reason, generation has to be refined to make sure that it doesn’t become a bottleneck in players’ skill development. For instance, the environment has to be generated in a predictable structure, like a grid. This brings significant simplification to players’ understanding of the in-game world. The world has to maintain a consistent challenge and keep the player engaged by constantly making them perform actions. All generated parts of the world must be solvable and valid. The game must not put a player into a situation that doesn’t have a way out, and not even let players voluntarily end up in them. The world has to be random enough to reject memorization of walkthrough patterns. In other words, every session has to be unique. The world should have multiple ways to navigate. This will significantly widen the freedom of choice.

Difficulty must be scalable in order for the game algorithm to adapt to the player’s development of the game’s knowledge. This can be done through decreasing the time it takes to make a decision or through introducing more sophisticated obstacle generation that would require more thorough puzzle solving. The skill balancing should also be rigid enough for high-tier players to still be willing to play the game.

When it comes to visual design, minimalism must be embraced as much as possible, but still have space for diverse looks of the game through different color palettes, varying environmental objects, etc. The game must look consistent, predictable, and at times even visually repetitive. The game is to maintain high visual contrast and make sure that all objects can be seen properly. Points of interest need to look and act intuitively enough for the player to understand what they do and how they can be interacted with right away.

Aesthetics are always a good feature to focus on. To maintain the previously mentioned requirements, the aesthetics must enforce a minimalistic look of the game, suggest a limited, but well-designed palette of colors that will be applied to surrounding objects or the user interface. Geometrical shapes should be favored over fabulous textures and effects.

The user interface must only contain functionality that is absolutely necessary for the game: sound controls, accessibility features, language switch, etc. Unnecessary use of complex user interface elements (dropdown lists, scrollable lists, etc.) is unacceptable. Usage of overlapping menus should be done with care and not include too many levels of nesting, as it is confusing to a hypercasual player.

Feedback, both visual and sensual, is among the most important features. It delivers the dopamine hit and is the defining reason why people are playing hypercasual games. All sorts of collisions, changes of state, and interactions must be followed by auditory and/or visual feedback. The more satisfying the feedback, the better.

Technical requirements are largely identical for all hypercasual games. To grab the widest range of players, hypercasual games must employ the most intricate optimization techniques and make sure that any entry-level device can run them.

More than 95% of Android users worldwide use Android version 5.0 and above, therefore, this version will be the bare minimum platform requirement for the game.

When it comes to telemetry, the following numbers should be considered as minimal:

* Frames per second: at least 30 FPS on devices of the lower margin, at least 60 FPS on mid-range phones
* Memory usage: no more than 500 MB of RAM has to be used at a time
* Loading time: no more than 3 seconds at launch

Total application size must not exceed 200 MB. This will ensure quick download and install times, as well as respect the internet connection of those who don’t have stable access to the global network. The game must also work without an Internet connection if it doesn’t have any social features, or at least disable them in a modular way, if the connection cannot be established.

Accessibility standards should also align with those widely used in the industry. Minimal complexity of the control scheme should be followed by tolerance for accidental or imprecise gestures provided by a threshold or error margin. Gameplay should not require complex combinations of multiple gestures. Learning curve should start low, walk the player through the basics of gameplay, and then hit them with more challenging situations. The cause for players’ failure must be clear and eligible for improvement. Gameplay cannot depend on the visibility of small or poorly contrasted features of the game environment.

The game should be as inclusive as possible. This means that it cannot be based on specific culture, age, or gender. The looks and concepts of the game have to be as neutral as possible to include any player group imaginable.

It would be beneficial for the game to have multiple interface language options if textual indications of menus or actions are needed.

The amount of testing put into quality verification of the game will define how refined a product players will have on their hands upon first launch of the game. There are multiple levels of quality acceptance, and they should all be followed to make sure that the final game is bug-free. Since gameplay-wise, hypercasual games are simple enough only to require manual testing for the detection of significant flaws, no test coverage is mandatory. Not only can it slow down development and waste project budget, but it is also rudimentary. For the most part, performance metrics are the most valuable object of interest. FPS has to remain above the acceptable margin all the time and cause no sudden spikes or freezes. The same applies to memory usage: memory leaks are to be investigated to avoid having performance degradation during long game sessions. All sorts of edge cases and input spamming must be investigated.

