# Game Illusionization: A Workflow for Applying Optical Illusions to Video Games


Po-Yao (Cosmos) Wang
po-yao.wang@hci.csie.ntu.edu.tw
National Taiwan University
Taipei, Taiwan

Cong-He Xu
National Taiwan University
Taipei, Taiwan
cong-he.xu@hci.csie.ntu.edu.tw

Ping-Yi Wang
National Taiwan University
Taipei, Taiwan
ping-yi.wang@hci.csie.ntu.edu.tw

Hsin-Yu Huang
National Taiwan University
Taipei, Taiwan
hsin-yu.huang@hci.csie.ntu.edu.tw

Yu-Wei Chang
National Taiwan University
Taipei, Taiwan
yu-wei.chang@hci.csie.ntu.edu.tw

Jen-Hao Cheng
National Taiwan University
Taipei, Taiwan
jen-hao.cheng@hci.csie.ntu.edu.tw

Yu-Hsin Lin
National Taiwan University
Taipei, Taiwan
yu-hsin.lin@hci.csie.ntu.edu.tw

Lung-Pan Cheng
National Taiwan University
Taipei, Taiwan
lung-pan.cheng@hci.csie.ntu.edu.tw


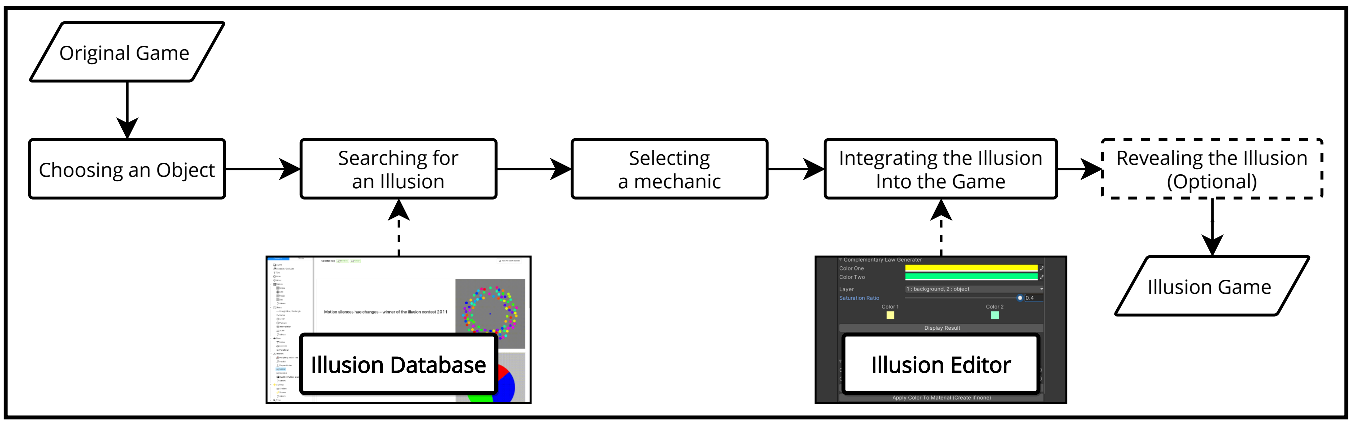


**Figure 1: We propose a workflow for game designers to *illusionize* their games without prior knowledge about optical illusions. The key components in our workflow are our illusion tag database that helps game designers to find matching illusions by describing their in-game visual elements, and our illusion editing interfaces that integrate parameters from prior research for adjusting illusions in designers' games.**


## ABSTRACT

Optical illusions have been brought into recent video games to enhance gaming experiences. However, a large corpus of optical illusions remains unused, while few games incorporate illusions seamlessly. To mitigate the gap, we propose a workflow to guide game designers in applying optical illusions to their video games, i.e., in making more *illusion games*. In particular, our workflow consists of 5 stages: (1) choosing a game object, (2) searching for a matching illusion, (3) selecting an illusion mechanic, (4) integrating the selected illusion into the game, and (5) optionally revealing the illusion. To facilitate our workflow, we provide a tag database with 163 illusions that are labeled by their in-game visual elements and desired effects. We also provide example editing interfaces of 6 illusions for game designers. We walk through our workflow and showcase 6 resulting illusion games. We implemented these 6 games (with and without illusion) and conducted a 12-participant study to gain a preliminary understanding of how illusions enhance gaming experiences. To evaluate our workflow, we invited 6 game designers and 6 experienced players to follow our workflow and design their own illusion games, where 3 experienced game designers completed 2-week in-depth developments. We report their games, qualitative feedback and discuss reflection on our workflow, database and editing interfaces.




*UIST '21, October 10–14, 2021, Virtual Event, USA*



ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474824

## CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**; *User interface toolkits.*

## KEYWORDS

Illusion game, optical illusion, illusion database, illusion editor, game design, game workflow

**ACM Reference Format:**
Po-Yao (Cosmos) Wang, Cong-He Xu, Ping-Yi Wang, Hsin-Yu Huang, Yu-Wei Chang, Jen-Hao Cheng, Yu-Hsin Lin, and Lung-Pan Cheng. 2021. Game Illusionization: A Workflow for Applying Optical Illusions to Video Games. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3472749.3474824

## 1 INTRODUCTION

Optical illusions—"something that tricks your eyes and makes you think you see something that is not really there" [47]—have been discovered for centuries and studied in recent decades. The ancient Greek philosopher Aristotle reported an optical illusion more than 2000 years ago: "when persons turn away from looking at objects in motion, e.g., rivers, and especially those that flow very rapidly, they find that the visual stimulations still present themselves, for the things really at rest are then seen moving" [4]. More optical illusions, such as the simultaneous color contrast illusion [28] that makes the same color looks different in different backgrounds (Fig. 2a) and the Zöllner illusion [64] that makes parallel lines look askew (Fig. 2b), have been deeply investigated by researchers. There are also communities [35, 53] that collect and encourage new explorations of optical illusions.

Optical illusions have long been used non-interactively for understanding the underlying mechanisms of human perception [6] or for viewing and exhibiting [61]. Until 2008, Echochrome [17] started the idea of *illusion game*, i.e., integrating optical illusions in the gameplay. Players interact with illusions such as impossible objects [46], modal completion, and amodal completion [27] to solve puzzles in this game, resulting in a new gaming experience that plays around human perception. This has inspired successful games such as Monument Valley [21] and Superliminal [20].

However, integrating illusions and games is challenging. The number of combinations between illusions and games is considerably large, while only a few illusion games exist for figuring out a heuristic. In the meantime, game designers have to fill in the knowledge gap, i.e., the underlying mechanisms of illusions by which to adjust the corresponding parameters for their games and thereby figure out how to interact with illusions. A large corpus of optical illusions thus remains unused and non-interactive.

In this paper, to illuminate the path to design/create more illusion games that draw inspiration from more kinds of optical illusions, we propose a workflow to guide game designers in applying optical illusions to their video games. Unlike previous game design processes, we integrate the knowledge of optical illusions by providing an illusion tag database and 6 specific illusion editors to facilitate game designers in developing their illusion games smoothly. We have the following 3 contributions:

- forming a workflow for game designers to integrate optical illusions into video games,
- developing the illusion tag database for game designers to retrieve candidate illusions by describing in-game visual elements and desired effects,
- building 6 example illusion editors that mitigate the knowledge gap for 6 specific illusions.

## 2 RELATED WORK

This work relates to research on (1) optical illusions, (2) illusion applications, and (3) game design guidelines.

### 2.1 Optical Illusions

A good number of optical illusions have been discovered in human history. The Best Illusion of the Year Contest [53] recruits and calls for new illusions every year. Macpherson et al. [35] and Michael Bach [5] maintain continually expanding databases with interactive interfaces for users to understand the collected optical illusions. As more illusions have been identified over the years, researchers propose taxonomies to provide systematic overviews. Jacques Ninio [43] proposes orientation profiles to classify geometrical illusions. Richard L. Gregory [23] classifies the illusions' phenomena into 16 classes in terms of appearance and causes. Michael Bach also comes up with six categories including luminance/contrast, motion, geometric/angle, 3D interpretation, cognitive/gestalt effects, and color [6]. However, these taxonomies do not take game design into considerations and focus mostly on the effect of illusions. This gives game designers a hard time narrowing down the search on certain illusions and putting illusions into games. We thus revisit the large corpus of illusions and come up with a new set of hierarchical tags that incorporate in-game elements to speed up the design process.

Researchers in psychology have studied, in addition to taxonomies, the underlying mechanisms that induce optical illusions. August Kirschmann [28] found the possible influences of simultaneous color contrast (Fig. 2a) and constructed five laws according to his experiment. For more than a century, the laws have led to further supportive experiments [33, 49] as well as to new refuting discoveries and laws [14, 15]. Tadasu Oyama [45] investigated the effects of stimulus variables, such as orientation, intersection angle, and gap, on the Zöllner illusion (Fig. 2b) and determined the factors that influence the strength of the illusion.

While there are several studies on optical illusions that measure and discover the underlying mechanisms, we take a closer look at the rest of the illusions in Fig. 2 that are used as examples in the following sections. The Munker–White illusion (Fig. 2c) makes the same color seem different by placing single color rectangles on the high contrast stripes background. Taya et al. [57] studied the influence of stereoscopic vision while Budimir et al. [11] investigated the relationship between the thickness of the stripes and the strength of illusion. The footsteps illusion (Fig. 2d) makes two objects with different colors seem to move step by step while they are moving smoothly by adding stripes on the background. Stuart Anstis [2] explained the illusion using a model of motion

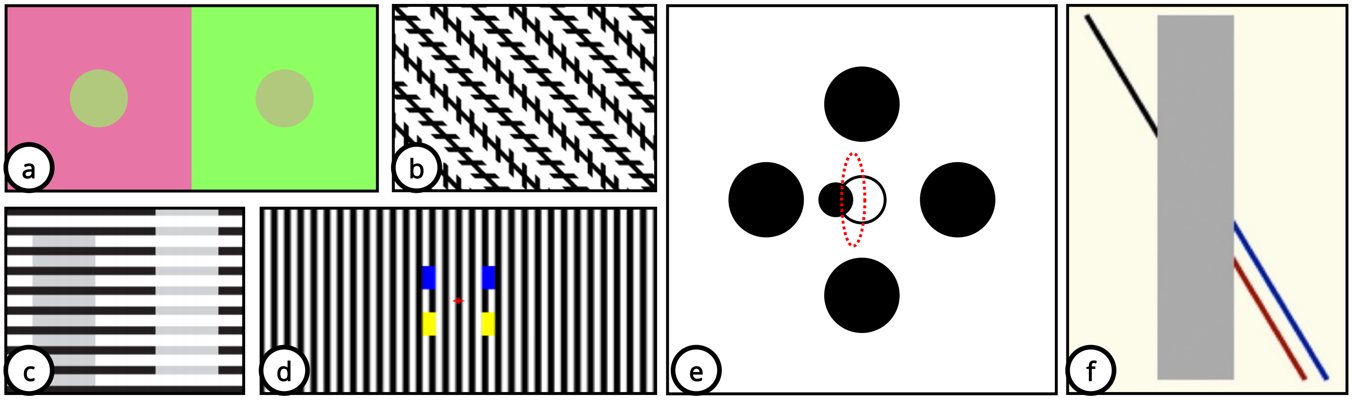


**Figure 2: (a) An example of simultaneous color contrast illusion [15]. The tan color in the circles seems different from the pink and green background color. (b) An example of the Zöllner illusion [35]. The parallel diagonal lines seem askew when adding short crossing stripes. (c) An example of the Munker–White illusion [11]. The grey colors are the same but are perceived differently as they are surrounded by black and white stripes separately. (d) An example of footsteps illusion [30]. The blue and yellow rectangles are aligned vertically while moving forward continuously, but they seem to advance step by step in turn. (e) An example of hierarchical motion organization [40]. The circle in the center orbits around the center point (the black trajectory). As the peripheral circles start orbiting, it seems that the trajectory of the center circle is distorted (the red trajectory). (f) An example of the Poggendorff illusion [35]. The black line links to the red line, while it seems to link to the blue line with the cover of the grey rectangle.**

coding induced by luminance contrast. Hierarchical motion organization (Fig. 2e) distorts the movement of the center object depending on the movement of peripheral objects. Ryan Mruczek and Gideon Caplovitz [40] explored several patterns to induce the illusion while other studies [7, 8] discussed their implementations. The Poggendorff illusion (Fig. 2f) makes a continuous line seem to offset to another by adding an occluder. Daniel Weintraub and David Krantz [62] conducted multiple experiments on varying angles to determine their influence.

