Game Design Patterns in Endless Mobile Minigames

By
David Cao

Supervisors
Dimitris Paraschakis
Radu-Casian Mihăilescu

Examiner
Jeanette Eriksson
Contact information

Author
David Cao
E-mail: cao.david@live.de

Supervisor
Dimitris Paraschakis
E-mail: dimitris.paraschakis@mah.se
Malmö University, Department of Computer Science

Supervisor
Radu-Casian Mihaiescu
E-mail: radu.c.mihaiescu@mah.se
Malmö University, Department of Computer Science

Examiner:
Jeanette Eriksson
E-mail: jeanine.eksson@mah.se
Malmö University, Department of Computer Science
Abstract

Mobile apps have emerged ever since the smartphone has been established into most peoples everyday life. Almost half of those available apps in the app stores are mobile games. We study game design patterns specifically for endless mobile minigames, as they are one of the emerging categories. This genre has become popular in the app stores with its unique characteristics which include very short play session iterations and its minimalist design. Game design patterns are focused on the interaction with the player and provide knowledge and experience with regards to games in general. Not only are they beneficial for game designers, but also for developers, practitioners and possibly researchers, as patterns provide a common terminology to share information between different professions.

We conduct a case study including five example games and analyze endless mobile games to identify and create genre specific game design patterns. We search for commonalities and major aspects of endless mobile minigames to facilitate the production of such games for developers. To confirm our results, we implement a prototype of an endless mobile minigame, which is then evaluated through a survey.

The result is a collection of game design patterns based on our cases. The questionnaire reveals which of those patterns are relevant and should be considered when developing an endless mobile game. The result outlines that game design patterns are considered supportive when designing a game, however requires adjustments and revisions.

Keywords. game design patterns, mobile, infinite, endless, game, minigame.
6 Evaluation

6.1 Game Design Patterns Evaluation .............................................. 43
  6.1.1 Results ................................................................. 44
  6.1.2 Discussion .............................................................. 51

6.2 Game Enjoyment of Brick Hit Evaluation .................................... 53
  6.2.1 Results ................................................................. 54
  6.2.2 Discussion .............................................................. 56

7 Conclusions ................................................................. 57
  7.1 Thesis summary .............................................................. 57
  7.2 Regarding the research questions ........................................ 57
  7.3 Validity of results and limitations ....................................... 58
  7.4 Future work ................................................................. 59

Appendix A - Game Enjoyment Questionnaire .................................. 60

Appendix B - Game Design Pattern Questionnaire ............................. 62

References ................................................................. 64
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google Play Store - &quot;New Top Game&quot; Apps Ranking, February 1st, 2016. Endless mobile minigames are marked with number 1 and 2</td>
</tr>
<tr>
<td>2</td>
<td>Evaluation plan of RQ1 and RQ2</td>
</tr>
<tr>
<td>3</td>
<td>Case 1 - Drill Up</td>
</tr>
<tr>
<td>4</td>
<td>Case 2 - Doodle Jump</td>
</tr>
<tr>
<td>5</td>
<td>Case 3 - Temple Run 2</td>
</tr>
<tr>
<td>6</td>
<td>Case 4 - The Line Zen</td>
</tr>
<tr>
<td>7</td>
<td>Case 5 - Crossy Road</td>
</tr>
<tr>
<td>8</td>
<td>Pattern &quot;Endless mode&quot; example. Source: The Line Zen - The game continues until the blue ball hits a red obstacle/wall.</td>
</tr>
<tr>
<td>9</td>
<td>Pattern &quot;Quick progressive difficulty&quot; example. Source: Doodle Jump - The game increases its difficulty after time by adding more monsters and less platforms to jump on.</td>
</tr>
<tr>
<td>10</td>
<td>Pattern &quot;Random level generation&quot; example. Source: Temple Run 2 - The last piece of the level is always randomly generated (land, gap, tracks, water) the further the player is advancing.</td>
</tr>
<tr>
<td>11</td>
<td>Pattern &quot;Protagonist&quot; example. Source: The Line Zen (left) - The blue ball is the main actor in the game. Temple Run 2 (right) - A person running represents the main actor in the game.</td>
</tr>
<tr>
<td>12</td>
<td>Pattern &quot;Obstacle&quot; example. Source: Crossy Road - Obstacle trees (static) on the left and vehicles (dynamic) on the right are highlighted.</td>
</tr>
<tr>
<td>13</td>
<td>Pattern &quot;Supporting object&quot; example. Source: Drill Up - The spinning discs are used as platform for the player to advance in the game. (Color highlighted)</td>
</tr>
<tr>
<td>14</td>
<td>Pattern &quot;Currency&quot; example. Source: Crossy Road - The coins are randomly placed throughout the level and are being added to the total number on the top right.</td>
</tr>
<tr>
<td>15</td>
<td>Pattern &quot;Challenge&quot; example. Source: Crossy Road - The challenge is to jump on 12 lilypads to receive a coin reward. (Color highlighted)</td>
</tr>
<tr>
<td>16</td>
<td>Pattern &quot;Character selection&quot; example. Source: Temple Run 2 - The characters can be selected in the store. Some are only available through a real money purchase.</td>
</tr>
<tr>
<td>17</td>
<td>Pattern &quot;Mini tutorial&quot; example. Source: The Line Zen - A quick introduction for the player when initiating the game.</td>
</tr>
<tr>
<td>18</td>
<td>Pattern &quot;Leaderboard&quot; example. Source: Doodle Jump - The leaderboard shows all highscores. Doodle Jump uses its own leaderboard instead of the Google Leaderboard Api.</td>
</tr>
<tr>
<td>19</td>
<td>Pattern &quot;Store&quot; example. Source: Temple Run 2 - The store offers power ups, coins and characters that can be purchased through real money or virtual diamonds from the game.</td>
</tr>
<tr>
<td>20</td>
<td>Brick Hit gameplay screenshots</td>
</tr>
</tbody>
</table>
Question 1: "For how long have you been developing mobile games?"

"Please check all patterns that you think are a must to create an endless mobile game. (meaning if you left them out, it wouldn’t be an endless game)"

Blue highlighted = Mandatory ("Essential"), Not highlighted - Optional ("Not a necessity")

"Game design patterns can be a good way for a developer team to share common terminology and know specifically what component the other is talking about."

"Game design patterns can support developers in the creation phase of an endless mobile game."

Developer questionnaire: "The prototype "Brick Hit" represents the game design patterns for endless mobile games well."

Player questionnaire: "The game Brick Hit fits to this definition of endless mobile games?"

"If you play mobile games, for how long have you been playing?"

"If you play mobile games, how often do you play?"

"If you play mobile games, how long is a single session usually?"

"The game would be better with levels and no endless mode."

List of Tables

1 Pattern description summary .................................................. 38
2 Pattern implementation table .................................................. 40
3 Pattern evaluation summary .................................................... 44
4 Pattern grading ................................................................. 52
5 Brick Hit game enjoyment summary: Number of votes from 25 participants. 55
1 Introduction

Over the last decade smartphones have become an essential device in everyday life and work. Its processing power and touchscreen have furthermore enabled a new type of gaming experience. The distribution of apps is eminently facilitated through the public availability of app stores and the new concept of free apps/games. In this research we specifically focus on game design patterns of endless mobile minigames. This type of game is relatively new and has spread over the last few years. Hence the term "endless mobile minigames" is not globally established and has been scarcely researched [23].

One of its main characteristics is a quick gameplay. One iteration usually takes no more than twenty seconds to a couple of minutes. The player attempts to score as many points as he can until a certain event happens which triggers the players defeat. The player usually does not require a tutorial or a handbook to be able to play. Thus endless mobile minigames seem very attractive to smartphone owners as they are short-lived and offer a way to socially interact with people from around the world [21].

Iconic endless games that have gained popularity are Temple Run, Doodle Jump or Flappy Bird. All of them record a download number of at least 10 million downloads. The app stores nowadays are filled with millions of apps. Some of those manage to reach the top surface of the rankings in the popularity/paid/free category while some end up with no downloads at all. This is no different when related to the endless mobile minigame genre.

Oh et al. [13] have studied the appealing characteristics of mobile games and claimed that high image quality, communication in terms of sharing scores and an attractive design play a vital role in the success rate. Similar results were stated by Park and Kim [17].

Our research aims at supporting the development of those games by creating unique pattern guidelines of endless mobile minigames.

1.1 Motivation

The market for mobile apps has been growing exponentially ever since Apple and Google both published their app stores (iPhone AppStore and Android Market, now Google Play Store) in 2008. The demand for mobile apps and games has steadily increased ever since, which leads many developers to this domain. Although the market is highly competitive and about one third of the apps in the app stores are games, many endless mobile minigames keep climbing to the top of the recent app store rankings, see screenshot in figure 1.

For us it is interesting to see how such a minimalist genre of mobile gaming can have such a huge impact. Many apps of this kind are deficiently created and lack good quality, since it is an emerging genre in the mobile industry. Game design patterns can be useful to game designers and developers. They provide a problem-solution approach [1], with which it is possible for different professions to exchange information using the same terminology and language. Especially in the initial creation phase of a game, patterns can
support determining the feasibility of a game by providing experienced driven information. Also previous research on the genre of endless games in combination with game design patterns is nonexistent and therefore requires a scientific approach to pave the way for game developers and respectively designers. Hence we believe that the mobile game industry may benefit from our study.

1.2 Goals and Research Question

In order to assist game developers creating an endless mobile minigame on smartphones, we attempt to create generalized game design patterns for this game category. Our goal is to create a common language to help game developers that are focusing on this genre. Consequently our research questions are:

- **Research Question 1** What game design patterns specific to a set of endless mobile minigames can be identified?

- **Research Question 2** How can game design patterns specific to endless mobile minigames be applied in a real-usage scenario?

To answer these questions we conduct a literature review, elaborate several case studies, surveys with game app developers and accordingly conclude game design patterns for endless mobile minigames.

Afterwards we implement a prototype mobile game using some of the inferred game design patterns to validate our results. See section 3 for an extended description of our research methodology.
1.3 Limitations and Delimitations

We limit our research to the literature review, case study and design and creation that we conduct. To validate our research we try to aggregate as much data from the chosen cases as we can. To the best of our knowledge there is no peer-reviewed literature specifically about endless games in relation with game design patterns, hence our studies aim to set an initial and basic foundation to start with. However we did find a number of peer-reviewed studies on mobile games in general which we considered when synthesizing the foundation of our research [13, 14, 20]. These studies focus on mobile game behavior, as well as general characteristics of mobile games. The aim of those research papers was specifically referring to all mobile games. A distinction of game genres of different types was absent. Our focus on endless mobile minigames sets our work apart from previous research, which have related to all kinds of genres. This study does not present a manual to develop a game software in a specific programming language. We focus on patterns and elementary characteristics of endless mobile minigames. Our time constraint reaches out to 4 months of consecutive work.

1.4 Outline

This research is structured in seven sections. We start with a literature review in chapter 2 to depict the current state of the art. Section 3 describes the methods that we have used and how we evaluated our outcome. Chapter 4 presents the results of our case study, followed by chapter 5, which explains the prototype that we have implemented. Chapter 6 contains the evaluation results. The thesis finishes with a conclusion and a prospect in chapter 7.
2 Literature Review

This section covers our literature review regarding mobile games in general, the consumers' behavior regarding mobile games and game design patterns for mobile games. The purpose is to extract viable information that can aid us in designing and developing an endless mobile minigame. More importantly, we identify the content of research papers and pick out their strengths and weaknesses to determine a gap. We critically evaluate the studies and motivate our research question by investigating what information is available and what is still absent in current literature. We begin with background information regarding mobile games in general, then take a more specific look into the endless game genre. To investigate more in player interaction, which game design patterns are focused on, we discuss metrics and consumer behavior related to mobile games. In the end we present what is known about game design patterns and what their strengths are, to be able to adapt them to the endless game genre.