The codebase should be ready for future additions, updates, and extensions in case the game becomes a hit and new content or bug fixes need to be released. This can be achieved by following the same programming and design principles as in traditional game development. Hypercasual games, despite having low value on all levels of business, should be initially treated as long-lived projects.

The described requirements form a solid foundation for developing a hypercasual game and follow all principles that have evolved over the last decade of hypercasual gaming. They address specific challenges of the creation of endless runners and try to solve the issues of modern short-form content by extending the game’s replay value without violating other core principles.

* 1. **Reasoning behind the technologies used for the development of the game**

The technological stack used to develop this hypercasual game had to go through thorough consideration, primarily based on previous programming experience and pricing. Other, less critical factors were: ease of use, learning curve, development efficiency, integrations, and ability to meet the requirements of development in the hypercasual genre without overloading the workflow with redundant tooling or functionality. Ultimately, the desired collection of technologies had to be free of charge in the first place, and support very rapid development of around 2 weeks from very beginning to very end. Each individual choice was based on a problem that this tool can solve, and not a random pick.

Table 3.1 describes the results of research that was done on development and design tools that were and could be used in the making of the project.

Table 3.1

Results of the research on the technological stack

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of the selected tool** | **Purpose of the tool** | **Reason behind the selection of the tool** | **Alternative tools** | **Reason why alternative tools were not selected** |
| Unity Engine | Game engine | Extremely rich documentation and knowledge base, previous experience | Unreal Engine, Godot | Unfamiliarity |
| C# | Programming language | Absence of alternatives within the selected game engine | Visual scripting | Raw feature, poor flexibility |
| JetBrains Rider | IDE | Powerful programming assistance, autocompletion, and live bug detection | Visual Studio, Visual Studio Code | Fewer coding features, bad performance |
| Blender | 3D modelling | Free to use | Cinema4D | High price |
| Pixlr | Photo editing | Free to use | Adobe Photoshop, Gimp | High price, unfamiliarity |
| git | Version control system | Absence of alternatives | None | None |

**Unity Engine**

The selection of this engine was based primarily on previous experience and decent knowledge of how it operates and how to work in it. This has saved a massive amount of time and invested in actual development instead of learning and experimentation. On a general scale, Unity Engine is the most popular engine for hypercasual game development, since a whopping 71% of mobile games of all genres are made using it. This results in a knowledge base and online forum activity being multiple times larger than that of Unreal Engine and Godot. The size of the knowledge base defines how quickly a problem with understanding the engine can be solved whenever it appears. With Unity, most of the issues can be resolved by looking up the solution on programming and game development forums, as a question that arises during development is most likely already answered somewhere on the Internet. This explains why Godot, only a 10-year-old engine, was rejected (not counting the fact of the absence of experience). Unreal Engine is only suitable for high-tier games with realistic graphics and complex systems; therefore, it creates an unnecessary load on hardware even in hypercasual projects, as well as produces built files of a higher size than in Unity.

Unity also has the ability to debug games remotely, via a USB cable connected to the phone on which the app is running. This is particularly useful for debugging issues that are not reproducible in Unity’s development environment and only appear on target users’ hardware or software. It is also a good way to check performance metrics on target devices and make sure that games run smoothly enough after the build.

Generally, Unity Engine is a lightweight game engine that greatly handles small projects with basic graphics, has all the necessary tools for user interface components, audio, animations, logging, debugging, and profiling. Documentation is large enough to cover most of the framework's use cases. It supports almost all modern platforms; therefore, games can be ported both to iOS and Android, and tested on Windows or Mac along the way.