## 2.2 Illusion Applications

Researchers have used sensory illusions in various applications. Mise-Unseen [36] used eye-tracking technology in virtual reality (VR) scenes to hide changes that occur in plain sight. "DeepTaste" [42] attempted to illusionize the taste of somen noodles to ramen noodles by generating visual and olfactory stimuli in augmented reality (AR). Similar usages were also seen in the field of haptics, where the idea of "Pseudo-haptic" [48, 51] was introduced, allowing users to perceive haptic feedback based on visual effects.

In recent years, game designers have brought optical illusions as part of their core gameplay. Echochrome [17] is the pioneer of illusion games. In Echochrome, the player's goal is to help a mannequin traverse the world by rotating the world, creating roads using the illusion of impossible objects [46], modal completion, and amodal completion [27]. Several games also use these illusions to create similar gameplay [9, 16, 34].

One of the most popular illusion games is Monument Valley [21]. Combining impossible objects with the aesthetic of minimalism, Monument Valley has won over 20 awards since 2014. Several games also integrated modal completion illusion [18, 19, 55]. In these games, players rotate their perspective to find the right angle from which to see separate objects as one. A recent role-playing game, Assassin's creed Valhalla [59], also uses modal completion illusion as puzzles to be solved in the quests. On the other hand, Superliminal [20] provides a new take on illusion games by telling players that *the perception is the reality.* In Superliminal, players escape from dreams and use the forced perspective [56] to build paths and Trompe-l'oeil [13] to instantiate objects. While most illusion games are reusing the same illusions to solve puzzles, there is a large corpus of unused illusions. We see great potential in borrowing the elements from the large body of optical illusions to facilitate the game design process.

## 2.3 Game Design Guidelines

Researchers have proposed several guidelines to make games. Joseph Saulter [52] suggests that the process of game development is divided into four phases: conception, preproduction, production, and postrelease. Zackariasson et al. [63] emphasize the importance of innovative and creative work and suggest that video game development needs continuous adjustment. Juergen et al. [41] derived a video game software development process based on Scrum, which fits better into creative projects with high uncertainties. To enhance the development of computer games, Guo et al. [24] introduced a workflow based on Domain Specific Modeling, which enables fast prototyping by generating codes with the help of Domain Specific Language development tools, boosting the performance in the production stage. We aim to develop a workflow to eliminate the entry barriers of illusion game design.

# 3 ILLUSIONIZING A GAME

We present a workflow (Fig. 3) for making illusion games. We provide an illusion tag database, 2 game mechanics, 6 illusion editing

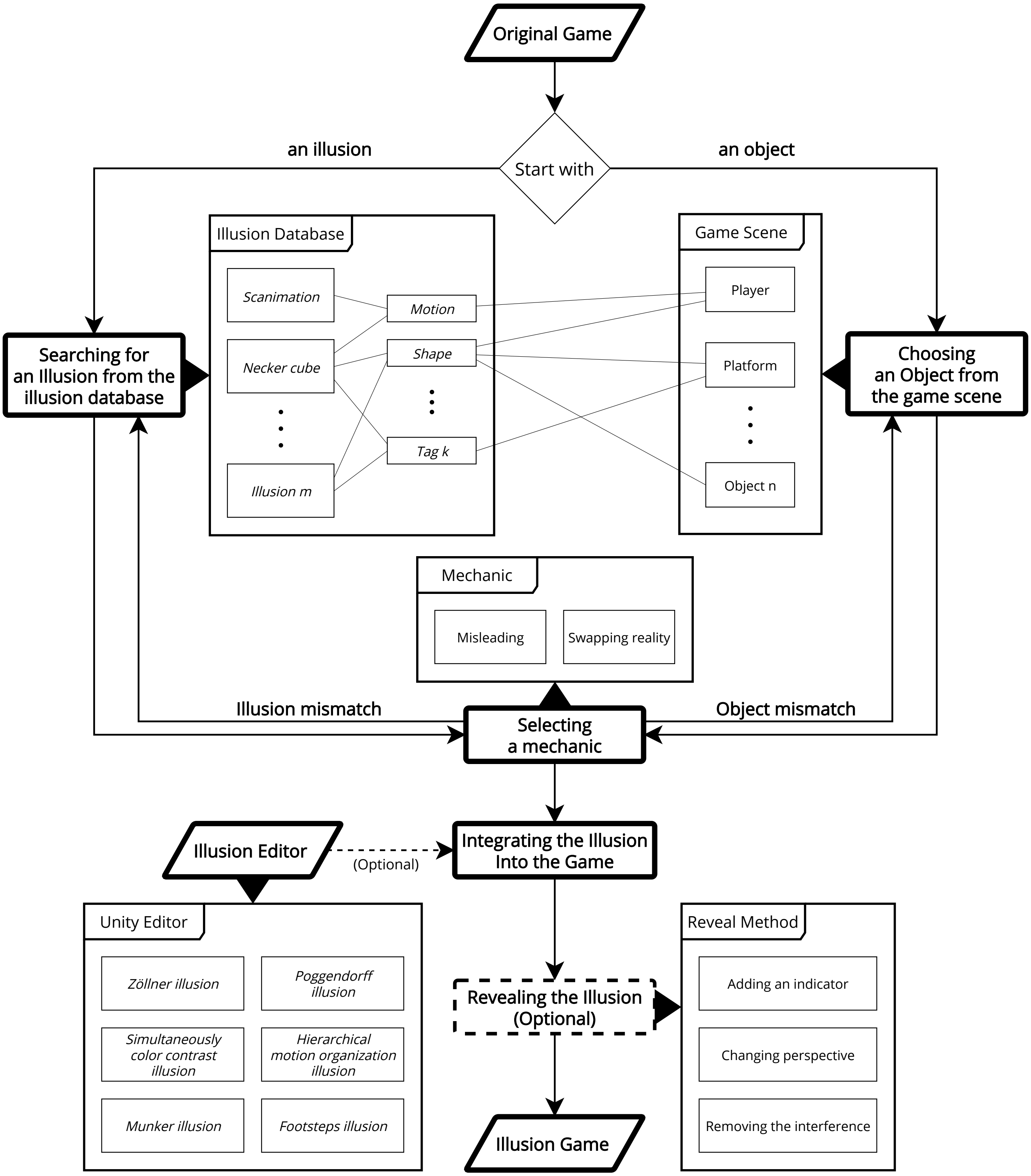


**Figure 3: The complete flowchart of our workflow. For the following walkthrough, we start with an in-game object and then search for a suitable illusion by selecting corresponding visual tags that describe the object and the effect in our database. Our database elicits candidate illusions based on the selected tags. The process may also start from an illusion with the desired effect and then look for in-game objects that contain the visual elements of the illusion's tag. We iterate the game design between the candidate illusions, in-game objects, and game mechanics until found a satisifying combination. We implement our design and integrate the illusion into the game using our illusion editors. Finally, we optionally add an explicit reveal of the illusion.**

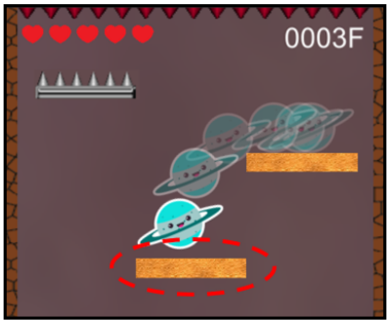


**Figure 4: We take our NS-Shaft-like game as an example. In this game, the player controls the character to move left and right and falling from platform to platform. The player's goal is to survive from being crushed to the spiked ceiling or falling off the abyss. We choose platforms as the target game object.**

interfaces, and 3 ways to reveal illusions to facilitate our workflow. We elaborate on our workflow by *illusionizing* our NS-Shaft-like game [32] (Fig. 4). In this game, players move the character left or right to drop on the next rising platform while avoiding falling into the abyss or crashing into the spiked ceiling.

## 3.1 Choosing an Object

In this example, we start our workflow by choosing a game object. In our NS-Shaft-like game, the character, spikes, and platforms can be chosen to apply illusions. There is no constraint at this stage as each game object has its potential to make a distinct illusion game with a matching illusion. Here we select platforms as the game object for applying illusions (Fig. 4).

## 3.2 Searching for a Matching Illusion

Next, we search for related illusions of the selected game object from our illusion database. Our database interface displays candidate tags, selected tags, and previews of related illusions (Figure 5). When tags are selected, our database updates selected tags and previews of related illusions in real-time. Each of the previews contains a GIF, a short description of the illusion, and related tags for searching for more related illusions.

Illusions in our database are labeled with two meta-level tags that associate to game design: *Visual Elements* and *Effects*. *Visual Elements* tags are used for matching the appearance of game objects, while *Effects* are used for matching possible actions of game objects, e.g., designers search the tag *Hiding* when they want to make a game object invisible. In this example, we observe the visual elements of the standing platforms. Each standing platform is a rectangle and parallel to the other. We thus select the tags *Shape-Straight line / Rectangle* and *Relation-Parallel.* A list of illusions labeled with these two tags is returned and shown. We then select the

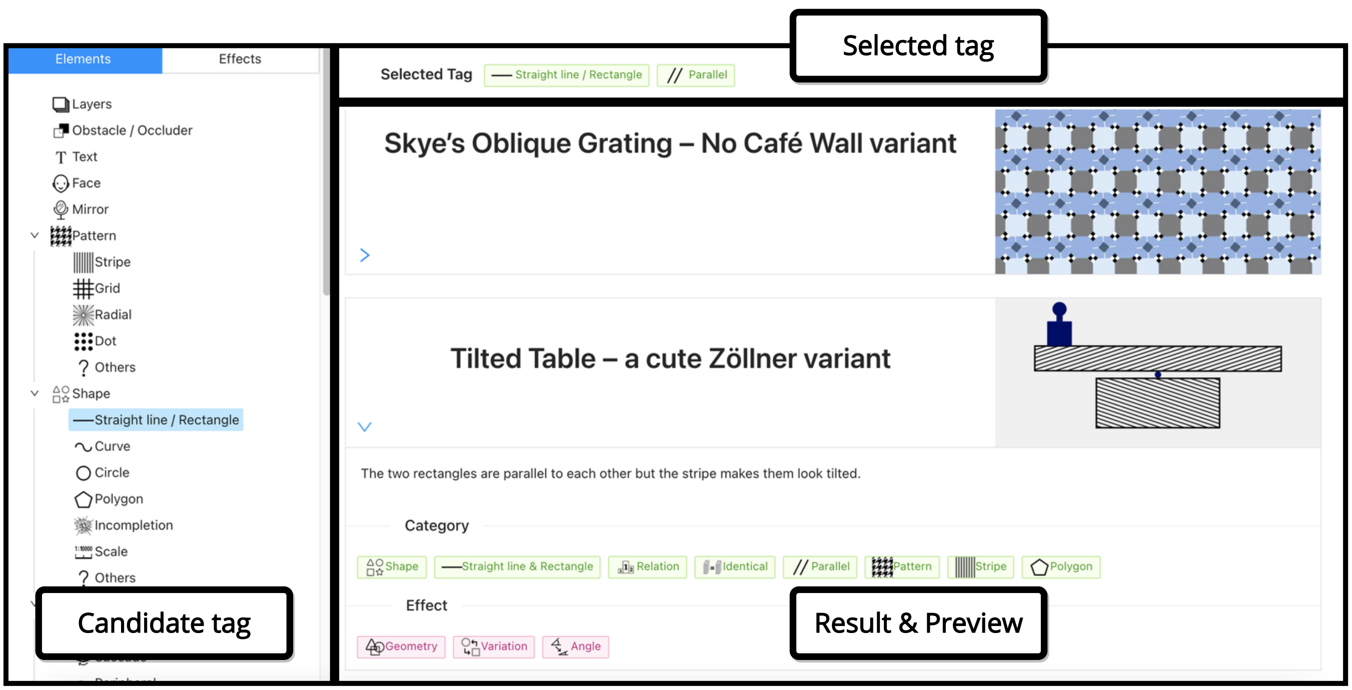


**Figure 5: An overview of our database interface. On the left side, the user selects tags based on the visual elements or the effects that the user wants to pursue. On the right side, our database returns a list of candidate illusions with their names and animated graphics. The user clicks the illusion to expand for more information, such as a short description and all tags attached to the illusion.**