2.1 Mobile Games

Typical video games have been around for a couple of decades. Ever since they have established themselves in the entertainment industry. Millions of mobile games have been uploaded to the mobile app market in the last few years along with the rise of smartphones. The number of mobile gamers is estimated between 400 million people and the mobile market itself had an approximate value of 5.4 billion dollars already in 2008 [18]; the period of time when the app stores from Apple and Google have just been released.

Mobile games differ from traditional video games in a way that a new kind of experience has emerged. One reason is the smartphones touchscreen which allows swiping gestures and interactions instead of plain buttons as in traditional video games [2]. Another reason is that one is not required to commit a vast amount of time on one mobile game [21]. Mobile games can be played while standing outside waiting for a bus, sitting inside of a waiting room at the doctor or anywhere else one could imagine. Therefore the lifetime of a mobile game is usually short-lived in comparison to previously known video games. The benefit of playing video games is often times indisputable, although they have proven to enhance creativity and stress release from the real world [6].

From a game developers point of view, things have changed as well. The opportunity of becoming a game developer has increased [9]. The only requirement is a computer, a smartphone and programming knowledge. Endless mobile minigames is a subset category of mobile games which some small businesses and developers have focused on. They offer an easy way of monetization and are fast to create [11]. Monetization can either be done by selling the app in an app store for a small amount of money or in-app advertisement, which for example Google AdMob offers expenseless.

Previous research has only provided a slight amount of information when it comes to endless mobile games. Therefore we have adjusted our research questions to fill this gap, which we believe is important in the current mobile game domain.
2.2 Endless Game Genre

An endless game is a concept in which the player is theoretically able to score points infinitely. The objective is to achieve as many points as possible by repeatedly interact against the game over trigger. One example is the game "Temple Run", the player plays until his character collides with an obstacle\(^1\). In this case the game over-trigger is the life bar reaching zero.

Endless games can be implemented in many different ways: endless runner, endless flyer, endless avoidance and basically every kind of interaction that can be repeated infinitely [15]. To create more variety in the game, it is common to use a random algorithm to generate level obstacles [15] which the user has to overcome (see section 2.3 for more information). The crucial feature here is that the difficulty of the game progressively is increased in a short period of time. The smartphones touchscreen has simplified the controls for mobile games [2]. In fact many endless games cope with only one single tap, which controls the entire gameplay, such as the game Drill Up (see section 4.1.1).

2.3 Metrics

Medeiros et al. [15] have elaborated a study about balancing a level in an endless runner game. Its main characteristic is that the game character runs in one direction and attempts to collect coins or/and avoid obstacles in order to reach the furthest distance possible.

To assure maximum enjoyment and adequate rewards while playing, they used metrics to identify what characterizes a balanced level [15]. An example for a barely balanced level is one where the user does not enjoy acquiring rewards or overcoming the obstacles of that level. However Medeiros et al. [15] have solely investigated in an endless runners level balance, which is only a segment of endless games. Hence we plan to further advance the theory behind this genre and give a more complete insight through our game design patterns.

Other metrics involve the players enjoyment while playing a mobile game. To determine decisive factors, Sweetser et al. [22] have elaborated a study, in which they investigate the players habits. They have defined eight elements which are viable: concentration, challenge, skills, control, clear goals, feedback, immersion and social interaction. In the following you find a brief description of each metric.

**Concentration** The player should be able to focus on the game without being distracted.

**Challenge** The game should be adequately challenging and progressively increasing.

**Skills** The game should require an amount of skills, that one can develop through playing.

\(^1\)Temple Run: https://play.google.com/store/apps/details?id=com.imangi.templerun
Control The controls of the game are supposed to be intuitive. A manual of the controls should only be necessary when having complex controls.

Clear goals The goal of the gameplay should be clearly noticeable and leave no uncertainties.

Feedback The player should be rewarded according to his achievements in the game.

Immersion The player is supposed to experience the game and get less aware of his surroundings. It should emotionally involve the player.

Social Interaction The support of sharing the highscore with others through social APIs increases competition throughout players.

These factors [22] are important to consider when creating a game in general. Although not related to endless mobile games in specific, the aforementioned metrics provide insights that might be useful when creating the patterns and the prototype.

2.4 Consumer Behaviour

To create a successful game, it is necessary to understand its players. Many remarks related to the gameplay do not show up until the game has been tested in the market. To figure out who exactly likes to play mobile games and under which circumstances, Seok and DaCosta [20] have elaborated a study about video game behavior in which they have analyzed the relationship between personality and mobile gameplay habits.

For this purpose they have conducted a questionnaire with 2600 participants from Seoul, South Korea. 1409 participants turned out to be active mobile game players as they played every day. 65% reported that the longitude of a daily game session was 30 minutes or less and about 69% stated to prefer free games over paid games [20].

When asking the participants what mobile game they like the best, they mentioned "Temple Run" (created in 2011 by Imangi Studios) on ranking number one. Temple Run is an endless runner game exclusively for mobile devices and has reached over 100 million downloads. It is also placed in the top 50 ranking of the apps that have been downloaded the most. This complies with the statement that a coherent and attractive design supports the success factor of mobile games positively [17]. The demand for short-lived games is high and seems to steadily increase. Hence our research question focuses on popular endless mobile games to ensure that players are fond of the gameplay.

2.5 Game Design Patterns

The purpose of game design patterns is to share experience and knowledge amongst game developers and designers. They are used to implement the game design and also facilitate game analysis [14]. It is therefore advisable to identify the components that define and exemplify a game genre [8]. An approach to do so is extracting game design patterns from existing games and their concepts. Björk et al. [8, 1] have stated a
generalized way to describe the attributes of a pattern in games, which involves following 5 attributes:

**Name** A unique, specific and idiomatic name that summarizes the pattern.

**Description** A precise and descriptive explanatory of the pattern. It can include a statement about the game that implements the pattern and mention examples.

**Consequences** This section describes what disadvantages the application of this pattern might cause and what other events can be expected. Cost and benefit are considered and alternatives discussed.

**Using the Pattern** This attribute is meant to help a game designer/developer to take decisions that are common when considering this pattern.

**Relations** Related patterns are discussed. These can be inherited patterns or subpatterns that are directly connected to this pattern.

This approach can also be applied to the mobile context and offers a facile method to depict game design patterns in mobile games. Naturally we are required to modify the proposed concepts to adapt to our context, as this is only a generalized pattern. So far, literature has not provided any evidence on doing so in the endless mobile minigame domain. This is where we are trying to expand knowledge by adding game design patterns for endless mobile games.

### 2.6 Gap analysis

The app stores already contain a vast amount of endless mobile minigames. Nevertheless there is not much information on how to design such games and what characteristics they comprise. We have ascertained that game design patterns do exist in current literature. They apply to general games by all means, however they only comply with our needs for our emerging genre to some extent. The endless mobile genre is rarely represented and only conceals a small segment. Therefore dedicated developers do not have a structured procedure and guideline to follow in order to produce an uniform/coherent endless mobile minigame. This gives us the opportunity to tackle this matter in our study and add on additional knowledge concerning endless mobile games in relation to game design patterns.
3 Research Methodology

This chapter provides an overview about the used methodology to gather and interpret data. The table below depicts the two research questions from section 1.2 and the assigned method for each question.

<table>
<thead>
<tr>
<th>RQ</th>
<th>Case Study</th>
<th>Design and Creation</th>
<th>Survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RQ2</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

In this section we describe how we conduct the case study, create a prototype and validate the results with a survey. This section closes up with a discussion about threats to validity in section 3.5, in which we critically address concerns that might affect the validity of our results. We compare each of our chosen methods to others, which we have not taken into consideration with a brief argumentation.

3.1 Case Study

A case study is utilized to provide context to a certain event, process or activity [5]. In our study it refers to mobile games which we analyze in order to reveal characterization elements of endless mobile minigames. We have chosen to conduct a case study since its primary advantage is that we can collect data and relate it to multiple specific cases [16]. The case study is from an exploratory and descriptive kind. Our case study investigates what game design patterns can be observed. Therefore we require multiple cases that fit our context. Each case represents one endless mobile game, which is thoroughly elaborated. Hence we believe that other research methods are less adaptive to our study.

An alternative approach would be the conduction of systematic interviews or focus groups with dedicated developers. These developers should have a focus on mobile gaming and experience with endless mobile games. The reason we have not followed this approach was the rare availability of those developers and our lack of contacts with persons from this field. Therefore a case study with systematic observation of the patterns is the most effective and beneficial method respective to our time frame.

3.1.1 Inclusion Criteria

We examined five endless mobile minigames from the current Google Play Store. Before we decided to select a specific game, we have defined what characteristics it should have, in order for us to consider it in our research. Therefore we arranged following inclusion criteria:

Infinite scoring A viable attribute of endless mobile games is that the player is able to score points infinitely.
Short gameplay iteration  Although the game is theoretically infinite, the gameplay should incrementally become more difficult and challenging the more points are scored. We chose cases in which one game takes less than five minutes.

Minimum downloads  We chose games with at least 100,000 downloads in the current Google Play Store. We ranked these as success factor of a game as well as positive feedback from the players, see next item.

Positive reviews  We chose games with at least a four-star rating, on a one to five star scale. We assumed that games with less stars are not suitable as their "fun factor" is not as high.

Release year  We chose games developed in 2014 or 2015 to assure their compatibility to the latest smartphone firmwares and technology.

These criteria items helped us finding representative mobile games that comply with the endless genre. To find suitable games, we browsed the app categories of the Google Play Store and selected adequate examples with respect to our inclusion criteria. A description of each case can be found in the case study section 4.

3.1.2 Data Analysis

From the cases we obtained commonalities, which characterize an endless mobile game. This acts as our foundation on creating a structured enumeration of patterns for developers and an example game which embraces all conducted game design patterns. In order to achieve this we thoroughly played the games one by one and transcribed all observations. Finally, we compared the results of each game and assembled a list of common characteristics. Every case is systematically observed in following order:

1 - Menu  We examined the menu regarding its options. Some features might reveal an incentive for players to continue playing the game.
2 - Gameplay  We observed the game mechanics itself by playing.
3 - Game Over  We observed the game over screen to detect other possible incitements.

3.1.3 Presentation of the Results

The results is a collection of game design patterns that have been assembled through the case studies. Each pattern is described with the following set of attributes and descriptions. This facilitates the comparison of amongst patterns. The following 9 attributes are depicted for each pattern.

Pattern Name  Title of the pattern.
Requirement  Describes under what circumstances this pattern can be applied.
Description A detailed definition of the pattern describing the problem and how to solve it.

Consequence What is the consequence of the pattern being used and what trade-offs and respectively benefits it implies.

Duration Describes when the pattern appears and for how long.

Example An example is depicted to represent a practical implementation of the pattern.

Emergence Defines in how many of the case study examples did this pattern appear.

Relevance A statement about the patterns necessity in endless mobile games. Is it mandatory ("Essential") to be implemented to be categorized as an endless mobile game or is it optional ("Not a necessity")? Note that this classification is a result of the questionnaire, not the case study. However we have included this attribute to give a more complete view, when developers read the pattern descriptions.

Relatedness to other patterns How it interacts with other patterns.

These attributes have partly been inherited from Davidsson et al. and Björk et al. [14, 1]. The attributes "Name, Description, Consequence and Relatedness" have been adopted. The "Requirement" and "Duration" attributes are originally depicted in the "Description" as well, but to achieve a better distinction between the content, we have separated them into individual attributes. The "Example" attribute has been replaced with the original "Using the pattern" attribute from Björk et al. [1], which now describes a concrete implementation of this pattern. The attributes "Emergence" and "Relevance" describe in how many cases this pattern has appeared from the case study and if the pattern is mandatory to implement when designing an endless mobile minigame.