**C#**

C# is the only programming language natively supported by Unity Engine. The standard library of Unity is exposed as C# methods for various engine components. It is technically also possible to import C++ DLLs into C# code and call functions, however, this approach has an extremely narrow use case (typically for components with highly critical performance considerations) and doesn’t fit into hypercasual game development. Some older versions of Unity Engine have native support for UnityScript and Boo, but those languages are far below C# when it comes to readability of code, performance metrics, and compatibility with frameworks, libraries and other languages:

* Syntax sugar: C# has numerous ways of neatly defining or shortening elements of code into more readable syntax structures
* C# code is much faster to compile and run, even with more features. It is also easier to debug
* A large number of frameworks and libraries can be imported into C# code

Due to its design, C# code is always safe in terms of memory: it is always managed by the Common Language Runtime, therefore doesn’t cause memory leaks, which are notoriously difficult to fix in applications with thousands of lines of code.

**JetBrains Rider**

Normally, Visual Studio is favored over Rider, since it is free of charge, unlike the latter. However, with the student’s license, Rider offers all of its benefits for free, and it is a much more powerful, performant, and intuitive tool than Visual Studio. Very rich autocompletion and pre-set shortcuts that generate whole chunks of code, make programming fast and satisfying. Its interface doesn’t overload with tons of buttons and dropdown menus. With proper plugins, Rider can detect places in code that can potentially cause unexpected behavior in the game. It is also fully supported by Unity and will automatically connect to the game’s process during debug sessions when they are launched from Unity.

A very prominent feature of Rider is ReSharper, which is a component that is responsible for live code analysis, refactoring, and navigation. It makes jumping between dozens of files seamless and quick. ReSharper not only detects performance issues and logical bugs, but also suggests fixes and is extremely effective at applying them across gigantic projects. Every warning and error is highlighted and can be either muted, fixed, or explored and explained by documentation within one click or key combination.

**Git**

Git is a version control system that doesn’t have any prominent alternatives. It is the most popular way to store, compare, and manage changes to any files, as well as collaborate with other developers and merge changes to the same parts of code in a useful way. It is a very powerful way of backing up changes and reverting to them in case some irreversible damage to the project is made.

**Blender**

Blender is one of the very few free 3D modelling applications. Being free is what made Blender the most popular choice among amateur and beginner specialists in 3D modelling. Since hypercasual games never require any sophistication when it comes to graphics and modelling, there is no point in using paid alternatives.

Just like Unity Engine, Blender has a very large fanbase; therefore, issues that appear during modelling can be resolved within minutes.

Some other auxiliary tools were used during development, for instance, DoTween, which is an extension package for Unity Engine. It serves as a very powerful animation engine and can be used both for in-game objects and UI elements to deliver satisfying and neat scaling and fading effects. It is an alternative to Unity’s animation engine, which only allows pre-defining a set of animation keys that control which position, orientation, and looks certain objects will have over time. It is not always flexible enough to cover some requirements for mechanics or visuals, therefore, DoTween is of much use when that flexibility is so desired. Since it is a code extension, it is very configurable and can create a massive number of various visual effects, arrange them in sequences, and invoke them whenever suitable.

Coolors.co is an online tool that can generate aesthetic color palettes. Every newly generated palette contains a set of colors that, according to the color theory, fit each other, and therefore can be used as textures for in-game objects to produce pleasant, good-looking visuals. Multiple color palettes can be generated and later used randomly to paint the game environment.

Pixlr was used for image editing. It is essentially a free clone of Adobe Photoshop that is accessible through a web browser, which reflects on its significantly diminished computational capabilities. Usually, UI elements or 2D graphics for hypercasual games are drawn in Adobe Illustrator or applications alike, but for this project, a free unlicensed pack of graphics was used to import into Unity and then assigned to UI components. It is good enough for splitting an atlas with multiple UI icons into separate files, but can also be used to draw original graphics.

In conclusion, the technology stack only comprises tools that are free to use, offer features suitable for the peculiarities of hypercasual game development, and don’t overload with unnecessary functionality.