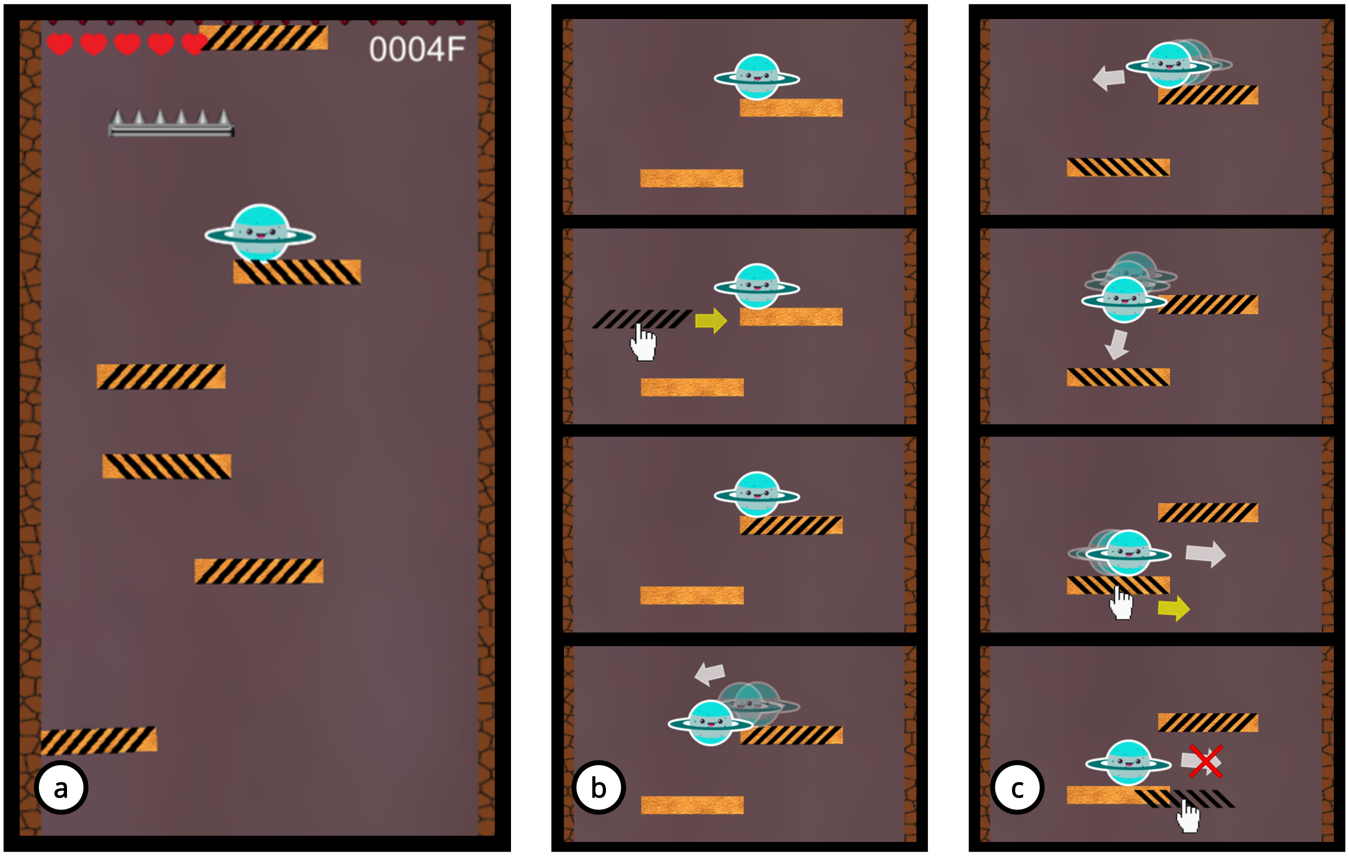


**Figure 6: We offer two mechanics to apply illusions to the gameplay. (a) *Misleading*: the illusion confuses the player as to whether the platforms are tilted, increasing the difficulty of the game. (b) & (c) *Swapping Realities*: the player manually adds/removes the stripes to/from the platform to straighten/tilt it. The character slides or stops after the stripes are added or removed.**

Zöllner illusion [64], an illusion that makes straight lines seem askew (Fig. 2a), to apply to those standing platforms.

On the other hand, if the designer aims to create certain visual effects, they search for illusions with the *Effects* tags. They then pick a game object that fits the selected illusion. The steps *Choosing an Object* and *Searching for a matching mechanic* are thus interchangeable.

## 3.3 Selecting an Illusion Mechanic

After determining an illusion, we proceed to consider applying illusions to the objects. In our workflow, we provide 2 illusion mechanics: *Misleading* and *Swapping Realities*.

*3.3.1 Misleading.* We consider misleading players with illusions to increase the games' difficulty. To mislead players, we place not only game objects with illusions but also their similar objects without illusions. In this example (Fig. 6a), we add both tilted platforms and platforms with the Zöllner illusion to make it hard for players to distinguish whether the platforms are tilted or not. The character only slips on the tilted platforms but not on the illusionized platforms that seem tilted. With *Misleading*, players have to pay extra attention to fight their own perceptions. We can also use this mechanic in competition games, such as Mario Kart [38], where players illusionize their opponents' game objects to interfere with their progress.

*3.3.2 Swapping realities.* We consider swapping games' realities from without illusions to with illusions. As shown in Fig. 6b, players control the character by manually adding the Zöllner illusion to the platforms to let the character slip down instead of directly moving it. This turns illusion into reality as the platform is actually flat but seems and acts tilted. The other way around is to swap from illusion back to reality. As shown in Fig. 6c, players now have to swipe out the skew lines from the platforms to stop the characters from slipping down. The key is to let players figure out what illusion is and how to use it during the gameplay. We can use this mechanic to create puzzles and quests in games.

We iterate through the combinations of game objects, illusions, and illusion mechanics back and forth as shown in Fig. 3 until we find a satisfying design. In this example, we are satisfied with the design that combines platforms, the Zöllner illusion, and the misleading mechanic.

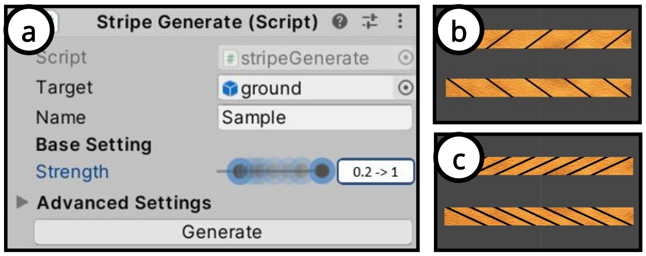


**Figure 7: The Zöllner editor: (a) the base setting for applying the Zöllner illusion to the selected game object. When adjusting the strength from weak (b) to strong (c), the editor displays the corresponding result on the scene simultaneously.**

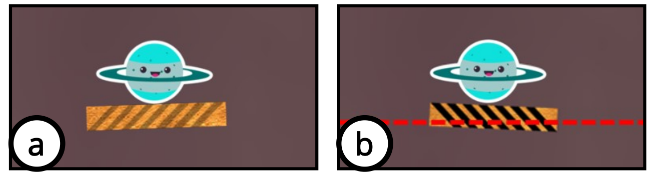


**Figure 8: We provide two examples for revealing the Zöllner illusion. (a) *Removing the interference*: When the character stands on a platform, the attached stripes fade out to make players see clearly whether the platform is tilted or not. (b) *Giving an indicator*: Give a horizontal red line as an indicator in the game. When the platform passes through the line, players will know whether the platforms are sloping or not.**

## 3.4 Integrating the Selected Illusion into the Game

After finding a satisfying design, we proceed to implement the illusion game. To reduce the effort of figuring out the underlying mechanisms of illusions, we offer 6 specific illusion editors that integrate the key features of each illusion from prior research and set them as parameters for further adjustment (more details in section [5]). In this example, we use the Zöllner illusion editor to apply the illusion to the platforms. As shown in Fig. 7, we drag the *Strength* value to find an appropriate setting for players to better perceived the illusion on the platforms. We can also adjust more parameters in the *Advanced Settings*, such as changing the orientation of the skew lines to make the platforms tilt to the other side. Finally, we click the *Generate* button and creates the corresponding game object that runs in one's game. At this stage, we have successfully illusionized the NS-Shaft-like game.

## 3.5 Optionally Revealing the Illusion

Since players may be unaware of illusions or regard illusions as visual bugs and glitches, we suggest 3 options to reveal illusions explicitly during the gameplay. We leave revealing optional as game designers can decide to which extent they would like to reveal. In other words, these are techniques that we use in our illusion games to give hints or make sure that players *notice* their perceptions are inconsistent with reality. We should note that there could be more valid options to reveal illusions, e.g., voice or text instructions that explain the illusions, which is beyond the scope of this paper.

*3.5.1 Adding indicators.* This technique is suitable for illusions with the tags of (1) *Motion*, (2) *Duality*, (3) *Hiding*, and (4) *Relation*. Adding the actual paths of moving objects affected by the *Motion* illusions helps players notice that the trajectories are different from what they perceive. Adding highlights of different contours in turns helps players notice the *Duality* illusions, i.e., images that can be seen in different ways. The *Hiding* illusions can be revealed by adding the hiding objects explicitly. The *Relation* illusions can be revealed by adding comparing objects.

*3.5.2 Changing perspective.* This technique is suitable for illusions with the tag of *Perspective/Anamorphosis*. Changing the camera's perspective makes players notice illusions that only happen from a certain point of view. Zooming in the camera also reveals some illusions, such as the footsteps illusions, because players can perceive more details. Since changing the camera's perspective may seriously disturb the gameplay, we suggest using this technique in cut scenes such as the death replay or the victory scene.

*3.5.3 Removing the interference.* This technique is suitable for illusions with the tag of *Obstacle/Occluder*. The *Obstacle/Occluder* illusions happen when there are occluders that interfere with players seeing objects as a whole. The illusions are revealed as long as the occluders are removed from the objects.

In our NS-Shaft-like game, it is clear that removing the interference on the platforms can make players know whether the platforms are actually tilted or not (Fig. 8a). However, adding a horizontal line in the game could also help players to distinguish the flatness of platforms (Fig. 8b).

## 3.6 More Examples

We show more examples that use our workflow to integrate other illusions with various types of games.

**Gerritory-like game with the simultaneous color contrast illusion**. We illusionize a territory occupation party game, Gerritory [22]. In the original Gerritory, each player rolls their own cube to color more tiles to their color in time to win (Fig. 9a). To illusionize this game using our workflow, we choose the colored tiles (*Choosing an object*). We see that the colored tiles should have these visual elements: *Color*, *Relation-Similar* and *Multiple Object*. After examining the illusions of these tags from our database, we select the simultaneous color contrast illusion (*Searching for an Matching Illusion*). The simultaneous color contrast illusion makes people perceive the same color differently in different backgrounds (Fig. 2b). We consider 2 illusion mechanics (*Selecting an Illusion Mechanic*): (1) *misleading* players that the tiles have been colored with their colors and (2) *swapping realities* so that players have to figure out a way to make all tiles seem the same color instead. We choose the *misleading* mechanic and implement the game using our simultaneous color contrast editor (*Integrating the Selected Illusion into the Game*). Fig. 9b show the illusionized Gerritory. The illusion is revealed when the cube cross from one background to another (Fig. 9c). The cube itself is the indicator.