3.2 Design and Creation

After we have completed the case study, we used the generated game design patterns to implement an endless mobile game. We followed a design science approach called Design and Creation. This method is used in the computer science context in order to create an IT product, also called "artefact" [12]. The reason we have decided to conduct this approach is that we wanted to implement a prototype which complies with our research problem for a practical purpose to answer RQ2. The result of this method is an "instantation" type of product, which represents a working artefact. Thus we have discarded other methods since we believe Design and Creation is the most suitable approach.

3.2.1 Design

We used the game design patterns which we have accomplished from the case studies to design our game prototype. This step heavily relies on our previously examined research and is therefore an important step to validate the applicability of the patterns.
3.2.2 Implementation

We worked with the game engine Unity 5 which uses C# scripts to implement functionality. The reason we selected Unity 5 is that it offers many game related functionality. The popular endless runner "Temple Run" was developed with Unity, which is a 3D game. Another example is a 2D game called "Bad Piggies" by Rovio which also made use of the aforementioned game engine. The outcome of this step is a working app which is executable on Android smartphones above OS version 4.1 (Jelly Bean). Another game engine that we have considered is called "Buildbox". It focuses specifically on endless games, but is also limited to that genre. The reason we have not selected Buildbox as our game developing tool, is that it does not require programming, which narrows down flexibility when creating the gameplay. We could confirm this after installing and testing the Buildbox 30 day trial, which is equivalent to the full version.

3.3 Survey

To assure that the created game design patterns are useful and applicable to endless mobile minigames, we conducted a survey and determined if the created patterns used on the prototype have been successfully implemented.

A survey is a method which is used to collect similar typed data from a group of people. It should be executed in a standardized and systematic way to ensure that we can conclude our results from [12]. The reason we used a survey is the structured way of acquiring user specific input/feedback. The participants should be regular smartphone users and optimally play mobile games casually. The result is used to determine the games "fun factor". A secondary survey involves developers, that have experience in the mobile game field. From the results we hope to ascertain whether the new game design patterns are useful when developing an endless mobile game and if the prototype is pertinent to its genre.

3.4 Evaluation

Since we are following a design science approach (see section 3.2), we need to evaluate our built IT artefact. Oates [12] has depicted three types of evaluation approaches: proof of concept, proof by demonstration and real-usage evaluation. We chose the proof by demonstration approach to assure that our created concept/results can be successfully implemented as a working game app. The survey mentioned in the sub-chapter before, therefore helps us to evaluate both game design patterns and the applicability of those in the prototype. Figure 2 gives an overview of our evaluation plan with respect to our research questions from section 1.2.

3.5 Threats to Validity

Since there is a vast amount of endless mobile minigames, which we are not able to examine entirely, our created game design patterns are limited to the ones that are
chosen for the case study analysis. The validity of the game design patterns therefore depends on our choice of games. To generalize the results, it is advisable to pick several examples. Palena et al. [16] question the reliability from generalizing from only a few case studies and assuming that they are representative. Therefore we attempt to examine as many cases as possible in our time frame. Also many researchers still consider case studies unscientific and less rigorous than surveys, since qualitative research is viewed not systematic enough [16]. Hence we define a structured specification when performing the case study.
4 Game Design Patterns

To answer research question 1 and identify game design patterns, we systematically observe five games. Therefore this chapter shortly depicts the five cases that we have selected for the case study. Each description introduces the general gameplay and some background information of the respective game. The inclusion criteria, that we have considered in the selection, can be found in section 3.1.1. All of the games are currently (March 2016) available in the official Google Play Store and can be downloaded on Android devices. The observation structure of the cases is described in section 3. The evaluation of the patterns is depicted in section 6.1.

4.1 The five cases

4.1.1 Drill Up

"Drill Up" was developed by a mobile game company called Ketchapp\textsuperscript{1}. It has reached about 250,000 downloads after 4 months after its publication in October 2015. The gameplay involves the main character, which has to elevate his height by jumping onto spinning disc platforms, see figure 3. The character jumps straight forward to the direction of its head as soon as the player taps the touchscreen once. For each disc, one point is added to the total score. The game progressively gets harder by fastening the gyration of the discs and lava that is approaching the main character from the bottom of the screen. The game over condition is only reached when either a disc is missed or the lava has touched the main character.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{drillup.png}
\caption{Case 1 - Drill Up}
\end{figure}

\textsuperscript{1}Drill Up: https://play.google.com/store/apps/details?id=com.ketchapp.drillup
4.1.2 Doodle Jump

"Doodle Jump" was created by Lima Sky in 2009\(^2\). Until now it has reached a total of 15 million downloads. The gameplay involves the main character which repeatedly jumps from platform to platform to reach the highest height possible, see figure 4. The score increases the further the player gets and is equivalent to the height that the player has reached. The player steers the main character by tilting the smartphone left or right. The movement is detected by the phones tilt sensor. Additional obstacles/enemies are placed within the infinite level to aggravate the gameplay. Also the player can tap the screen to shoot a directional projectile at the enemies. The game over condition is reached when the main character either misses a platform and falls down below the screen, or when the main character collides with an enemy.

![Doodle Jump](image)

**Figure 4: Case 2 - Doodle Jump**

4.1.3 Temple Run 2

"Temple Run 2" was developed and published in 2013 by Imangi Studios\(^3\). It has reached more than 100 million downloads and is one of the most downloaded mobile games of all time. Temple Run 2 was created using the Unity3D engine. The score is equivalent to the distance that the player has reached. The gameplay involves a main character, which runs along a path, see figure 5. The path can split up into left/right or both directions. The player can swipe left/right/up/down to choose the path that the player is supposed to follow. If the players reaction is too slow, the player will lose. The further the main character gets, the more points will be reached.

---


4.1.4 The Line Zen

"The Line Zen" was developed by the mobile app company Ketchapp in 2015\(^4\). It has been downloaded at least 1 million times according to the official Google Play Store. The goal of the game is to manoeuvre a ball by swiping left/right to avoid red obstacles, see figure 6. The score is equivalent to the reached distance. The ball moves from bottom to top and is able to touch green obstacles without losing the game. The purpose is to reach the furthest distance as possible. The level is segregated to small areas, which repeat themselves randomly. The higher the score is, the more difficult these areas become.

4.1.5 Crossy Road

"Crossy Road" was published in the end of 2014 by an app company called Yodo1\(^5\). It has reached at least 50 million downloads in less than 14 months. The game is an advancement of an old arcade game called "Frogger". The purpose of the game is to walk past as many streets, rivers, tracks and grass as possible by swiping into a direction to move the protagonist, see figure 7. Each step forward will add one point to the players score. The game over condition is reached when the protagonist is run over by a car/train or jumps into the river without reaching a piece of timber. Also if the player dwells in one position for too long, a raptorial bird will appear and snap the character out of the screen, which ends the game. The further one gets, the more points will be earned.

\(^5\)Crossy Road: https://play.google.com/store/apps/details?id=com.yodo1.crossyroad
Figure 6: Case 4 - The Line Zen

Figure 7: Case 5 - Crossy Road
4.2 Results

The problem that we are trying to solve is the lack of game design patterns specific to endless mobile games. In this section we present our results that we have concluded by observing the aforementioned 5 cases. Each pattern includes the description of the 9 attributes that we have summarized in section 3.1.3 - Pattern name, requirement, description, consequence, duration, example, emergence, relevance, relatedness to other patterns. We primarily aim at answering research question 1:

Research Question 1 What game design patterns specific to a set of endless mobile minigames can be identified?

The patterns are evaluated by survey in section 6.1.

4.2.1 Game design patterns for endless mobile minigames

Pattern 1

Pattern name Endless mode

Requirement Game does not have a "winning" state, in which a certain level is completed or all objectives are met, for example the levels in Super Mario. The game continues endlessly in a single level, but one "play until game over" remains short-lived due to its progressive difficulty.

Description This pattern leads the developer to choose a game structure which allows infinite playtime. The player is able to score an endless amount of points until he has reached the game over condition. One game can only technically last infinitely.

Consequence As the concept of "completing a level" is taken away with this pattern, the single level should be designed in a diversified way. Hence developers are advised to design a gameplay, in which the player is constantly challenged with new obstacles or level pieces. Given that one single game is held shortly, it is important to consider that a quick adaptiveness should be possible for the player.

Duration The average playtime of the cases we have observed was about 50 seconds, with minimum values from 20 to a maximum of 300 seconds. An outlier is the game Temple Run 2, which has an average playtime of about 100 seconds. All other cases fluctuated around the aforementioned average play time.\(^1\)

Example 1 The Line Zen: The circle ball keeps moving forward until it has collided with an obstacle. The game will therefore continue infinitely until this event has occurred, see figure 8. New level areas are being added constantly.

\(^1\)Note that this attribute is subjective to the authors skill level. This is further discussed in the section "Threats to validity" 3.5.
Example 2 Crossy Road: The character will endlessly move forward as long as it is not run over by a vehicle or jumps into the water.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Essential.²

Relatedness to other patterns It is common to use the pattern "Quick progressive difficulty" when using the "Endless mode" pattern. Also "Random level generation" is a requirement when developing an endless game.

![Figure 8: Pattern "Endless mode" example. Source: The Line Zen - The game continues until the blue ball hits a red obstacle/wall.](image)

²Note that this attribute is inherited from the developer questionnaire results and not from the case study. As for a complete overview, we have included this classification of "Essential" (mandatory pattern) and "Not a necessity" (optional pattern) to each of the 15 patterns.
Pattern 2

**Pattern name** Quick progressive difficulty

**Requirement** A gameplay which allows manipulating the difficulty.

**Description** The further the player advances, the more difficult the game will become. The progressive rate has to be chosen carefully so it does not make the gameplay too difficult or too easy. An alternative would be a constant level of difficulty. This approach does not apply with the endless mobile games that we have considered in our study. The gameplay can be aggravated after a certain amount of distance/points/time. Difficulty is expressed by more obstacles, faster gameplay pace or more spawning enemies.

**Consequence** The player has to adapt quickly to changing circumstances, which are usually difficult objects to avoid or simply the constantly increased gameplay speed. As the player needs a higher focus on the game, the performance of the game should be smooth so that lags do not hinder the control. Developers should highly pay attention to balancing the difficulty to the right extent to assure enjoyment and avoid frustration. Increasing the difficulty just the right amount will challenge the player and maintain a motivation to continue playing.

Another important aspect is the difficulty limit. The hardest state that the game is able to reach, should be playable, but demand a lot of focus and precision. If that limit is too easy, the player might get stuck into the game for too long, which is why this should be thoroughly tested by players with different skill levels to determine an appropriate difficulty.

**Duration** The period of time that it takes for the game to get to a difficulty level, in which the player is likely to lose, varies in a range from 30 seconds to 100 seconds (nevertheless depends on players skill level). Hence the average time would be at about one minute. From there, the games usually become harder even faster. The more extreme cases are Drill Up, Temple Run 2 and The Line Zen. The cases Doodle Jump and Crossy Road do get progressively harder, but stay in a moderate state.

**Example 1** Temple Run 2: The game starts slowly when initiating the game. The pace of the game increases, which forces the player to react quickly in order to score a further distance. Obstacles are emerging quickly and the player has to focus to be able to avoid them.

**Example 2** Doodle Jump: As the player advances, more monsters appear and less platforms are being spawned, which the character needs to jump on to. This makes it more difficult for the player to precisely land on one of the platforms.

**Emergence** 5/5 games from the case study have implemented this pattern.
Relevance  Essential.

Relatedness to other patterns  "Quick progressive difficulty" is a common pattern when developing endless games, therefore it is connected to the pattern "Endless mode". "Random level generation", "Obstacle" and "Supporting object" are used to set the difficulty balance.

![Pattern "Quick progressive difficulty" example](image.png)

Figure 9: Pattern "Quick progressive difficulty" example. Source: Doodle Jump - The game increases its difficulty after time by adding more monsters and less platforms to jump on.

---

Pattern 3

Pattern name  Random level generation

Requirement  An algorithm to generate the level or environment that the player is situated in.