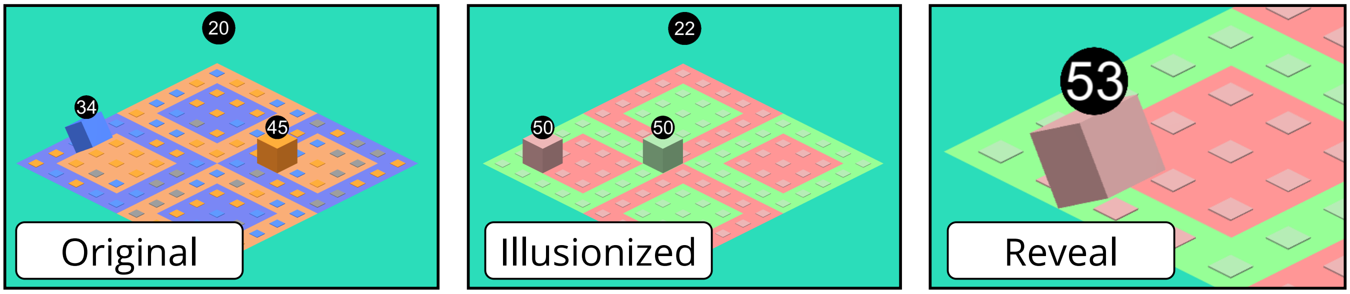


**Figure 9: A Gerritory-like game with simultaneous color contrast illusion. Each player moves their cube and takes over territories to win the game. The simultaneous color contrast illusion makes it difficult for players to distinguish their territories. They realize that there are more territories to take over when they move outside the floors in their same color.**

**Road-Fighter-like game with the Munker–White illusion.** We illusionize a 2D side-scrolling Road Fighter [31]-like game (Figure 10). In the original game, players steer the forward-moving character to collect coins that match the character's color while dodging obstacles and coins with other colors. Since color is an important element in this game, we *mislead* players by varying the colors on the coins (*Choosing an object*). We look up our database with the tag *Color-Variation* and select the Munker–White illusion (*Searching for a Matching Illusion*). The Munker–White illusion makes the same color seem different under the high contrast stripes (Fig. 2c). To *mislead* players, we add the high contrast stripes on the right half of the scene (*Selecting an Illusion Mechanic*) to interfere with the players' color perceptions. We implement the game using our Munker–White Editor (*Integrating the Selected Illusion into the Game*). The illusion is revealed when a coin moves out of the stripes area. We also reveal the illusion by changing the camera's perspective in the death replay.

**Billiards with the Poggendorff illusion.** We illusionize a billiards game (Fig. 11). In the original game, the aiming line is shown. Players control the aiming to hit balls into pockets. We choose the aiming line (*Choosing an object*). The aiming line appears to be *Shape-Straight line* and may achieve the effect of *Geometry-Variation-Straight line/Rectangle*. With these 2 tags, we select the Poggendorff illusion from our database (*Searching for a Matching Illusion*). The Poggendorff illusion generates an offset between the objective and the perceived collinearity of lines (Fig. 2d). We *mislead* players with the decoy aiming line (*Selecting an Illusion Mechanic*). We implement the game using our Poggendorff editor (*Integrating the Selected Illusion into the Game*). The illusion is revealed by removing the obstacle so that the players can observe the real aiming line. This is used as a debuff for the leading player to make the competition more intense.

**Space-Invaders-like game with hierarchical motion organization.** We illusionize Space Invaders [44], a classic Shoot 'em up game (Fig. 12). In the original game, players shoot down enemies and avoid enemies' fire. We choose the enemies (*Choosing an object*). With the tags of *Motion*, *Relation-Peripheral and Center*, and *Relation-Similar*, we select the Rotating Circles Illusion from our database (*Searching for a Matching Illusion*). The Rotating Circles Illusion is an instance of hierarchical motion organization where

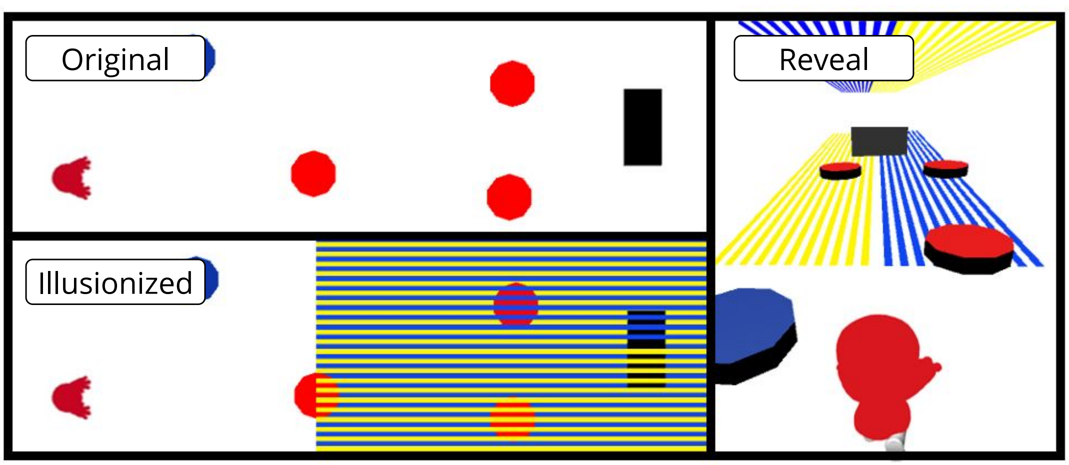


**Figure 10: A Road-Fighter-like game with the Munker–White illusion. Players steer the forward-moving character to collect the coins with the same color as the character. After adding the illusion, the colors of coins are perceived differently under the stripes. In the death replay, the illusion is revealed using the other perspective.**

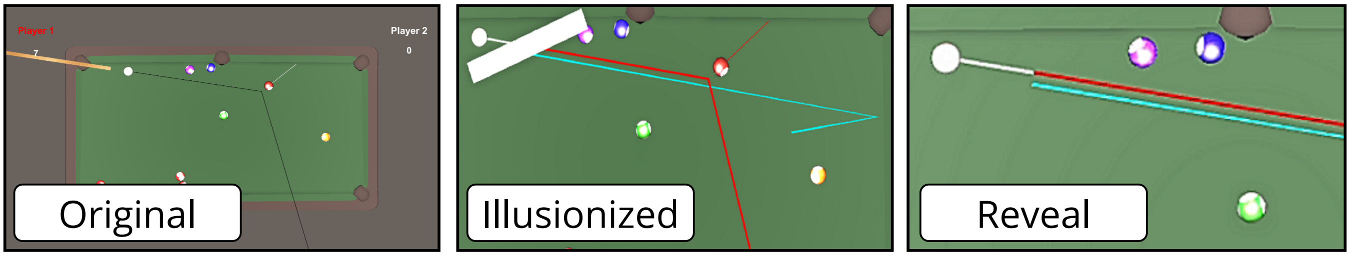


**Figure 11: A billiards game with the Poggendorff illusion. Players control the aiming line to hit balls into pockets. The Poggendorff illusion misleads players to take the decoy blue aiming line. The illusion is revealed after removing the occluder.**

the motion in the center is misperceived because of the motion in the periphery (Fig. 2e). We *mislead* players by making the boss's motion unpredictable by adding satellites rotating around the boss (*Selecting an Illusion Mechanic*). We implement the game using our Hierachical Motion Organization editor (*Integrating the Selected Illusion into the Game*). The illusion is revealed by showing the trails of the boss.

**Taiko-No-Tatsujin-like game with the Footsteps illusion.** We illusionize a famous rhythm video game, Taiko No Tatsujin [54] (Fig. 12). In the original game, players beat the drum when the notes arrive at the receptor. We choose the note (*Choosing an object*). The note is *Shape-Circle* and *Motion-Linear*. With these two tags, we select the footsteps illusion from our database (*Searching for a Matching Illusion*). The footsteps illusion makes people perceive an asynchronous movement of two high-luminance-contrast objects moving at the same speed (Fig. 2f). We use the illusion to *mislead* players by affecting their sense of rhythm (*Selecting an Illusion Mechanic*). We also consider *swapping realities* where players manually add/remove stripes to correct the timing of the notes. We implement the *misleading* version using our footsteps illusion editor (*Integrating the Selected Illusion into the Game*). We reveal the illusion by zooming in the notes and the stripes in the replay.

## 4 FORMING WORKFLOW AND TAG DATABASE

We designed and developed the workflow and the tag database through a formative study. The initial goal of the workflow was to integrate a suitable illusion into an existing game. The tag database should help game designers quickly identify a suitable illusion.

We started by collecting 143 illusions from Michael Bach's website [5] (denoted as $S_1$) and another 20 illusions from the Best Illusion of the Year Contest [53] (denoted as $S_2$). Each author extracted an initial set of descriptive words about illusions in $S_1$ in two meta-levels: *Visual Elements* for matching the appearance of game objects and *Effects* for matching actions. We then merged those initial sets by an affinity diagram and established the initial *TagTree*, i.e., a hierarchy of tags. We then recruited 9 game designers (more than 1 year experience, 3 females, aged 22 to 24, $M = 23.33, SD = 0.67$) from our institute for our formative study. Their task was to use the TagTree to find a suitable illusion and describe how they would integrate it into their games step by step. We brought in participants in groups of 2, 4, and 3 people at a time and thus went through 3 iterations. We asked them to speak out about confusing tags and collected their design principles when they were integrating the illusion into their games. We then refined the TagTree based on their feedback and provided it along with extracted principles as our workflow to facilitate the design process for the next group. The study ended as all tags were distinguishable and considered generalized enough by the study's participants, i.e., more than one

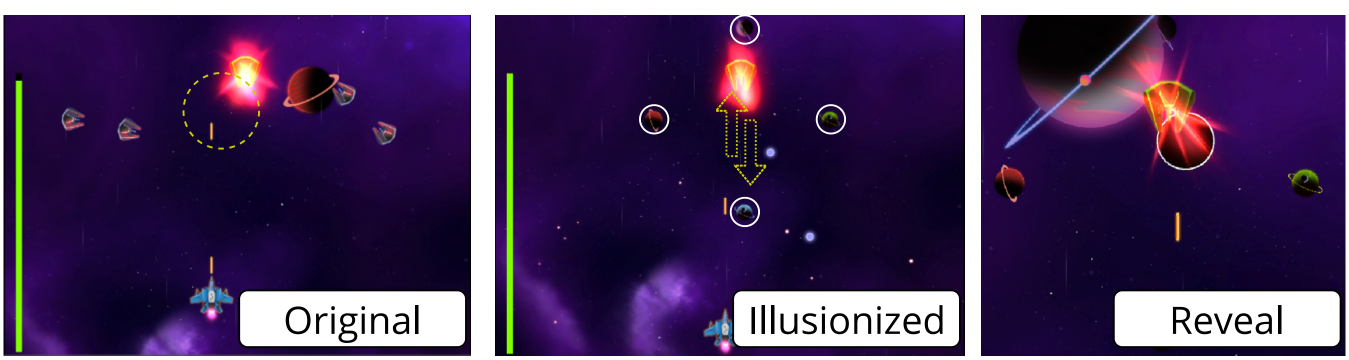


**Figure 12: A Space-Invaders-like game with hierarchical motion organization. Players shoot down enemies and the glowing boss who rotates in a circle all the time. With the hierarchical motion organization, players see irregular motion instead because of the satellites' movement around the boss. The motion trail of the boss is used to reveal the illusion.**