Description  Random level generation is the process of creating a level, which is assembled by a random algorithm. Predefined level pieces can be used to put in a random order. It increases variety for the players to keep the game challenging. When using this pattern, the appending of the level pieces should be carefully elaborated to assure seamless levels and avoid unpredictable situations.
**Consequence** The player has to adapt quickly to the changing environment. The gameplays difficulty can vary, depending on the level generator. A disadvantage is that the developer has to carefully select the sequentially added level pieces so that no "impossible situation" occurs. An impossible situation is when the player has no option to score more points because the level is assembled incorrectly or imprecisely. An alternative would be to create a level with no random function, which none of the examined games have done, as it might result in a monotonous gameplay.

**Duration** As long as the player has not lost the game, the level will continue to be generated randomly.

**Example 1** Crossy Road: The path that the character can walk on is either wood, street or grass. Whilst advancing these elements are being spawned randomly along with the obstacles, which cross the screen to interfere with the character.

**Example 2** Drill Up: The spinning circles, which the character uses to hang on to, are being spawned randomly in size and position. They all have a minimum distance though to assure that no circles are colliding when being instantiated randomly.

**Example 3** Temple Run 2: The level blocks are set up randomly so the player experiences a diverse amount of combinations in taking different paths. The level blocks of Temple Run 2 are ground, water, tracks and gaps, see figure 10.

![Figure 10: Pattern "Random level generation" example. Source: Temple Run 2 - The last piece of the level is always randomly generated (land, gap, tracks, water) the further the player is advancing.](image-url)
Emergence 5/5 games from the case study have implemented this pattern.

Relevance Essential.

Relatedness to other patterns The random level generation is connected to the patterns "Endless mode", "Obstacle", "Supporting object" and "Quick progressive difficulty". Level blocks can be categorized in distinct difficulty categories which are then added randomly according to the current difficulty level. Meaning the further the player gets the more difficult the level pieces come.

Pattern 4

Pattern name Protagonist

Requirement A gameplay which needs a main character or physical object to interact with the level.

Description The protagonist is the focus of the game and under direct control of the player. Using different controls and timing, the protagonist is being guided. The visuals of the protagonist can be changed often times by selecting another character in the store (pattern "Character selection").

Consequence The player focuses mainly on the protagonist to progress further in the game. The player therefore needs to become familiar with the protagonists movements and behavior according to the touch input. There should be a clear visual distinction between the protagonist and other appearing characters in the game to avoid ambiguity.

Duration Throughout the entire gameplay.

Example 1 Temple Run 2: The protagonist is a human, which is running through the level, see figure 11. It interacts with the environment by colliding with objects and reacting to user interactions.

Example 2 The Line Zen: The protagonist is a ball, which acts as the main "character", see figure 11. It can collide with obstacles in the game and is supposed to stay in the bright area of the level. The color of the protagonist differentiates the ball from other objects visibly.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Not a necessity.
Figure 11: Pattern "Protagonist" example. Source: The Line Zen (left) - The blue ball is the main actor in the game. Temple Run 2 (right) - A person running represents the main actor in the game.

Relatedness to other patterns This pattern is directly related to the patterns "Character selection", "Store", "Obstacle", "Challenge", "Power ups and specials", "Supporting object", "Currency" and "Simple control".

Pattern 5

Pattern name Score system

Requirement A steadily increasing numerical value based on an interaction with the game.

Description The score starts at 0 and increases depending on different actions in the game. Most common indicator for the score is the distance that a player has reached while playing. But also other kinds of metrics can be used to define the score increments, such as "number of obstacles avoided". The highscore is the most amount of points the player has reached and is stored locally and eventually in the leaderboard.

Consequence The player can compare the scores to friends or others, which can be a motivation to continue playing.

Duration Throughout the gameplay.
Example 1 Temple Run 2: The distance that is run by the player is equivalent to the score. The measurement is in meters.

Example 2 Crossy Road: The score increases for each step forward that the player takes. Remaining in one spot will stop the increment.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Essential.

Relatedness to other patterns The "Score system" pattern is related to "Leaderboard", as that is where the scores are stored.

Pattern 6

Pattern name Obstacle

Requirement A gameplay which allows physical intersection between an obstacle and the player is a requirement for this pattern.

Description An obstacle is any kind of impediment, that the player is encountered with. The player usually needs to avoid the obstacle in order to continue the game. It is usually static, meaning it does not move, but there are also examples of dynamic obstacles. Static impediments can be walls, rocks, trees or any other kind of physical object. Dynamic obstacles can be moved by itself or gravity, such as platforms and falling objects. Examples for both static and dynamic objects are The Line Zen, Temple Run 2 and Crossy Road, in which both types of obstacles are attendant.

Consequence The player is required to overcome the obstacles by getting to know the gameplay physics. The player will lose the game when his movements are imprecise, which leads to intersecting with an obstacle. Developers should carefully choose the amount of obstacles that show up and also consider how difficult they are to avoid for the player. Not doing so might result in imbalanced level difficulty.

Duration An obstacle in endless mobile games is usually short-lived. After it has been overcome, the obstacle vanishes. Dynamic obstacles can move across the screen to aggravate the gameplay.

Example 1 The Line Zen: If the character collides with a red obstacle, the game will end. These obstacles can be of any size or geometric shape and move along the screen.
Example 2 Crossy Road: Trees are immovable obstacles, which do not cause a loss of the game, but they appear to block the path. Moving cars, trains are obstacles that need to be avoided in order to continue the game.

Emergence 4/5 games from the case study have implemented this pattern. Drill Up has no obstacles that hinder the player.

Relevance Essential.

Relatedness to other patterns This pattern is closely related to the "Protagonist" pattern, since both are required to intersect in some way (if protagonist is part of the game). Also the pattern "Random level generation" and "Quick progressive difficulty" often requires obstacles, which are endlessly spawned, when the level continues to build up. The "Supporting object" pattern is also connected, as these can be utilized to overcome obstacles. The pattern "Challenge" can involve obstacles as an event, which triggers an achievement with coin rewards.

Figure 12: Pattern "Obstacle" example. Source: Crossy Road - Obstacle trees (static) on the left and vehicles (dynamic) on the right are highlighted.
Pattern 7

**Pattern name**  Supporting object

**Requirement**  A situation in which the player requires a supporting object to continue the game, or a situation in which the player benefits from using a supporting object.

**Description**  A supporting object can either be necessary to continue the game, or as an addition to help the player advancing faster. Any object, such as platforms, branches, slides, can act as a supporting item.

**Consequence**  Depending on if the supporting object is mandatory to use or not, the player has to look out for them and decide whether to use them or not. Usually multiple objects are provided to involve the player in decision making and create multiple possible options. It is important to regulate how often they appear as the game might become too easy or too difficult depending on the instantiation rate.

**Duration**  A supporting object is short-lived and disappears after being used, by being scrolled off the screen usually.

**Example 1**  Doodle Jump, Drill Up: Platforms are being used, so that the protagonist of the game can jump onto those in order to reach a higher distance, see figure 13.

Figure 13: Pattern "Supporting object" example. Source: Drill Up - The spinning discs are used as platform for the player to advance in the game. (Color highlighted)
Example 2 The Line Zen: This game uses physical objects, which are marked with a different color than obstacles. A collision with these supporting objects will not lead to game over, but help the player go forward by pushing away obstacles.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Not a necessity.

Relatedness to other patterns This pattern is used with the pattern "Obstacles", "Protagonist", "Quick progressive difficulty" and "Random level generation". The protagonist interacts with the supporting objects to overcome obstacles in a more efficient way. Therefore the level needs to provide the supporting objects by spawning them continuously, depending on the game mechanics more or less often.

Pattern 8

Pattern name Simple control

Requirement This pattern has technical requirements, which in our cases were the smartphones touchscreen and tilt sensor. The touchscreen is used to detect swipes and taps, the tilt sensor to detect when the player physically tilts the phone.

Description The control is held simple, due to the short-liveness of this game genre. The controls that are the most common are taps, swipes and tilt (opposed to traditional controls using graphical buttons to press). With these types of input, the player is enabled to play the game by controlling the character. Drill Up uses a tap only approach, whereas Doodle Jump, Temple Run 2 and Crossy Road implemented a hybrid of all control types. The action that follows is always some type of movement, which the player influences through the controls.

Consequence The simple controls of endless mobile games differ from games with complex controls. The player is encouraged to focus on the timing and accuracy of the short movements in the game. Developers should thoroughly implement touch gestures for accuracy and also keep the reaction time short to assure smooth performance. The lack of intuitiveness of the controls might affect the gameplay negatively.

Duration As long as the game is being played, controls are required.

Example 1 Drill Up: One tap is the only movement that is possible in this game. It triggers the character to jump. Timing and precision are important to play the game.
Example 2 The Line Zen: By swiping left or right, the ball moves accordingly in order to slip through the gaps.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Essential.

Relatedness to other patterns "Simple control" is used next to the "Protagonist" pattern, if there is a character to navigate (if a protagonist is part of the game). Other than that simple control depends on the game mechanics and what interaction it causes.

Pattern 9

Pattern name Power ups and specials

Requirement A situation, in which the player benefits from using an enhancing power up item, meaning the player will reach a higher score or survive longer.

Description The effect of a power up usually retains for a short amount of time and allows the player to run faster, be invincible or collect coins faster. The player has to collect the item by controlling the character to collide with the item. They appear every now and then in the games Drill Up, Doodle Jump and Temple Run 2. In both Drill Up and Doodle Jump, the power ups are used to reach a higher distance by vaulting the character. In Temple Run 2 it is also possible to buy power ups and use them in the game.

Consequence The player has to get acquainted with the power ups to use them efficiently in the game. Also the player has to take the decision if it is worth going for the item, since obtaining an item is sometimes coupled with a more "dangerous" situation. Developers should carefully decide how often power ups are being instantiated and for how long they last, since this considerably affects game difficulty.

Duration Power ups prevail for an average of 4 seconds ranging from a minimum of 1 second to a maximum of 8 seconds.

Example 1 Temple Run 2: One possible power up is the coin magnet. It is used to draw all nearby coins automatically without collision with the coins.

Example 2 Drill Up: An elevator appears, which can be activated by collision. Once triggered, the elevator moves up and instantly rewards the player with multiple points.
Emergence 4/5 games from the case study have implemented this pattern. The Line Zen does not implement power ups or specials.

Relevance Not a necessity.

Relatedness to other patterns This pattern is closely related to the "Protagonist", which the power up is meant for using. "Challenges" can involve the usage of power ups to trigger a challenge achievement.

Pattern 10

Pattern name Currency

Requirement This pattern requires a store to be able to exchange the currency objects with items or new characters.

Description The currency is usually represented in the forms of coins, diamonds or any other type of collectible object. The currency shall be used to purchase new items in the store. They are placed randomly throughout the level and can be collected by collision with the character, that the player is controlling. The amount of currency objects that are spawned is a single one in Doodle Jump and Crossy Road. In Temple Run 2 and Drill Up, multiple coins are spawned simultaneously. It is also possible to collect more coins by watching video advertisement. After the player has watched the video advertisement, a reward in form of coins will be added.

Consequence The player can decide to collect the currency objects or not. It is not a requirement to further advance in the game. An outlier is Temple Run 2, in which the player is able to purchase power up items, which do enhance the outcome of the next gameplay iteration. It is crucial to decide how often a currency is being instantiated in the game and how quickly one can obtain enough to purchase something in the store. Making them appear less often might take away motivation to continue as it seems tedious to gather. Making them appear too often might make the currency seem "worthless".

Duration The currency objects vanish once obtained. The duration until the player can purchase an item from the store varies, depending on how often coins/diamonds are spawned. In Temple Run 2, Crossy Road and Drill Up it takes about 8 game iterations until a new item or character can be acquired.

Example Crossy Road and Drill Up: Both games provide coins in the gameplay, see 14, as well as in form of presents, which the player gets, when returning back to the game after 5 hours in Crossy Road, and 25 minutes in Drill Up. The amount of coins varies from 20 to 60 coins in each game. This is an incentive for the player to return for the game in order to obtain these presents.
Emergence 4/5 games from the case study have implemented this pattern. The Line Zen does not have a currency system.

Relevance Not a necessity.

Relatedness to other patterns This pattern is closely related to the "Store", "Character selection", "Challenge" and "Protagonist" patterns. The character usually gathers the currency items in order to be solvent when purchasing in the store.

![Figure 14: Pattern "Currency" example. Source: Crossy Road - The coins are randomly placed throughout the level and are being added to the total number on the top right.](image)

**Pattern 11**

**Pattern name** Challenge

**Requirement** The gameplay should allow different types of actions/events that can occur.

**Description** A challenge represents a specific event, that can occur throughout a gameplay. The purpose of it is to keep the player motivated to play and obtain all challenges. In Doodle Jump, one of the challenges is to "score a total of 100.000 coins". It can also be a certain action, which completes an achievement, such as "Jump on 5 wagons in one game" in Temple Run 2. The developers have included a randomness factor to make it rare and desirable. It is also possible to implement challenges (or achievements) through the official Google Achievements API. Therefore the players challenges are directly connected to the official Google Play Games Account. For each obtained challenge, points will be added to that account in order to level up. Since the account can be connected to all games, which implement the official Achievements API, the player has a stimulus to complete all achievements.
**Consequence**  After accomplishing a challenge, the player can check them like a trophy list. The achievement of a challenge is rewarded with currency objects like coins and diamonds, which is an incentive to continue playing. Developers should create a decent amount of challenges, so that the player has some kind of sub goal to pursue apart from the main gameplay.

**Duration**  Once a challenge is completed, it can not be obtained again. An outlier is Crossy Road in which the challenges repeat themselves from time to time.

**Example**  Crossy Road: In this game the appearance of the challenges are implemented in a slightly different way. The objectives of the challenge only emerge now and then after a game is completed and shows what needs to be done in order to complete the challenge.

**Emergence**  4/5 games from the case study have implemented this pattern. The Line Zen does not implement challenges.

**Relevance**  Not a necessity.

**Relatedness to other patterns**  Challenges usually include actions involving the patterns "Protagonist", "Obstacles", "Currency" and "Power Ups and specials". All of these patterns can have certain events that trigger the achievement of a challenge.

![Crossy Road Challenge Example](image)

Figure 15: Pattern "Challenge" example. Source: Crossy Road - The challenge is to jump on 12 lilypads to receive a coin reward. (Color highlighted)
Pattern 12

**Pattern name**: Character selection

**Requirement**: The pattern "Character selection" requires a character, which the player is controlling, and some kind of currency to exchange it with.

**Description**: Characters can be purchased or unlocked in the store. Selecting a character will change its appearance while playing, but will not have an effect the gameplay itself. It is possible to randomly obtain a character or simply choosing the desired one. The amount of characters that can be selected varies from 5 to 60 in our cases, with an average of about 21 characters. Some characters can only be purchased with real money purchases.

**Consequence**: This pattern can be an incentive for players to keep playing in order to obtain all different characters. In Doodle Jump and Crossy Road, not only the protagonists visual appearance changes, but also the environment (level). Developers should choose appealing visual looks or iconic characters to enhance enjoyment and motivation when obtaining characters.

**Duration**: Once a character has been unlocked, the player can choose to select other characters at any time.

![Character selection example](image)

Figure 16: Pattern "Character selection" example. Source: Temple Run 2 - The characters can be selected in the store. Some are only available through a real money purchase.
Example 1 Temple Run 2: It is possible to unlock 8 characters. 6 can be purchased through the virtual currency that has been collected, the other 2 can be bought with real money (an equivalent of 10 Swedish crowns), see figure 16. The visuals of the character will change accordingly in the game.

Emergence 5/5 games from the case study have implemented this pattern.

Relevance Not a necessity.

Relatedness to other patterns "Character selection is used next to the "Protagonist", "Currency" and "Store" patterns.

---

Pattern 13

Pattern name Mini tutorial

Requirement A gameplay which requires explanation/introduction regarding its controls.

Description Due to the simplicity of endless mobile games, the introductory text for the player usually appears in the very beginning of the game. Using graphical and a few textual tips, the player is being taught how to play. The tutorials are not detailed, but concise.

Consequence The player learns how to play the game and controls. Developers should stick to a minimalist design with precise and graphical hints. Since it is a short-lived genre, there is no need for extensive introductions to the game. The introduction has been undertaken in only very few words in all cases and relied mostly on the graphical aspect. The objective of a game is not described and it is assumed that the player will find out. The introductions only contained information about the controls. Actions like avoiding enemies or jump over gaps are assumed to be learned by trial and error, as there is no tip in this matter.

Duration The tutorial is displayed and disappears after a tap on the screen.

Example 1 The Line Zen: When the game starts an introduction sentence saying "swipe left/right" and two arrays pointing left/right show up. The message vanishes after a couple of seconds, see figure 17.

Example 2 Crossy Road, Temple Run 2: Both games additionally display some hints while playing when the player happens to stumble over a specific situation, which requires a certain type of control.

Emergence 5/5 games from the case study have implemented this pattern.
**Relevance** Not a necessity.

**Relatedness to other patterns** The pattern "Simple control" is being explained by the tutorial, hence the connection.

![Pattern "Mini tutorial" example](image)

Figure 17: Pattern "Mini tutorial" example. Source: The Line Zen - A quick introduction for the player when initiating the game.

---

**Pattern 14**

**Pattern name** Leaderboard

**Requirement** A game with a score system which allows numerical comparison of the scores.

**Description** The leaderboard can be implemented in two different ways. One method is to have a local leaderboard, which is only accessible through the players own smartphone. Another is to globally provide a server, which the data can be polled from. A common way is to implement Google’s official leaderboard. An API is provided which is used to do all necessary actions as a developer, such as committing a new score, comparing scores and displaying all scores. To be able to use the official leaderboard, the player needs to sign in with a Gmail account.
Consequence  The player is encouraged to keep playing to get to the top of the leaderboard and compare the score to friends and others. The Google API offers a wide range of functionality, which is easy to implement. Hence developers are propelled to use the API as it is a complete and up to date module.

Duration  Always accessible in the menu.

Example 1  Drill Up: The official Google Leaderboard API is used to store global information about the players highscores. Players can compete and have access to other users highscores.

Example 2  Doodle Jump: Doodle Jump has a local and a global leaderboard, but does not use the official leaderboard provided by Google. The local leaderboard stores all scores reached by the smartphone owner, the global leaderboard is connected to a server, storing all highscores from Doodle Jump players, see figure 18.

Figure 18: Pattern "Leaderboard" example. Source: Doodle Jump - The leaderboard shows all highscores. Doodle Jump uses its own leaderboard instead of the Google Leaderboard API.

Emergence  5/5 games from the case study have implemented this pattern.

Relevance  Essential.

Relatedness to other patterns  This pattern is closely dependent on the "Score system" pattern, as the numerical value is used for the leaderboard.
Pattern 15

Pattern name Store

Requirement A set of items that can be purchased is required for a store. Also if the store offers real money deals, the player needs to log into a Gmail account in order to purchase.

Description The store provides a set of items, which can be purchased by the player in exchange for coins or real money. Different types of items that are obtainable are characters or virtual coins. In our cases, the store is mainly used to provide characters.

Consequence The player has an incentive to collect all items or buy items to enhance the gameplay. Stores offer a great way to maintain player retention and are therefore most common in endless games. Adding appealing visual graphics enhances enjoyment in purchasing items and keeps the player active in the game.

Duration Always accessible in the menu.

Figure 19: Pattern "Store" example. Source: Temple Run 2 - The store offers power ups, coins and characters that can be purchased through real money or virtual diamonds from the game.

Example 1 Temple Run 2: This game is an outlier case, in which it is possible to purchase different types of items, such as power ups, maps and abilities. In Temple Run 2 it is possible to purchase virtual coins with real money, with which the player can obtain more items.
Example 2  Crossy Road: The store offers solely characters, which can be either purchased through virtual currency in the game or real money.

Emergence  4/5 games from the case study have implemented this pattern.

Relevance  Not a necessity.

Relatedness to other patterns  This pattern is related to the patterns "Currency", "Protagonist", "Character selection" and "Power ups and specials", as these are items which can possibly be sold in the store.

4.3 Summary

To create game design patterns specific to endless games, we have observed 5 cases and concluded 15 patterns. To allow comparison of the patterns and provide an overview, we present all patterns briefly in table 1. The table shows, in which of the cases the patterns have been observed and provides descriptive keywords for each pattern to get a glance. Additionally it shows the "Relevance" attribute. Note that the relevance classification is a result of the questionnaire, not the case study. However we have included this attribute to give a more complete view, in case developers read the pattern descriptions.

The evaluation of the case study is depicted in chapter 6. The observation of the cases has revealed more patterns than the 15 that were mentioned, however they have been detected in less then 4 cases, which is the reason why we have not included them in our pattern collection. Thus the available patterns are present in at least 4 cases and mostly in all 5. As the implementation of each pattern differs from case to case, we have included examples in the description to provide a better understanding.
<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Cases</th>
<th>Keywords</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Endless mode</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Infinite gameplay, no winning state, one level</td>
<td>Essential</td>
</tr>
<tr>
<td>2</td>
<td>Quick progressive difficulty</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Elevation of difficulty, short-liveliness, difficulty limit</td>
<td>Essential</td>
</tr>
<tr>
<td>3</td>
<td>Random level generation</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Progressive addition of level pieces, difficulty of level pieces</td>
<td>Essential</td>
</tr>
<tr>
<td>4</td>
<td>Protagonist</td>
<td>DU,DJ,TR2,CR</td>
<td>Main character, interaction through physics</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>5</td>
<td>Score system</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Point system, highscore</td>
<td>Essential</td>
</tr>
<tr>
<td>6</td>
<td>Obstacle</td>
<td>DJ,TR2,LZ,CR</td>
<td>Impediment, avoidance, dynamic/static</td>
<td>Essential</td>
</tr>
<tr>
<td>7</td>
<td>Supporting object</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Helps player, dynamic/static, platform</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>8</td>
<td>Simple control</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>No complex controls, taps/swipes/tilt</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>9</td>
<td>Power ups and specials</td>
<td>DU,DJ,TR2,CR</td>
<td>Facilitates gameplay for short period</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>10</td>
<td>Currency</td>
<td>DU,DJ,TR2,CR</td>
<td>Collected while playing, usage in store, exchange with items</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>11</td>
<td>Challenge</td>
<td>DU,DJ,TR2,CR</td>
<td>Specific mission while playing, earn extra coins</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>12</td>
<td>Character selection</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Visuals of protagonist</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>13</td>
<td>Mini tutorial</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Instructions, graphical/textual</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>14</td>
<td>Leaderboard</td>
<td>DU,DJ,TR2,LZ,CR</td>
<td>Comparison of highscore, global/local</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>15</td>
<td>Store</td>
<td>DU,DJ,TR2,CR</td>
<td>Buy items/characters, real money/virtual currency</td>
<td>Not a necessity</td>
</tr>
</tbody>
</table>

A brief summary of each pattern. Abbreviations of the cases: Drill Up = DU, Doodle Jump = DJ, Temple Run 2 = TR2, The Line Zen = LZ, Crossy Road = CR
5 Prototype of an Endless Mobile Game

The prototype aims at answering research question 2:

\textbf{Research Question 2} How can game design patterns specific to endless mobile minigames be applied in a real-usage scenario?

This section describes the concrete implementation of an endless mobile game based on a set of patterns, which we have identified in the previous chapter. We depict the patterns that we have used to develop the prototype "Brick Hit".