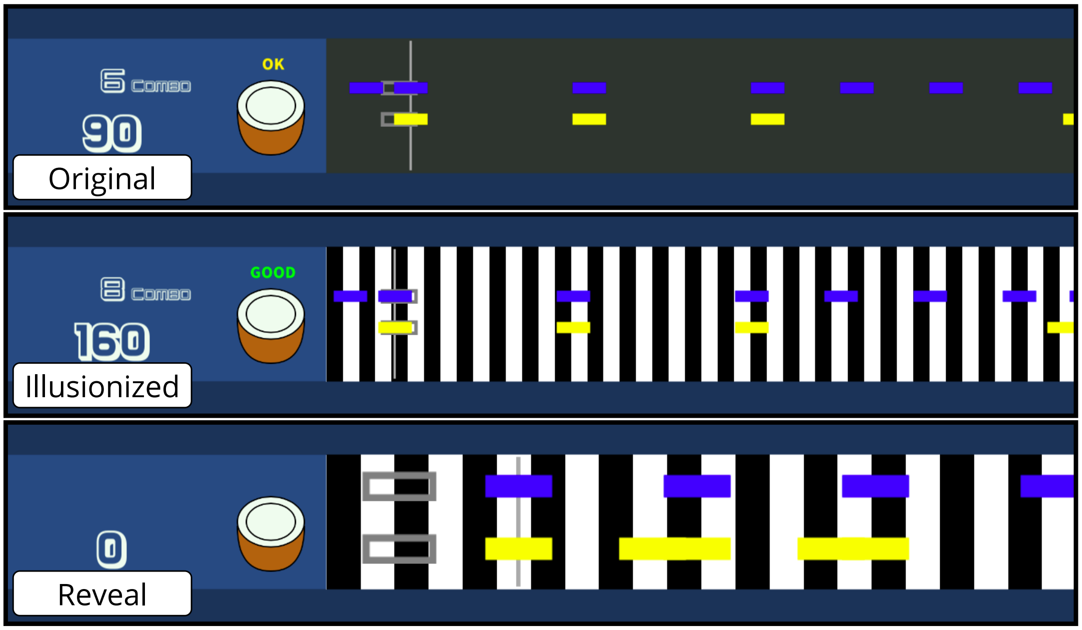


**Figure 13: A Taiko-No-Tatsujin-like rhythm game is illusionized with the footsteps illusion. The player beats the drum face/rim when the yellow/blue notes arrive at the grey receptors. The footsteps illusion makes the notes move like footsteps to mislead the tempo. The replay reveals the illusion by zooming in the view.**

illusion could be retrieved by each tag. We spent about 2 hours for each group and finalized our TagTree and workflow for the later study.

The first two groups used a Google Sheet mock-up displaying all tags and illusions in a text table at once. However, during the study, we observed that some of the participants' confusions came from the interface. We thus built a web interface and a tag database on a website[1] to mitigate the problem. The interface (Fig. 5) was implemented using React, a Javascript front-end framework. Selecting tags on the interface automatically generates a partial TagTree as a new query, and the page is automatically refreshed with the new result accordingly. We built a back-end application running MongoDB (a NoSQL database) with Koa, a Node.js framework, to handle requests from the front-end. The last group validated the finalized tags and our web interface.

After the formative study, we applied our finalized tags to the illusion set $S_2$ to validate the generality of our tags. Fig. 14 shows our finalized tags.

## 5 ILLUSION EDITORS

We provide game developers with 6 specific illusion editors: (1) the Zöllner illusion, (2) the simultaneous color contrast illusion, (3) the Munker–White illusion, (4) the Poggendorff Illusion, (5) the hierarchical motion organization, and (6) the footsteps illusion. Inspired by python-based parametric frameworks for illusions [7, 50], we extract the illusions' parameters from prior researches [2, 7, 8, 11, 14, 15, 28, 33, 40, 45, 49, 57, 62] and wrap them up as editors that run in Unity [58] game engine. With the editors, game developers can adapt the illusions for their games without extra effort in understanding the underlying mechanisms.

[1]http://hci.csie.ntu.edu.tw/illusiondb

### 5.1 Zöllner Illusion Editor

Fig. 7 shows our Zöllner illusion editor. Users assign the target object in the editor interface to apply the Zöllner illusion. The editor automatically generates diagonal stripes based on the strength value and shows the preview of the attached object. Stronger the value more askew the two parallel objects seem. Users can adjust the strength value easily by dragging the strength slider to fit their applications. The strength value is based on Determinants of the Zöllner illusion [45] where the density, the length, and the angle of the strips are counted together. These details can be further controlled in the advanced settings.

### 5.2 Simultaneous Color Contrast Illusion Editor

Fig. 15 shows our simultaneous color contrast illusion editor. The editor has 2 modes: (1) making a color seem identical to another distinct color in a certain background using the direction law [15] (Fig. 15a), and (2) making the same color seem different in different backgrounds using the complementary law [14, 33, 49] (Fig. 15b).

In the direction law mode, users assign 2 colors in the first layer and the second layer in the editor. The editor automatically calculates, for example, the target color that seems the same as the first layer color if using the second layer color as its background. The target color can be further adjusted with the alpha and the error sliders to calibrate for individual perception. The editor shows the preview of the results in real-time and exports the corresponding

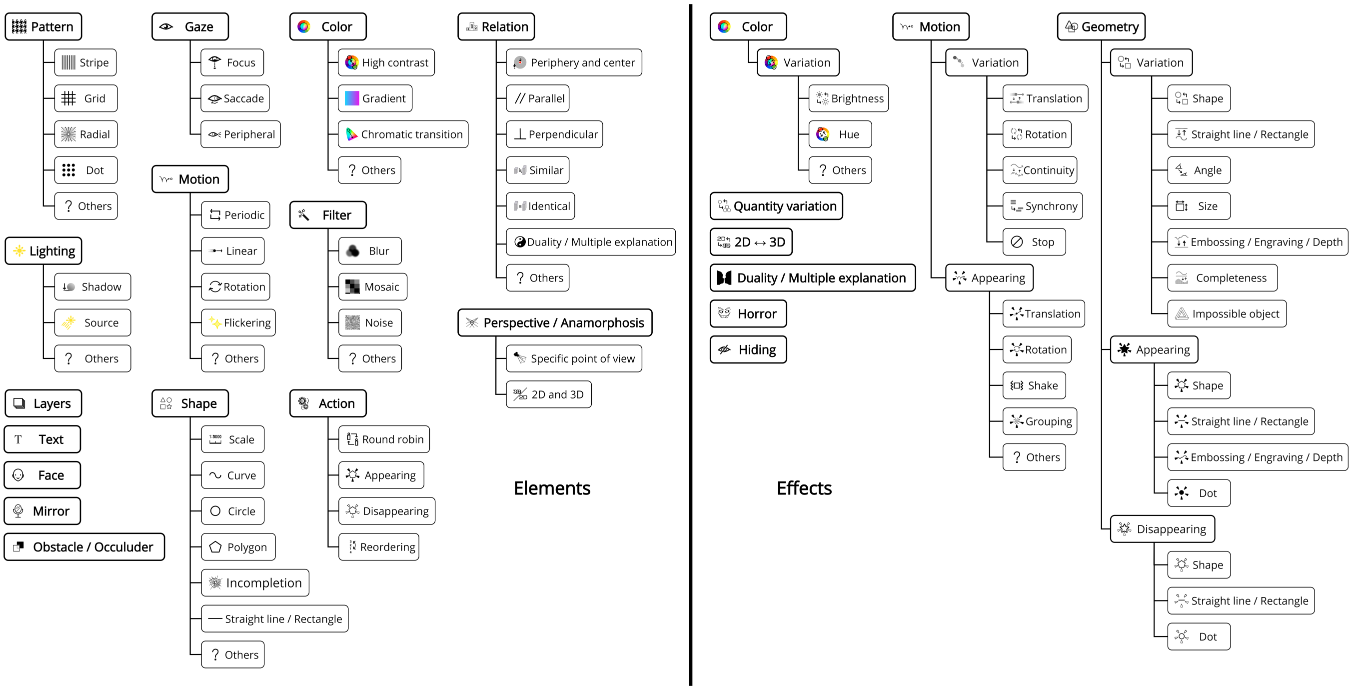


**Figure 14: All tags in our database. There are two types of tags in our database, the *Visual Elements* tags and the *Effects* tags. The *Visual Elements* tags describe the characteristics of game objects, such as *Pattern* and *Relation*. The *Effects* tags describe the actions, such as *Color-Variation-Brightness* and *Geometry-Appearing-Dot*.**

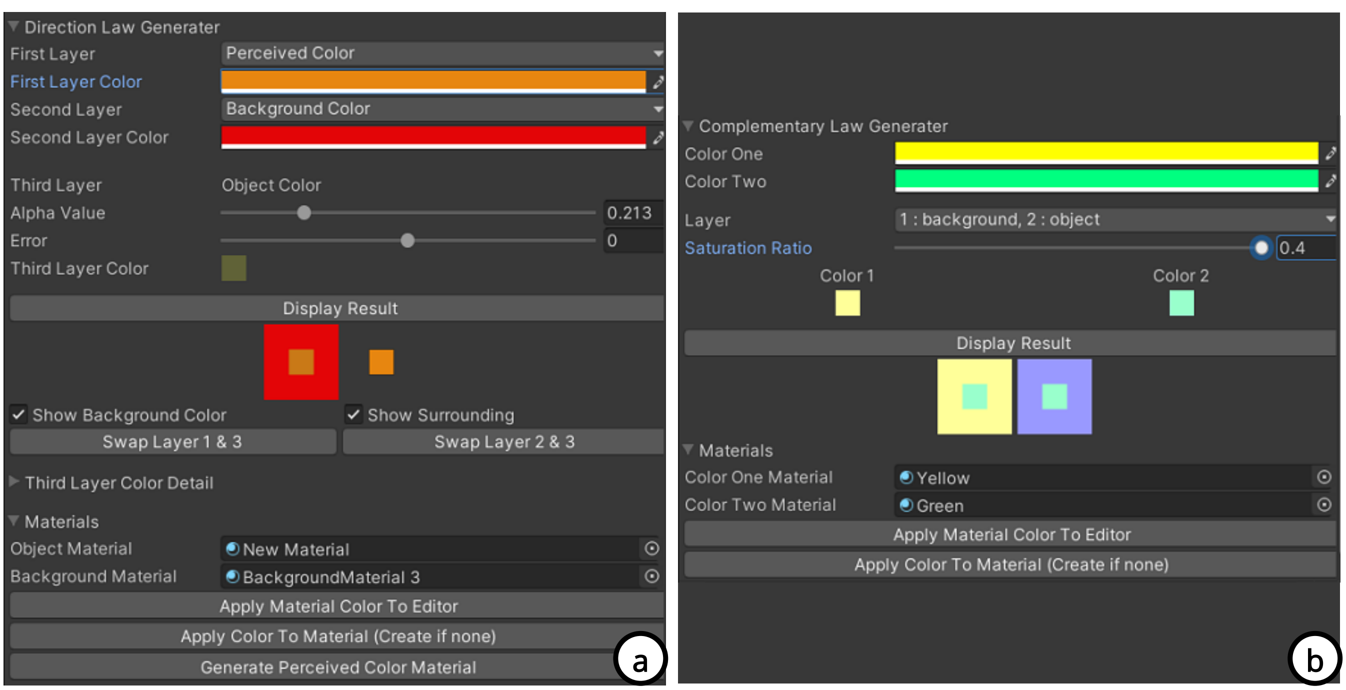


**Figure 15: Our simultaneous color contrast illusion editor. (a) The direction law editing interface. The editor generates the color of the target layer according to the given color of the reference layers. (b) The complementary law editor interface. The editor generates the color of the target layer that maximizes the perceived foreground's contrast on different backgrounds.**

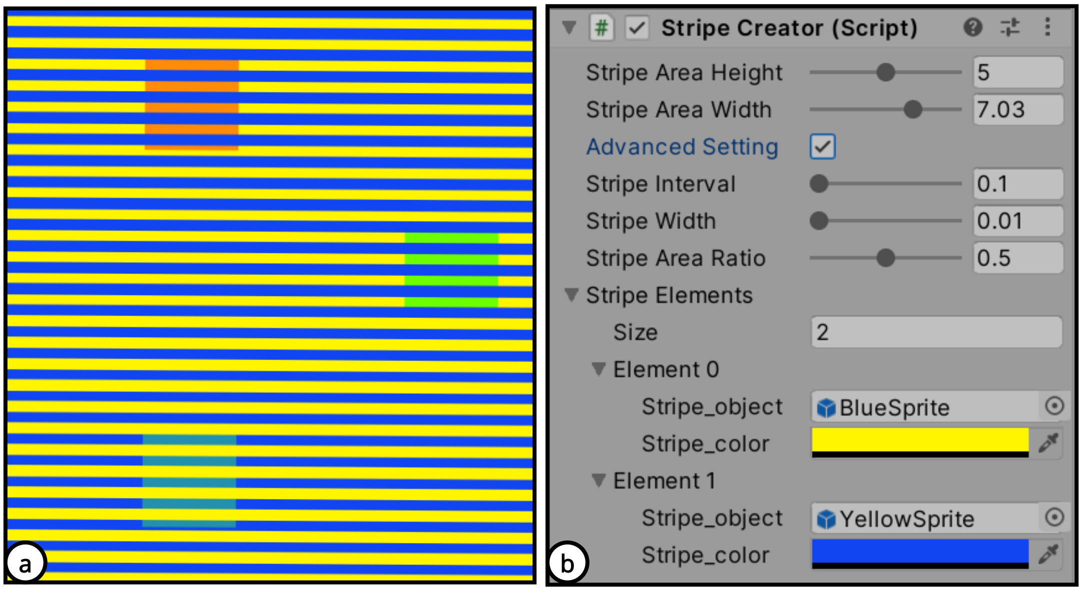


**Figure 16: The Munker–White illusion editor. (a) The preview generated by the editor. (b) The advanced settings for generating different layers of stripes to form the Munker–White illusion.**

materials that can be used for game objects once users click the generate button.

In the complementary law mode, users assign 2 colors as in Fig. 15b. Color One is one of the background color. Color Two is the color to be perceived differently in different backgrounds. The editor automatically calculates the rest color (e.g. the second background color) to maximize the perceived difference of the same color. While the perceived difference is maximized when the target's hue falls in the midpoint of the hue of two distinct surroundings [49], the target's saturation also affects the strength [14, 29, 49]. Users can thus use the saturation slider to control the strength.

## 5.3 Munker–White Illusion Editor

Fig. 16 shows our Munker–White illusion editor with the advanced settings. The editor, by default, generates 2 layers of stripes in 2 distinct colors (see Fig. 10 in a different perspective). This generation is based on varying strength of the Munker–White effect by stereoscopic viewing [57]. Users drag the strength slider to adjust the illusion in the basic settings. The strength is based on a prior work [11] where the interval and the width of the stripes are two major factors. The interval is the distance between the centers of two color lines. The width is the thickness of a single line. The strength increases when both the interval and the width are small and decreases vice versa. In the advanced settings, users can set them individually in the editor.

## 5.4 Poggendorff Illusion Editor

Fig. 17 shows our Poggendorff illusion editor. In practice, applying the Poggendorff illusion directly to a game object may not work as the object might not fit the theoretical requirement [62] for generating the illusion. The editor thus only allows users to modify the preview with parameters determined according to experiments done by Weintraub and Krantz [62]. They state that the illusion is stronger with a larger occluder and smaller intersecting angle of the transversal. Users can modify the size and color of the occluder, as well as the offset, angle, and appearance of lines in the advanced settings.

## 5.5 Hierarchical Motion Organization Editor

Fig. 18 shows our hierarchical motion organization editor. Users assign a center object and a number of surrounding objects. The editor then automatically animate these objects to generate the Rotating Circle Illusion [40] that makes the circular motion of the center object seem not circular. While Bertamini et al. [8] and Programming Visual Illusions for Everyone [7] have pointed out some possible factors, these are not yet parameterized. Instead, we offer two parameter settings to make the target object, which rotates in a circle, seem to be moving horizontally or vertically. We achieve this by controlling the surrounding objects to oscillate horizontally/vertically along the barycenter's horizontal/vertical axix. In the advanced settings, users can also adjust the rotating radius, the path of the orbital center, speed, and initial rotating angle for both surroundings and central.

## 5.6 Footsteps Illusion Editor

Fig. 19 shows our footsteps illusion editor. The editor automatically generates a background material with high luminance contrast stripes [3] using Unity Shader. The darker object seems to stop when the leading or trailing edge reaches the darker stripe, while the lighter object seems to keep moving. Users have to prepare their own pair of a darker object and a lighter object to move in the background. The appearance of game objects can be varied as long as they have a consistent luminance. Users adjust the advanced settings using the Unity Shader properties.

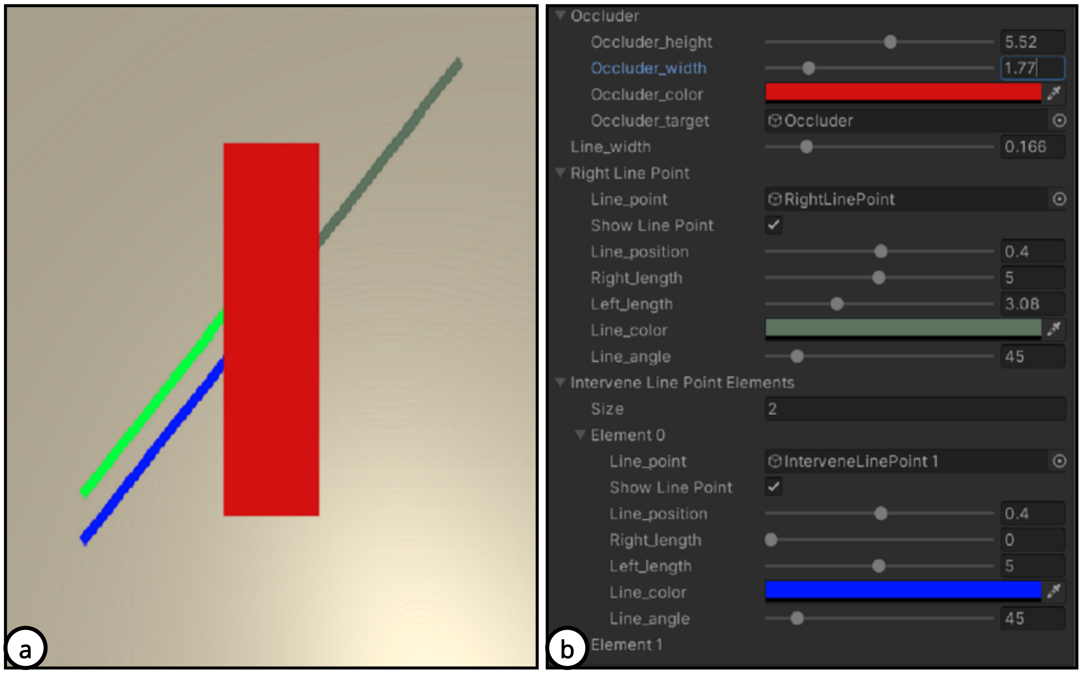


**Figure 17: The Poggendorff illusion editor. (a) The preview generated by the editor. The blue line is actually connected to the gray line. (b) The editing interface. Users can modify the size and color of the occluder, as well as the offset, angle, and appearance of lines in the advanced settings.**

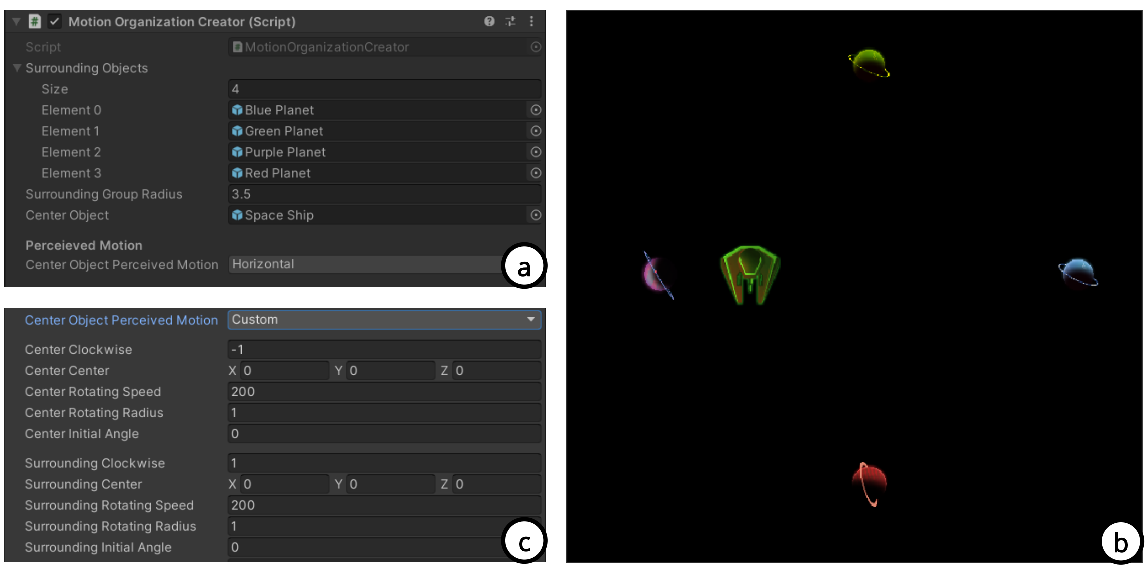


**Figure 18: The Hierarchical motion organization editor. (a) Users can assign the surrounding objects and the center object and select *horizontal* or *vertical* to change players' perceptions about the circular motion of the center object. (b) Users can see the preview in the game scene. (c) Users can also customize the motion of objects by adjusting the rotating speed and radius in the advanced settings.**

## 6 STUDY 1: ILLUSION GAME EVALUATION

In this study, our goal was to provide a preliminary understanding of illusion games, especially in terms of enjoyment and difficulty.

### 6.1 Participants

We recruited 12 participants, 7 females, aged from 20–23 ($M = 22.16, SD = 1.14$) from our institute. 6 participants mostly played

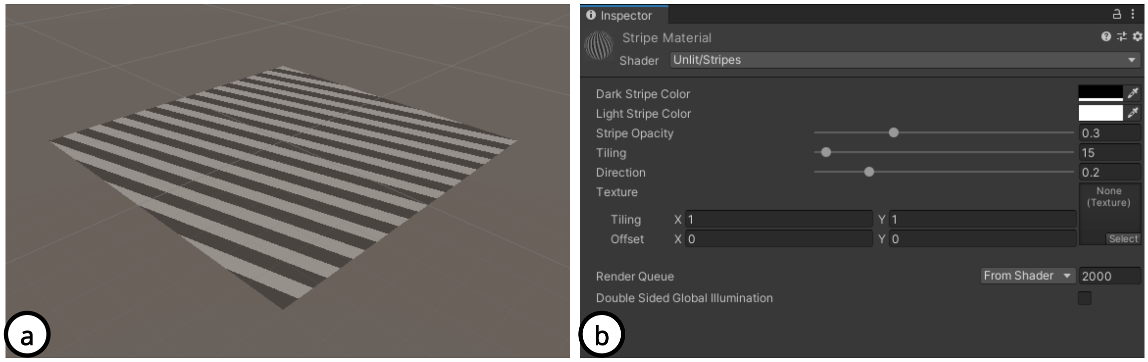


**Figure 19: The Footsteps illusion editor. (a) The preview generated by the editor. (b) The editing interface. We use Unity Shader properties as the interface to adjust the colors of the darker stripes and lighter stripes, the opacity, the repeat times, and the orientation of the stripes.**

puzzle games and role-playing games. 5 participants mostly played action games and first-person shooting games. 1 participant did not have a specific preference.

## 6.2 Task and Procedure

The task for the participants was to play 2 versions (*Baseline* and *Illusionized*) of 6 game types:

- G1: Gerritory-like game (Fig. 9);
- G2: Billiards (Fig. 11);
- G3: Taiko-No-Tatsujin-like game (Fig. 13);
- G4: NS-Shaft-like game (Fig. 4);
- G5: Road-Fighter-like game (Fig. 10);
- G6: Space-Invaders-like game (Fig. 12);

Each participant thus played 12 games in total.