The purpose of creating game design patterns is to enable developers to communicate about specific game details through common notation and terminology. To prove that those patterns are valid, we use them in a prototype of an endless mobile game and assure that developers can benefit from their presence. The implementation of the pattern is also important to validate the applicability of the patterns, which is a main attribute of game design patterns [1]. Also we try to find out if the mandatory patterns can support developers in creating and designing the game.

5.1 Implemented Game Patterns

This section briefly describes how we have approached the game design and development phase by utilizing 12 patterns. As game design patterns are supposed to help practitioners [1] by providing applicable information, we have attempted to to answer RQ2 by implementing a game ourselves. The required patterns have been identified through the questionnaire with mobile game developers, see in the evaluation section 6.1.1. We have asked the participants to state, which patterns they believed are mandatory when developing an endless mobile game and the result was our classification in either "Essential" or "Not a necessity".

We have used the patterns as a guideline and attempted to recreate the game according to the description of a pattern as closely as possible. The following patterns have been applied to develop Brick Hit, see table 2. A detailed description of each pattern can be found in section 4.2.1 of the prior chapter. All patterns marked as "Essential" are mandatory and therefore implemented in the prototype. However we have added more patterns aside from the mandatory ones to enhance the fun factor. The importance of the optional patterns is discussed in section 6.1. We attempted to use all 15 patterns in our prototype, but due to time constraints and gameplay type, we have left three aside - Protagonist, Character selection, Store (see table 2).
<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Essential</th>
<th>Not a necessity</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Endless mode</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>2</td>
<td>Quick progressive difficulty</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>3</td>
<td>Random level generation</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>4</td>
<td>Protagonist</td>
<td></td>
<td></td>
<td>Not implemented due to gameplay type</td>
</tr>
<tr>
<td>5</td>
<td>Score system</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>6</td>
<td>Obstacle</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>7</td>
<td>Supporting object</td>
<td></td>
<td>✓</td>
<td>Feasible</td>
</tr>
<tr>
<td>8</td>
<td>Simple control</td>
<td>✓</td>
<td></td>
<td>Core pattern</td>
</tr>
<tr>
<td>9</td>
<td>Power ups and specials</td>
<td></td>
<td>✓</td>
<td>Feasible</td>
</tr>
<tr>
<td>10</td>
<td>Currency</td>
<td></td>
<td>✓</td>
<td>Feasible</td>
</tr>
<tr>
<td>11</td>
<td>Challenge</td>
<td></td>
<td>✓</td>
<td>Feasible</td>
</tr>
<tr>
<td>12</td>
<td>Character selection</td>
<td></td>
<td></td>
<td>Not implemented due to time constraint</td>
</tr>
<tr>
<td>13</td>
<td>Mini tutorial</td>
<td></td>
<td>✓</td>
<td>Feasible</td>
</tr>
<tr>
<td>14</td>
<td>Leaderboard</td>
<td></td>
<td>✓</td>
<td>Core pattern</td>
</tr>
<tr>
<td>15</td>
<td>Store</td>
<td></td>
<td></td>
<td>Not implemented due to time constraint</td>
</tr>
</tbody>
</table>

All patterns with a checkmark in columns "Essential" or "Not a necessity" have been implemented. All patterns with no checkmarks have not been implemented.
5.2 Gameplay

The goal of Brick Hit is to break upcoming bricks by vaulting a ball at them through a swipe. The camera moves upwards and progressively becomes faster to create a more challenging gameplay. The bricks are positioned randomly and instantiated above the screen, which creates the feeling of an endless level, see figure 20. The difficulty of the level pieces increases the further the player gets. The score represents how far the player has gotten, meaning the faster the camera moves, the faster the amount of points will be added to the score value. The player loses the game once any colored brick collides with the lowest part of the screen, which means that the player missed to break one brick. White bricks are solid and cannot be destroyed. They represent static obstacles, whereas the colored bricks can be either static or dynamic. The ball is being controlled by swiping into the desired direction. The ball has a fixed speed to ensure a deterministic behavior for the player. It is possible to collect coin rings by completing challenges or hit them with a ball, as they are spread randomly within the game. Power ups can be picked up and allow the player to gain different kinds of balls that facilitate breaking the bricks.

![Figure 20: Brick Hit gameplay screenshots](image)

5.3 Implementation

This sub section gives a brief overview of the game from a technical point of view. The game was developed with a game engine called Unity (Version 5.3). More on Unity: [http://unity3d.com/unity](http://unity3d.com/unity)
able to export the created game onto many different platforms, such as iOS, Android, Windows, PS4, Xbox One, Playstation Mobile and many others. Unity facilitates game development by creating a intuitive program user interface. Nevertheless programming is eminently required in either C#, JavaScript or UnityScript (Unity's household language). Therefore Unity is perfectly adequate to create an endless mobile game of our needs. Each brick, ball or any other object in the game uses a script to control its behavior.

5.4 Summary

The problem that we have attempted to solve was finding out if the new game design patterns are applicable in a real-usage scenario and how the patterns support us thereby. To solve this problem we have implemented the prototype "Brick Hit", which we then evaluated by conducting a questionnaire. We asked players and developers to what extent they agree that the game has implemented the patterns well. The result has revealed that the prototype can be considered a functional endless mobile game, see in the evaluation sub section 6.2. We have designed the game in close relation with the patterns, which have supported us in determining which elements can be considered in our type of endless game.
6 Evaluation

After the game design patterns have been assembled and the prototype implemented, we conducted two surveys to evaluate our results - A developer questionnaire and a player questionnaire. This section contains an outline of the conducted survey and evaluation. This is used to determine the patterns applicability and players enjoyment regarding the prototype. We start with the game design pattern evaluation and end with the prototype evaluation in two separate sub-chapters.

6.1 Game Design Patterns Evaluation

To prove the applicability and usefulness of our game design patterns described in section 4, we conduct a questionnaire addressing mobile developers to confirm our results. The questionnaire (see Appendix B) contains the list of game design patterns, that we have examined, so the developer can determine each patterns viability. The questionnaire also includes questions regarding the prototype to analyze the developers opinion about its aptitude. To find mobile game developers, we have browsed the Google Play Store and contacted them via mail. In the end 450 mails have been sent to mobile game developers. The response rate was 6% (27 answered questionnaires). The amount of time the developers have been working on mobile games varied from less than 1 year to more than 4 years, see figure 21. Of those, 48,1% (13 votes) have stated that they have worked with game design patterns. 81,5% (22 votes) have stated that they have developed an endless mobile game, which helps us in determining the patterns validity better as they have experience in this genre. The questionnaire presented a definition of endless mobile games and showed some example games, before the developers answered this question.

Figure 21: Question 1: "For how long have you been developing mobile games?"
Table 3: Pattern evaluation summary

<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>NA</th>
<th>SD</th>
<th>D</th>
<th>N</th>
<th>Agree</th>
<th>Strongly agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Endless mode</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Quick progressive difficulty</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Random level generation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Protagonist</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Score system</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Obstacle</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Supporting object</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Simple control</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>14</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Power ups and specials</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>Currency</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>Challenge</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>Character selection</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Mini tutorial</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>Leaderboard</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>Store</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>12</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Statement: "This pattern characterizes an element of an endless mobile game"
NA = Not applicable, SD = Strongly disagree, D = Disagree, N = Neutral

6.1.1 Results

Our goal was to ascertain that the new patterns are applicable and useful. To identify the developers' opinion regarding each pattern we have provided a summarized version, containing a brief description and example images. Followed by a question asking the developers if "This pattern characterizes an element of endless mobile games". The answers were given on a Likert scale [7] expressing the participants' consent to the pattern - "I strongly disagree" to "I strongly agree". Additionally, we added the option "Not applicable" in case the participant does not want to answer the question. The items remain in a clear order and naturally it is only possible to mark one item. Our scale of measurement for the patterns is therefore ordinal and isomorphic [19], which allows us to compare the patterns among themselves. The result is the magnitude of a pattern's pertinence regarding endless games. This subchapter depicts the results, which is discussed in section 6.1.2.

Table 3 contains the outcome of the questionnaire involving 27 participants. The highlighted area represents the amount of developers agreeing to the pattern being an element of an endless game, meaning either "I agree" or "I strongly agree" has been marked. The following list summarizes the evaluation of the pattern in percentage:

- 1 Endless mode - 55.6% have strongly agreed and 37% agreed resulting in a total of 92.6% agreement, whereas 3.7% were neutral and 3.7% disagreed.
• 2 Quick progressive difficulty - 48,1% have strongly agreed and 33,3% agreed resulting in a total of 81,4% agreement, whereas 7,4% were neutral. No participant disagreed to any extent.

• 3 Random level generation - 48,1% have strongly agreed and 44,4% agreed resulting in a total of 92,5% agreement, whereas 7,4% were neutral. No participant disagreed to any extent.

• 4 Protagonist - 25,9% have strongly agreed and 51,9% agreed resulting in a total of 77,8% agreement, whereas 11,1% were neutral. 7,4% disagreed and 3,7% strongly disagreed, resulting in a total of 11,1% disagreement.

• 5 Score system - 44,4% have strongly agreed and 48,1% agreed resulting in a total of 92,5% agreement, whereas 7,4% were neutral. No participant disagreed to any extent.

• 6 Obstacle - 33,3% have strongly agreed and 44,4% agreed resulting in a total of 77,7% agreement, whereas 18,5% were neutral and 3,7% disagreed.

• 7 Supporting object - 11,1% have strongly agreed and 51,9% agreed resulting in a total of 63% agreement, whereas 25,9% were neutral. 7,4% disagreed and 3,7% strongly disagreed, resulting in a total of 11,1% disagreement.

• 8 Simple control - 25,9% have strongly agreed and 51,9% agreed resulting in a total of 77,8% agreement, whereas 14,8% were neutral. 3,7% disagreed and 3,7% strongly disagreed, resulting in a total of 7,4% disagreement.

• 9 Power ups and specials - 14,8% have strongly agreed and 40,7% agreed resulting in a total of 55,5% agreement, whereas 33,3% were neutral. 3,7% disagreed and 7,4% strongly disagreed, resulting in a total of 11,1% disagreement.

• 10 Currency - 18,5% have strongly agreed and 29,6% agreed resulting in a total of 48,1% agreement, whereas 33,3% were neutral. 11,1% disagreed and 7,4% strongly disagreed, resulting in a total of 18,5% disagreement.

• 11 Challenge - 29,6% have strongly agreed and 44,4% agreed resulting in a total of 74% agreement, whereas 22,5% were neutral. 3,7% strongly disagreed.

• 12 Character selection - 3,7% have strongly agreed and 37% agreed resulting in a total of 40,7% agreement, whereas 33,3% were neutral. 18,5% disagreed and 7,4% strongly disagreed, resulting in a total of 25,9% disagreement.

• 13 Mini tutorial - 22,2% have strongly agreed and 44,4% agreed resulting in a total of 66,6% agreement, whereas 29,6% were neutral and 3,7% disagreed.

• 14 Leaderboard - 37% have strongly agreed and 33,3% agreed resulting in a total of 70,3% agreement, whereas 29,6% were neutral. No participant disagreed to any extent.
15 Store - 18.5% have strongly agreed and 18.5% agreed resulting in a total of 37% agreement, whereas 44.4% were neutral. 7.4% disagreed and 7.4% strongly disagreed, resulting in a total of 14.8% disagreement. 3.7% marked "Not applicable".

To find out if a pattern should be considered mandatory ("Essential") when creating an endless mobile game, we included a question with the following outcome, see in figure 22. The developers then selected multiple patterns through check buttons. The purpose was to find out which patterns should be used when creating an initial prototype of an endless mobile game. Having this information could possibly help determining the feasibility and time constraint of the project. The core part of the game could be implemented with the least amount of patterns possible to create a working skeleton.

Figure 22: "Please check all patterns that you think are a must to create an endless mobile game. (meaning if you left them out, it wouldn’t be an endless game)"
Blue highlighted = Mandatory ("Essential"), Not highlighted - Optional ("Not a necessity")

8 patterns have received only a maximum of 40.7% of the developers votes (11 out of 27 votes), hence we did not consider them to be mandatory patterns, see figure 22. The 7 remaining patterns were considered to be essential by more than at least 63% of the participants (17 out of 27 votes), see blue highlighted patterns in figure 22.

- 1 Endless mode with 96.3%.
- 2 Quick progressive difficulty with 63%.
- 3 Random level generation with 77.8%.
- 5 Score system with 92.6%.
- 6 Obstacle with 70.4%.
• 8 Simple control with 74.1%.
• 14 Leaderboard with 63%.

Follow up questions
After we have conducted the questions for each pattern, we have additionally asked some further statements regarding game design patterns in general. As follow up questions we have therefore stated "Game design patterns can be a good way for a developer team to share common terminology and know specifically what component the other is talking about" and asked the participants to provide to what extent they agree. 40.7% strongly agreed and 48.1% agreed, while 11.1% stated a neutral opinion, see figure 23. None of the participants disagreed to any extent.

![Figure 23: "Game design patterns can be a good way for a developer team to share common terminology and know specifically what component the other is talking about.

We then provided a more specific statement and asked the participants to what extent they agree that game design patterns can support developers in the creation phase of an endless mobile game. 44.4% strongly agreed to that statement, 44.4% agreed and 11.1% were neutral, see figure 24. Again none of the participants disagreed to any extent.
Figure 24: "Game design patterns can support developers in the creation phase of an endless mobile game."

Prototype evaluation
To evaluate the prototype that we have designed and implemented using the aforementioned mandatory and optional patterns (section 5), we provided a demo video of Brick Hit in an early stage and included it in the questionnaire. The developers were then supposed to watch and then decide whether they believe the game fits the definition of the game design patterns of an endless mobile game well. 14.8% strongly agreed to the previously mentioned statement, 63% agreed, whereas 14.8% were neutral, see figure 25. 3.7% disagreed and 3.7% as well stated "Not applicable". None of the participants strongly disagreed.

Secondly we have given the participants of the other conducted questionnaire (the player questionnaire) a short description of an endless mobile game and some examples. Subsequently we have asked if they believe that our prototype Brick Hit fits this definition.
after they have played the game. 72% agreed to that statement (24% strongly agree, 48% agree), while 20% stayed neutral and 8% disagreed. None of the participants strongly disagree, see figure 26.

Figure 26: Player questionnaire: "The game Brick Hit fits to this definition of endless mobile games?"
Developer comments

Some developers have given us helpful advice and comments or left us an email response. In the following we approach some comments that we believe are worth mentioning. We have asked the participants to watch a video of our prototype and if they believe there are patterns missing in the prototype. We received the following answers:

- "Powerups"
- "Tutorial"
- "None it's fine"
- "Challenges"
- "Currency"
- "It doesn't have a protagonist. It's an endless game, but not a runner I think"

Since the video that we have provided was an early stage of the game, we did not manage to put it in all patterns yet. However we have implemented power ups, tutorials, challenges and currency afterwards. The last comment says the prototype does not use the protagonist pattern, which is correct. Nevertheless the participant states that "Brick Hit" is still an endless game, just not a runner type (like for example Temple Run 2). We also asked the developers to provide additional pattern suggestions and received 2 replies:

- "Straight movement path or curved path; three line movement (from the left side of the road to the right side) or free movement"
- "Daily rewards, to force user login into the game every day for receiving some coins or some big reward after 5-7 days of continuously logins."

The first suggestion is to add a pattern, which describes the movement direction of the game. In our prototype, the game moves upwards, but it differs from game to game even in our case study examples. Therefore a look into this might be considerable. The second suggestion is to add a pattern depicting a reward system, which is available in 2 of our case study examples. We have definitely considered this pattern, but discarded it due to the lack of emergence in our cases. Future research might incorporate more cases with reward systems and so we definitely think this advice was supportive.

And lastly we received an email response that has stated the following comment regarding the relevance of the patterns and their importance:

- "Note that if several of the other patterns were missing, it might be endless, but not very fun. For example, I could have an "endless" game where the player literally stares at a blank screen. Whoever stares at the screen the longest gets the highest score. Now this, in every right, could be considered a mobile endless game (similar to cookie clicker but without the explicit interaction or achievements). But I would hardly consider something like this to be very fun"
The developer claims that some patterns might not be crucially mandatory for an endless mobile game, but they do enhance the fun factor and can still have a significant impact on the game. We believe this aspect is highly important and definitely worth investigating. Our goal was to find out what patterns are mandatory, but we did not examine to what extent a missing pattern would influence the game. However, we hope to elaborate this matter in future research.

6.1.2 Discussion

Our goal was to investigate the developers opinion regarding the created patterns and the prototype. The results have provided some classification of the patterns, which we will now discuss. Table 4 summarizes each pattern describing its pertinence and relevance according to the developer evaluation results from the previous chapter. The attribute "pertinence" depicts how specific the pattern is to endless mobile games. To have a more understandable interpretation than the pure statistics from the survey, we categorize the results and provide a scale from 1 to 4. The higher the number, the more did developers agree to the respective pattern being an element of an endless mobile game. A pattern is classified with a pertinence level of 4 if it has received more than 80% agreement of the developers. The other classifications decrease accordingly in 20% intervals, meaning 60% to 80% agreement is graded with pertinence value of 3. Respectively 40% to 60% agreement results in a pertinence value of 2 and less than 40% agreement implies a pertinence value of 1, which is the lowest score to be reached. The "relevance" value is binary and can either be "Essential", meaning the pattern is mandatory, or "Not a necessity", meaning the pattern is not required for a game to be considered being an endless mobile game. The difference between both pertinence and relevance is that one describes how specific a pattern is to endless mobile games, and the other describes if the pattern is mandatory or not.

Pertinence

1 pattern - the "Store" - has been identified with a pertinence value of 1. Although only 14.8% disagreed in total, this pattern has only reached a total agreement percentage of 44.4%, which is the lowest value reached, see table 4. Many developers seemed unsure about the store being considered an element of an endless game, as the number of neutral responses was higher than the agreeing ones.

3 patterns with a pertinence value of 2 - "Power ups and specials, Currency, Character selection" - have been identified. These patterns have achieved a mediocre result, where the agreeing responses were slightly higher than the neutral ones. Not using these patterns when creating an endless game is hence not crucial, although still implemented by almost all of our example cases.

7 patterns have been identified with a pertinence value of 3 - "Protagonist, Obstacle, Supporting Object, Simple control, Challenge, Mini tutorial, Leaderboard". These patterns have achieved a high number of agreeing responses (at least 60%) and are therefore more specific to endless mobile games.

Lastly, 4 patterns have achieved a pertinence level of 4 - "Endless mode, Quick pro-
Table 4: Pattern grading

<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Pertinence</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Endless mode</td>
<td>4</td>
<td>Essential</td>
</tr>
<tr>
<td>2</td>
<td>Quick progressive difficulty</td>
<td>4</td>
<td>Essential</td>
</tr>
<tr>
<td>3</td>
<td>Random level generation</td>
<td>4</td>
<td>Essential</td>
</tr>
<tr>
<td>4</td>
<td>Protagonist</td>
<td>3</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>5</td>
<td>Score system</td>
<td>4</td>
<td>Essential</td>
</tr>
<tr>
<td>6</td>
<td>Obstacle</td>
<td>3</td>
<td>Essential</td>
</tr>
<tr>
<td>7</td>
<td>Supporting object</td>
<td>3</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>8</td>
<td>Simple control</td>
<td>3</td>
<td>Essential</td>
</tr>
<tr>
<td>9</td>
<td>Power ups and specials</td>
<td>2</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>10</td>
<td>Currency</td>
<td>2</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>11</td>
<td>Challenge</td>
<td>3</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>12</td>
<td>Character selection</td>
<td>2</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>13</td>
<td>Mini tutorial</td>
<td>3</td>
<td>Not a necessity</td>
</tr>
<tr>
<td>14</td>
<td>Leaderboard</td>
<td>3</td>
<td>Essential</td>
</tr>
<tr>
<td>15</td>
<td>Store</td>
<td>1</td>
<td>Not a necessity</td>
</tr>
</tbody>
</table>

Pertinence scale: 1-4 (1: not specific for endless games, 4: specific for endless games)
Relevance scale: binary ("Essential" or "Not a necessity")

Progressive difficulty, Random level generation, Score system. They have reached a high number of strongly agreeing responses and with very few to none disagreeing statements. The patterns with pertinence value 3 and 4 are specific to endless mobile games according to this questionnaire, whereas patterns with a pertinence value of 1 and 2 are debatable, as many participants were rather neutral than disagreeing. However 3 patterns received a pertinence level 3 but are still considered mandatory ("Essential") - Obstacle, Simple control and Leaderboard. This is due to our two different questions in the survey regarding the pertinence and relevance of each pattern.

Relevance
The cases have indicated that not all of the 15 patterns are crucially required to develop an endless mobile game. 7 patterns have been identified as the mandatory ("Essential") patterns, which are required for the game to be considered an endless mobile game. These patterns have reached the majority of the votes, see figure 22, and should therefore be the minimum amount of patterns, when creating the initial prototype. Nevertheless the remaining patterns were present in at least 4 out of 5 of our case study examples and are still viable elements when creating an endless game. However knowing which patterns are required can help determining the feasibility and setting a time constraint for the project. The follow up questions confirm the developers view about the usefulness of game design patterns, see figure 24, as the majority of the participants agreed to the statement that patterns can support creating an endless mobile game.
Prototype
Lastly we have asked developers if they believed that the prototype has implemented the game design patterns well. 77.7% of the participants agreed to that statement. Furthermore we have provided a description of an endless mobile game and asked them to confirm if they believed that Brick Hit was a game of that genre. 70% of the players agreed to that statement. Despite the fact that the majority of both questionnaire participants have positively evaluated our prototype, there is still a limitation. The developers have only judged the prototype by watching a video, which showed our game in a very initial phase. Whereas the players have played the finished game. Therefore the developers have a different perception of the game as they have not played it. Finally the outcome of this questionnaire item could have been different.

However the general gameplay of Brick Hit has not changed in the latest version and we therefore conclude (according to previous figures 25 and 26) that Brick Hit can be classified as an endless mobile game.

Contribution
Our contribution comprises a collection of game design patterns specific to endless mobile games with the purpose to help developers share experience and knowledge in the mobile gaming domain. To validate these patterns, we have asked 27 developers to evaluate each pattern and find out their general opinion about game design patterns. Our results were mostly positive with some indistinct outcomes for a few patterns, which require further study.

6.2 Game Enjoyment of Brick Hit Evaluation
A questionnaire was created (see Appendix A) and handed out to participants to test out the prototypes qualification as endless game. The questionnaire holds 16 questions determining the prototypes enjoyment regarding the game. Measuring the players experience reveals insights about the games fun factor, which approves or disapproves the games success. However we believe it is important to measure the enjoyment to ensure which patterns might have affected the fun factor negatively or positively. The answers are radio buttons expressing the players opinion on a Likert scale [7]. Fang et al. [4] have elaborated a study about measuring enjoyment of video games, in which they have claimed that a players affective, cognitive and behavioral reactions are influenced by several factors, such as prior knowledge, direct experience, personality traits and current well-being. Our questions incorporate this statement in order to be able to extract the most pertinent data out of the survey.
6.2.1 Results

We had 25 participants play our prototype Brick Hit and answer our player dedicated survey. 80% of the participants stated to be 21-30 years old, whereas 16% were 10-20 years old and 4% 31-40 years old. Of those, 40% have stated to be playing mobile games for more than 3 years, 20% for 3 years, 20% for 2 years and 12% for 1 year, see figure 27. No one has stated to have played less than 1 year.