We brought in 2 participants into our lab at a time. Each group played the two-player competitive games (G1 and G2) first, and then the single-player games (G3–G6), wherein the 2 versions were played consecutively. The game order was shuffled for each group. The version order was counterbalanced. For each game, participants practiced until they were familiar with the game mechanics. After playing 2 versions of a game type, the participants took a break. During the break time, the participants were asked to fill in our questionnaires and report their feedback. We asked the participants to answer on a 7-point Likert scale (1= not at all) about the games' difficulty, mental effort, and enjoyment that were extracted from Game Experience Questionnaire [10] and NASA-TLX [25]. The study ended when the participants finished playing all games. We selected these most relevant questions to prevent fatigue as our pilot study lasted for 1.5 hours. We paid each participant 15 dollars as compensation.

## 6.3 Result and Discussion

Our results are listed in Table 1. We used the nonparametric Mann-Whitney U test.

The difficulty and mental effort increased in the illusionized versions of the single-player games. This result supports our suggestion that the *misleading* mechanic increases the game's difficulty. However, there was no statistical difference in enjoyment between the illusionized and the baseline version. Our interpretation is that adding optical illusions to games could be an implicit way to increase difficulty without significantly affecting players' enjoyment. However, more investigation is required to verify the impact.

# 7 STUDY 2: WORKFLOW EVALUATION

The goal of this study is to understand how our proposed workflow help game designers to make their illusion games.

## 7.1 Participants

We recruited 6 game designers and 6 players (2 females, aged from 18–29 ($M = 23.75, SD = 2.74$)) from the Internet. All game designers had at least 3 years of experience (5 of them had more than 5 years) and had developed 5 games on average. All participants had played about 300 video games. They were very unfamiliar to familiar with optical illusions.

## 7.2 Task and Procedure

The task for the participants was to illusionize a game using our workflow. We brought in 1 participant at a time. In the beginning, we introduced our step-by-step workflow with the Fig. 1 and showed the instance of our NS-Shaft-like game (Fig. 4) to the participant. We then asked the participant to choose a commercial or their own video game and follow our workflow to illusionize it. We asked the participant to speak out loud about their thinking process, design considerations, and confusions about the workflow. The study ended when the participant was satisfied with their design. We collected the participant's feedback and asked them to evaluate our workflow with the System Usability Scale (SUS) [60]. We did not give a time limit while all the participants finished the study in an hour. We paid each participant 15 dollars as compensation.

## 7.3 Result and Discussion

We collected not only the modifications of existing games but also brand new game designs created by the participants. These games' genres varied from first-person shooter games, rhythm games, puzzle games to action games.

All participants thought that their illusionizations added variety, difficulty, and surprise to the original games. All participants also stated that our illusion database with hierarchical tags benefited them the most as it provided a good number of illusions and a systematic search method. P6, P7, and P8 said that a concrete process was helpful to stay on track. The average SUS score of the given workflow was 73.3 (*Good* in the adjective rating).

There were some contradicting comments. Most participants appreciated starting by choosing an object, while P3 and P5 said their thoughts were limited. P2 stated the step *Selecting an Illusion Mechanic* was not helpful for those who used *Effects* tags to search illusion since they had concrete thoughts about how to use the illusions. Half of the participants asked for more examples and details of illusion mechanics since they only could come up with *Misleading*. Meanwhile, other participants stated that these examples limited their imagination.

We also learned more about whether the reveal step should be involved in the workflow. P11 said the reveal step reminded him to think about the meaning of applying illusions. Without reveal, the illusion might be an annoying feature (such as Rhythm Doctor [1]) or an Easter egg. If figuring out the illusion was the core objective, the explicit reveal might not be necessary.

However, there were 3 participants doubting the applicability of our workflow. P10 and P11 reported their reason as there was no back arrow in the given workflow (Fig. 1). They commented that the workflow should suggest iterations between *Choosing an Object*, *Searching for a Matching Illusion* and *Selecting an Illusion Mechanics*. We thus revised our workflow to the one as in Fig. 3 and clarified it in the next study (Section 8).

## 8 STUDY 3: IN-DEPTH INTERVIEW

Since the participants in Study 2 did not actually implement their illusion game, the goal of this study was to further understand how the remaining workflow, in particular our illusion editors, facilitates the implementation of illusion games.

### 8.1 Participants

3 experienced game designers who were also developers in Study 2 (all males, aged 23, 27, and 29) participated in this study. They all had at least 3-year experience in game development and had made at least 4 games. P2 was a little familiar with illusions, while P1 and P3 were not familiar at all.

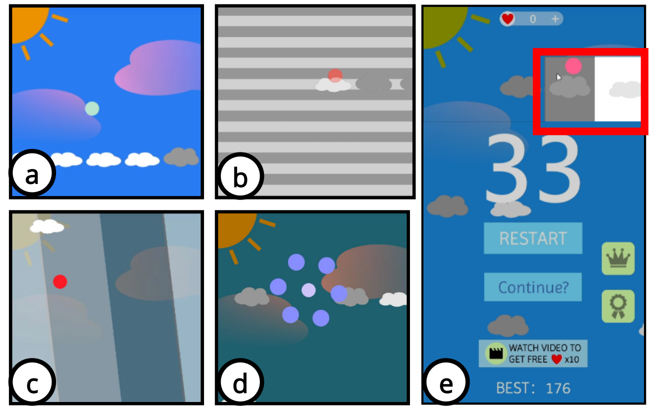


**Figure 20: The game *Dropping Dot*. (a) Players tilt their device to steer the dropping dot. They score points by passing the white clouds while dodging the dark clouds. Following our workflow, the designer made the following modifications using 3 kinds of illusions. (b) Using the footsteps illusion to *mislead* players about the cloud's movement. (c) Using the simultaneous color contrast illusion to *mislead* players about the colors of the clouds. (d) Using the dynamic Ebbinghaus illusion [39] to *mislead* players about the size of the dot, making it seem smaller than the gap between two dark clouds or seem larger to cross multiple white clouds. (e) If players are game-over, they have to figure out a way to erase the fact that the dot hits a dark cloud. The way is to drag the prompting gray-white background to the right position to make the dark cloud, and the white cloud seems the same color. Players thus revive since the dot seem to hit the "white" cloud instead now.**

### 8.2 Task and Procedure

The task for the participants was to implement the illusion games they came up with in Study 2 in 2 weeks. We provided the participants with our workflow document, illusion database, and 6 illusion Unity editors. They were allowed to modify their design in Study 2. The participants worked whenever and wherever they wanted but were asked to record their screens and take (voice) notes during the development. After two weeks, we conducted an in-depth interview with each participant asking 2 major questions:

| | Difficulty | | | | | Mental Effort | | | | | Enjoyment | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *P* | Baseline | | Illusionized | | *P* | Baseline | | Illusionized | | *P* | Baseline | | Illusionized | |
| | | *M* | *SD* | *M* | *SD* | | *M* | *SD* | *M* | *SD* | | *M* | *SD* | *M* | *SD* |
| G1 | 0.1860 | 4.33 | 1.31 | 5.08 | 1.55 | 0.5144 | 4.17 | 1.34 | 4.5 | 1.32 | 0.7832 | 5.42 | 1.32 | 5.33 | 1.11 |
| G2 | 0.2583 | 4.67 | 1.65 | 5.33 | 1.60 | 0.1231 | 4.08 | 1.55 | 5.08 | 1.38 | 0.3044 | 4.83 | 1.91 | 5.33 | 1.80 |
| G3 | **0.0281** | 5.00 | 1.15 | 6.17 | 1.07 | **0.0332** | 4.42 | 1.19 | 5.58 | 1.26 | 0.2734 | 5.58 | 0.86 | 5.08 | 1.04 |
| G4 | **0.0205** | 4.50 | 1.26 | 5.67 | 1.03 | **0.0231** | 3.50 | 1.61 | 5.08 | 1.38 | 0.2703 | 5.00 | 1.08 | 5.50 | 1.44 |
| G5 | **0.0030** | 4.58 | 1.38 | 6.42 | 1.04 | **0.0357** | 4.33 | 1.75 | 5.75 | 1.69 | 0.4183 | 5.25 | 1.23 | 4.58 | 1.85 |
| G6 | **0.0455** | 3.42 | 1.71 | 4.75 | 1.42 | **0.0383** | 3.33 | 1.25 | 4.50 | 1.19 | 0.5247 | 4.92 | 1.38 | 5.33 | 1.03 |

**Table 1: The result of difficulty, mental effort and enjoyment scores between the baseline and the illusionized versions of our 6 examples.**

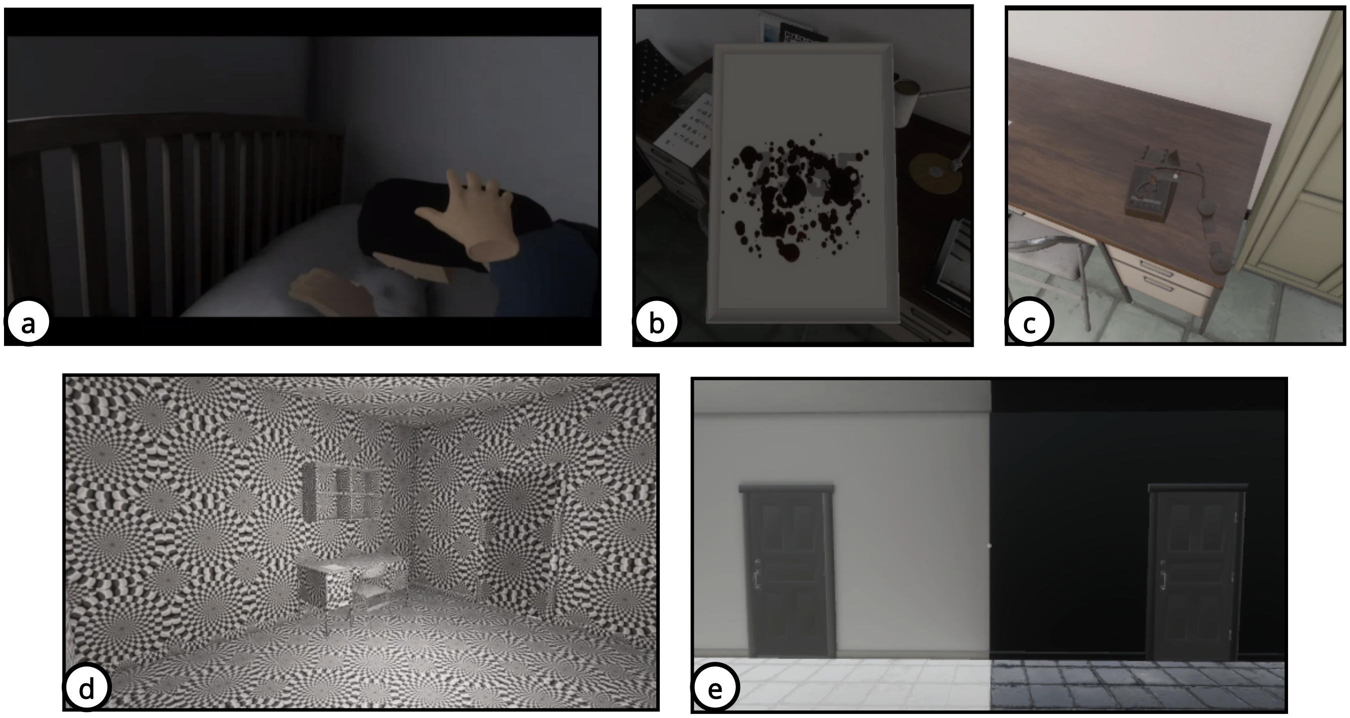

**Figure 21: (a) *Defect* is a VR room escape game that helps empathize people with depression. After following our workflow, the designer expanded his game with 4 more stages. (b) Players see an incomprehensible hint in the first room. By spilling coffee on the hint, they now see the password: 7995 by the illusion of Amodal completion [37]. (c) Players are asked to call for help with a telephone in a room with no telephone. The telephone only appears when they look at the pieces of the phone at the right angle (Modal Completion [27]). (d) Players then go into a room full of rotating snake illusion [12], making players feel that all the walls in this room seem to be rotating. (e) Players are asked to choose a brighter door to go out. However, the two doors have, in fact, the same color because of the simultaneous color contrast, hinting that the person with depression in the game gets stuck because they do not perceive the reality.**

(1) How our workflow, database, and illusion editors help them, and (2) what problems they encountered during the implementation. Each interview lasted about 3 hours. We paid each participant 200 dollars as the 2-week compensation.