![Figure 27](image1.png)

Figure 27: "If you play mobile games, for how long have you been playing?"

The participants have stated to play "every other day" (40%) and "weekly" (32%). Smaller groups have claimed to play "daily" (16%) and another group "monthly" (12%), see figure 28. The amount of time that they spent for a single gaming session was predominantly 15 minutes (44%) and 5 minutes (28%), see figure 29.

![Figure 28](image2.png)

Figure 28: "If you play mobile games, how often do you play?"

The next step was to find out the participants opinion regarding our created prototype in terms of attributes such as visuals, difficulty and experience but also its suitability as endless game. The results are represented in table 6.2.1, showing the amount of votes
for each statement. Generally speaking the majority with about 48% seem to have enjoyed the experience when playing Brick Hit. However 32% were neutral and 20% did not enjoy the game. To find out if Brick Hit was suitable as an endless game, we have stated "The game would be better with levels and no endless mode." and asked the participants to share their view. 29.1% agreed to that statement, whereas 37.5% were neutral. 33.3% disagreed to that statement and believe that an endless game was the right genre.

<table>
<thead>
<tr>
<th>Statement</th>
<th>NA</th>
<th>SD</th>
<th>D</th>
<th>N</th>
<th>A</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>When I lose a game, I want to restart to do better.</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>I like the way the game reacts accordingly to my interactions.</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>The game is visually appealing.</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>The game is too difficult.</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>11</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>The gameplay is interesting.</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>The instructions are clear.</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>I don't mind playing the game without being able to &quot;win&quot;.</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>The game would be better with levels and no endless mode.</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>I enjoyed the experience.</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5: Brick Hit game enjoyment summary: Number of votes from 25 participants.

Some participants have left out questions, hence some rows might not equal 25.
Abbreviations: NA = Not applicable, SD = Strongly disagree, D = Disagree, N = Neutral, A = Agree, SA = Strongly agree

Additionally we attempted to investigate the players attitude regarding not being able to "win" the game. Therefore we have stated "I don't mind playing the game without being able to win." and asked the players if they consent. 45.8% have agreed to that statement, while 25% stayed neutral. 29.2% have disagreed and thought it would be more fun to actually have a winning state.
6.2.2 Discussion

The purpose of this evaluation was to investigate the prototypes enjoyment factor and qualification as endless game. Furthermore, the prototype is supposed to implement the majority of our created game design patterns from section 4. The majority of the players have enjoyed the game experience, although other questions have clearly revealed that the game has some improvements to undertake, such as the tutorials, as many stated having difficulties when initially playing.

45.8% of the participants have stated that they are not reluctant to play a game, which they are unable to win, while about 29.2% have claimed to rather win (rest stayed neutral). Therefore, we believe that the endless game genre is not repelled by players, despite its different characteristics. Also, our statement if the prototype would be better as a non-endless mobile game, has exposed that the slight majority does not prefer Brick Hit with levels and a winning state, see figure 30. Hence, we conclude that some uncertainty is present and Brick Hit can be implemented as an endless and non-endless mobile game. However, it may depend on the players preference in gaming genre.

Figure 30: "The game would be better with levels and no endless mode."
7 Conclusions

7.1 Thesis summary

In this thesis we have identified game design patterns specifically for endless mobile games, as current literature has shown a lack regarding this genre. The patterns are supposed to support game developers and designers to exchange and share experience.

Thenceforward we have undertaken case study observations to create a common language for developers in the form of game design patterns specific to our 5 example cases. Our result covers 15 patterns, from which we have utilized 12 to create a prototype instantiation. The prototype "Brick Hit" implements our patterns, which supports mobile developers in understanding and communicating the concept of game design patterns in endless mobile games in a real-usage scenario.

To evaluate our results, we have conducted two questionnaires aimed at developers and players. One questionnaire helped us determining if the game design patterns are pertinent to endless mobile games and useful when designing a game. The other questionnaire was conducted to ascertain that our implementation of the prototype resembles the patterns as a functional game.

7.2 Regarding the research questions

Research question 1

The case study that we have conducted is directly related to the first research question, which guided our thesis outline:

**RQ1** What game design patterns specific to a set of endless mobile minigames can be identified?

The question required us to take a deep look into our 5 case examples. Hence the deduced 15 patterns are generalizations of the observations we have made, which might be supportive from a mobile developers point of view. Each pattern represents a certain characteristic and has been described in section 4. The patterns do not only describe characteristics that are related to the gameplay itself, but also subsidiary attributes such as a leaderboard, challenges or character selection. Although only 7 patterns are mandatory - endless mode, quick progressive difficulty, random level generation, score system, obstacle, simple control, leaderboard - it is crucial to consider the optional patterns to create a complete and enjoyable endless game, as one of our developer comments stated as well (section 6.1). The correlation between the fun factor and the presence of a pattern is definitely a matter to solve in future researches. Each pattern was present in at least 4 out of 5 examples from our case study. However the developer questionnaire shows that the participants were uncertain about 20% of the patterns. Future research and more participants might clarify this matter. As yet we have created a collection of patterns, which can be extended to other genres or revised to comprise more insights and experience. We hope that game designers, developers,
researchers and possibly other related professions can benefit from our study. We believe that game design patterns can be supportive when creating a game and have at least some participants that agree with us:

- "I think that your project is very important for Game Developing and I am very proud to help you with it." - Developer comment regarding the patterns (email response)

Research question 2

RQ2 How can game design patterns specific to endless mobile minigames be applied in a real-usage scenario?

The prototype Brick Hit was developed in order to put the game design patterns into practical use and ascertain that they would be applicable under authentic circumstances. The implemented patterns are depicted in section 5.1. We used 12 out of 15 patterns due to time constraints, leaving 3 patterns. The prototype was developed under the premise of the required patterns from section 5.1, which we strictly followed. Using the patterns helped us create a structured "blueprint" of the game and gave us ideas on how to implement some of the game elements. However since we were not working with different teams, we could only slightly investigate the patterns ability to act as a common language to exchange information. However we have conducted a questionnaire for both players and developers to reveal if the patterns have been appropriately implemented, see evaluation section 6.2. The outcome was that Brick Hit has been accepted as an endless mobile game, which confirms our results to some extent. The entire process of creating the initial prototype with the mandatory patterns and then extending it with the optional patterns is therefore our answer to research question 2.

7.3 Validity of results and limitations

Our game design patterns are derived from five cases, which shall represent the genre of endless mobile games. Hence we limit the results to the 5 representative example cases and are aware that they might not reflect the entire genre. Also the observation attribute "Duration" of each pattern is subjective to our gaming skill level and therefore requires data from more participants to be crucial.

The prototype was created under our interpretation of the patterns, which might differentiate from other developers implementing the same patterns. The created questionnaire from section 6.1 has provided a short-version of each pattern for the developers to evaluate, which might not reflect a patterns description completely. To evaluate the patterns more thoroughly, it is viable to have practitioners understand the patterns entirely and bring them into practical use. This approach would give a more rigorous insight regarding the patterns validity and usefulness.
7.4 Future work

As smartphones are still on the rise and present in our daily lives [9], mobile gaming will most likely stay relevant and hence the demand for developers to strive for better game conception is emerging. Although the majority of the created patterns have been accepted by game developers, there are still advancements to undertake. To improve the game design patterns more cases can be considered to add or revise knowledge. Also a comparison of each pattern with other genres might be useful. We could then answer the question if, for example, a tutorial is really that different in another genre.

Taking a step further, it would beneficial to interview experienced mobile developers of endless games and extract information in person. The research could be extended to a longer period of time to actually test the game design patterns in a small group of developers. This will give a more complete answer to the question whether the patterns are applicable and useful amongst developers.

Interesting to see would be that developers use the patterns for broadly speaking communication. Hence it would be necessary to spread the word in game developer forums or such. More research can then be undertaken with developers that have actual knowledge about the game design patterns for endless games, once they have been established.

Another research area might involve an expansion into a different genre apart from the endless mobile game context. Our study could then be utilized as a template or model to create new game design patterns. Naturally an adaptation of the patterns to the new genre will be required, hence we hope that our work is beneficial to future researchers and practitioners in this or other mobile game related fields.
Appendix A - Game Enjoyment Questionnaire

This questionnaire is designed for players. The sub items contain the type of answer that the participant was provided with.

Personal questions:

- How old are you?
  10-20 | 21-30 | 31-40 | 41-50 | 51+

- If you play MOBILE games, for how long have you been playing?
  Never | Less than 1 year | 1 year | 2 years | 3 years | more than 3 years

- If you play mobile games, how often do you play?
  Never | Daily | Every other day | Weekly | Monthly

- If you play mobile games, how long is a single session usually?
  Not applicable | 5 minutes | 15 minutes | 30 minutes | 1 hour | More than 1 hour

- What is the most appealing about these type of games that you like? (mark multiple if applicable)
  Visuals | Gameplay | Story | They distract me | I can pass the time with them | They are short-lived | They don't require complex instructions or controls

Game prototype related questions:

- When I lose a game, I want to restart to do better.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- I like the way the game reacts accordingly to my interactions.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- The game is visually appealing.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- The game is too difficult.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- The gameplay is interesting.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- The instructions are clear.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree
• I don’t mind playing the game without being able to "win".
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

• The game would be better with levels and no endless mode.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

• I enjoyed the experience.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

**Endless mobile game related questions:**

• A short definition and example images of endless mobile games were provided and then asked: Only one infinite level. Objective examples: reach the furthest distance, avoid as many obstacles as possible, jump on as many platforms as possible. Gameplay progressively becomes harder by adding speed, obstacles, enemies etc. If "Game Over" state is reached, the game starts from the beginning. There is no "winning the game". Game examples: Temple Run 2, Crossy Road, Doodle Jump, Jet Pack Joy Ride.

  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

• The game "BRICK HIT" fits to this definition of endless mobile games.
  No answer | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

• Please mention some endless mobile games that you have played; otherwise leave empty. (Examples: Temple Run, Crossy Road, Flappy Bird, Doodle Jump, Subway Surfers, Despicable Me: Minion Rush, Agent Dash, Robot Unicorn Attack)
  
  Textbox

  61
Appendix B - Game Design Pattern Questionnaire

This questionnaire is designed for developers. The sub items contain the type of answer that the participant was provided with.

Personal questions:

• For how long have you been developing mobile games?
  Less than 1 | year 1 year | 2 years | 3 years | 4 years | More than 4 years

• Have you worked with game design patterns when creating the initial prototype of a game?
  No | Yes | Not sure | Not applicable

• Have you developed an endless mobile game? (Genre examples: Temple Run, Crossy Road, Doodle Jump, Flappy Bird)
  No | Yes | Not sure | Not applicable

Game design pattern questions:
This question is asked for each of the 15 patterns. The description of the patterns is presented in the questionnaire, followed by the statement.

• This pattern characterizes an element of "endless mobile games".
  Not applicable | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

Follow up questions

• Please check all patterns that you think are a MUST to create an endless mobile game. (meaning if you left them out, it wouldn’t be an endless game)
  List of all 15 patterns with check boxes.

• Do you know any other pattern, which is not available in the presented list? If yes, please note down which one. Otherwise skip.
  Textbox

• Game design patterns can support developers in the creation phase of an endless mobile game.
  Not applicable | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

• Game Design patterns can be a good way for a developer team to share common terminology and know specifically what component the other is talking about.
  Not applicable | I strongly disagree | I disagree | Neutral | I agree | I strongly agree
Prototype questions:

- Please watch this short video to answer the next question. It shows the game play of a prototype of an endless mobile game, called "Brick Hit".
  
  An embedded video is shown: https://www.youtube.com/watch?v=GMDyYlBXDAA

- The prototype "BRICK HIT" represents the game design patterns for endless mobile games well.
  
  Not applicable | I strongly disagree | I disagree | Neutral | I agree | I strongly agree

- What features could be missing in the prototype?
  
  Textbox
References