## 8.3 Result and Discussion

All designers successfully implemented their illusion games with more features. P1 modified his game, *Dropping Dot* (Fig. 20), with 4 game mechanics that used 3 illusions. P2 extended his game, *Defect* (Fig. 21), a room escape VR experience that helps empathize people with depression, with 4 illusion stages. P3 created three instances that used the functionalities of illusions in AR experiences (Fig. 22). We describe all these games in the captions of the figures. With their illusion games, we observed that (1) one illusion can be used in different situations and even in different game genres, and (2) more than one illusion can be applied in one game.

With regard to our workflow, as we clarified the final version of the workflow (Fig. 3) during the interview, they all agreed with us that their 2-week development validated the final workflow. P1, however, suggested that we could retain our step-by-step workflow (Fig. 1) since it would be easier to start with. P3 pointed out the two starting points of our final workflow: (1) applying illusions to existing games by choosing game objects first, and (2) creating a brand-new illusion game by selecting illusions first as the core gameplay. P2 further supported creating new games instead of applying illusions to existing games for the reason that designers usually do not want to change their core gameplay during their development.

All developers reported that they were inspired most by our illusion database. P1 and P2 stated that the tags helped to find their desired illusions most of the time. P1 and P3 suggested that we should label illusions supported by our editors and describe the known limitations of the illusions. They all found it difficult to use illusions that were not supported by our editors.

With regard to our illusion editors, P1 used our simultaneous color contrast illusion editor and our footsteps illusion editor, P2 used only our color contrast illusion editor, and P3 did not use any. P1 spent 15 minutes manually creating the grey-white background for the revival (Fig. 20d) but spent only 5 minutes achieving the same background using the direction law mode (Figure 15a) in our simultaneous color contrast illusion editor. P1 spent 10 minutes manually making the striped background for the footsteps illusion (Fig. 20b) but spent merely 2 minutes achieving the same result using our footsteps illusion editor. Our editor sped up 3 and 5 times


**Figure 22: One of the designers used scanimation [26] illusion for augmented reality (AR) games. With the help of AR, players see moving stripes over the following images in the mobile app, making them animated. (a) The image has a sentence: "Every epic adventure begins with a single step," which can only be seen using the AR app. (b) The stick figure seems to be exploding a bomb but is actually inflating heart balloons in the AR App. (c) The static fire is actually flaming in the AR app.**

in these two cases as it saved time to find the colors to produce the illusion while matching their visual theme. P2 used the complementary law interface (Fig. 15b) to make the last stage's background materials (Fig. 21e). P2 was surprised that our editor worked even for 3D materials since our editor only gives a 2D color preview. Two designers were satisfied and delighted with the illusion customization for their own games. On the flip side, they both urged us to provide complete documentation for our editors.

We conclude from the result in this study that our workflow, database, and illusion editors help game designers to create, iterate and implement illusion games without illusion expertise.

## 9 LIMITATIONS AND FUTURE WORK

We noticed several limitations during the studies, which also inspired us to see from a higher level to improve our workflow in the future.

### 9.1 Improving Design Guidelines

In Study 2, while some participants did quickly iterate with our illusion mechanics and reveal techniques, some participants have concerns that these guidelines were non-exhaustive or these guidelines were restricting their design space. We see more opportunities in developing these guidelines to uncover this design space.

### 9.2 Understanding More Illusions and Games

While we cannot conclude that every type of game is suitable for applying illusions, we have shown several combinations from our examples and study results that cover a wide range of games and illusions. For instance, some designers questioned about the feasibility of applying 2D illusions to 3D games. Since 2D is a part of 3D, most 2D illusions potentially can be applied to 3D games, as P2's game shows. While our illusion database facilitates searching for a valid combination, i.e., making an illusion game, the validation has to be done with a good understanding of both games and illusions. On the same note, there are still illusions that are not parameterized or interfaced. We see more and deeper studies are still required in researching optical illusions and more engineering effort has to be put into interfacing this knowledge.

### 9.3 Recommending illusions

Our current database dumps a query result in default order. This may influence the user's priority in iterating their designs. In the future, we will try to detect and recommend illusions based on games' visual properties, genres, and rules automatically by further parameterizing these factors and including statistics of existing illusion games. As a whole, we aim to build a plug-and-play illusion recommendation system embedded in the game engine in a continuum that covers our current workflow. We also see opportunities in expanding the system to other sensory illusions and even cross-modal illusions.

## 10 CONCLUSION

In this paper, we have presented how to illusionize a video game using our workflow. We have shown that our workflow helps game designers to create or iterate their game designs with illusions with our study results. Our illusion database helps game designers to quickly find a desired illusion. We also have introduced our 6 specific illusion editors to help game developers integrate illusions in their games. With our workflow, illusion database, and illusion editors, we hope to see more synergy between the large body of optical illusions and games.

## ACKNOWLEDGMENTS

This work was supported by the Ministry of Science and Technology in Taiwan (MOST 110-2636-E-002 -010). We thank Pei-Chih (FlaMvp) Wang for drawing tag icons in our illusion database. We thank our participants for facilitating our workflow and tags in our illusion database. We thank Guan-Wei Chen, Ching-Yu Hsieh, and Tz-Yang Chao for actually building the illusion games with us.

## REFERENCES

[1] 7th Beat Games. 2021. Rhythm Doctor. Retrieved March 30, 2021 from https://rhythmdr.com/
[2] Stuart Anstis. 2001. Footsteps and inchworms: Illusions show that contrast affects apparent speed. *Perception* 30, 7 (2001), 785–794.
[3] Stuart Anstis. 2004. Factors affecting footsteps: Contrast can change the apparent speed, amplitude and direction of motion. *Vision Research* 44, 18 (2004), 2171–2178.
[4] Aristotle, David Ross, Erwin Wentworth Webster, E. S. Forster, J. A. Smith, J. I. Beare, G. R. T. Ross, and J. F. Dobson. 1931. *The works of Aristotle. Vol. 3, Vol. 3,*. Oxford University Press, Oxford.
[5] Michael Bach. 2021. https://michaelbach.de/ot/
[6] Michael Bach and Ch M Poloschek. 2006. Optical illusions. *Adv Clin Neurosci Rehabil* 6, 2 (2006), 20–21.
[7] Marco Bertamini. 2017. *Programming visual illusions for everyone.* Vol. 2. Springer.
[8] Marco Bertamini and Andrew Irving. 2018. Once Upon A Time. http://illusionoftheyear.com/2018/10/once-upon-a-time/
[9] Team BlueBacon. 2019. ORTHOISO. https://orthoiso.com/
[10] Jeanne H Brockmyer, Christine M Fox, Kathleen A Curtiss, Evan McBroom, Kimberly M Burkhart, and Jacquelyn N Pidruzny. 2009. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of experimental social psychology* 45, 4 (2009), 624–634.
[11] Ivan Budimir, Nikola Mrvac, and Mile Matijević. 2015. The influence of the thickness of the grid in Munker-White effect. *Technical Gazette* 22, 2 (2015), 425–430.
[12] Bevil R Conway, Akiyoshi Kitaoka, Arash Yazdanbakhsh, Christopher C Pack, and Margaret S Livingstone. 2005. Neural basis for a powerful static motion illusion. *Journal of Neuroscience* 25, 23 (2005), 5651–5656.
[13] Sybille Ebert-Schifferer and Sybille Ebert-Schifferer. 2002. *Deceptions and illusions: five centuries of trompe l'oeil painting.* National Gallery of Art Washington, DC.
[14] Vebjørn Ekroll and Franz Faul. 2012. New laws of simultaneous contrast? *Seeing and Perceiving* 25, 2 (2012), 107–141.
[15] Vebjørn Ekroll and Franz Faul. 2013. Transparency perception: The key to understanding simultaneous color contrast. *JOSA A* 30, 3 (2013), 342–352.
[16] Illam Software Entertainment. 2017. Mirage: Illusions. https://illamsoft.com/
[17] Sony Interactive Entertainment. 2008. Echochrome. https://www.jp.playstation.com/scej/title/mugen/
[18] Ivanovich Games. 2015. PERFECT ANGLE: The puzzle game based on optical illusions. http://www.ivanovichgames.com/web/perfect-angle-zen-vr-ar-and-no-vr/
[19] Ivanovich Games. 2018. Sea of memories. http://www.ivanovichgames.com/web/optical-illusions-reach-virtual-reality-with-sea-of-memories/
[20] Pillow Castle Games. 2019. Superliminal. Retrieved June 21, 2020 from http://www.pillowcastlegames.com
[21] Ustwo Games. 2014. Mounument Valley 1 & 2. Retrieved June 21, 2020 from https://www.monumentvalleygame.com/mv2
[22] Party Goose. 2020. Gerritory: A party game. https://partygoosestudio.wixsite.com/gerritorythegame
[23] Richard L Gregory. 1997. Visual illusions classified. *Trends in cognitive sciences* 1, 5 (1997), 190–194.
[24] Hong Guo, Hallvard Trætteberg, Alf Inge Wang, and Shang Gao. 2015. A workflow for model driven game development. In *2015 IEEE 19th International Enterprise Distributed Object Computing Conference.* IEEE, 94–103.
[25] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology.* Vol. 52. Elsevier, 139–183.
[26] Henry V Hopwood. 1915. *Hopwood's Living Pictures: Their History, Photo-production, and Practical Working.* Hatton Press.
[27] Gaetano Kanizsa. 1955. Margini quasi-percettivi in campi con stimolazione omogenea. *Rivista di psicologia* 49, 1 (1955), 7–30.
[28] August Kirschmann. 1890. *Ueber die quantitativen Verhältnisse des simultanen Helligkeits-und Farben-Contrastes.*
[29] August Kirschmann. 1892. Some effects of contrast. *The American Journal of Psychology* 4, 4 (1892), 542–557.
[30] Akiyoshi Kitaoka and Stuart Anstis. 2021. A review of the footsteps illusion. *Journal of Illusion* 2 (2021).
[31] Konami. 1984. ROAD FIGHTER. https://www.konami.com/games/jp/ja/products/dl_wiiu_roadfighter_vc/
[32] Akihiko Kusanagi. 1990. NAGI-P SOFT: NS-SHAFT Information. https://www.nagi-p.com/v1/eng/nsshaft.html
[33] R Beau Lotto and Dale Purves. 2000. An empirical explanation of color contrast. *Proceedings of the National Academy of Sciences* 97, 23 (2000), 12834–12839.
[34] Noxus Ltd. 2016. Miika - Illusion Puzzle Games. https://nox.us/
[35] Fiona Macpherson and Keith Wilson. 2021. The Illusions Index. https://www.illusionsindex.org/
[36] Sebastian Marwecki, Andrew D Wilson, Eyal Ofek, Mar Gonzalez Franco, and Christian Holz. 2019. Mise-unseen: Using eye tracking to hide virtual reality scene changes in plain sight. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology.* 777–789.
[37] A Michotte and L Burke. 1951. Une nouvelle enigme dans la psychologie de la perception: Le'donne amodal'dans l'experience sensorielle. *Actes du XIII Congrés Internationale de Psychologie* (1951), 179–180.
[38] Shigeru Miyamoto. 1992. Mario Kart. Retrieved April 6, 2021 from https://www.nintendo.co.jp/n02/shvc/mk/index.html
[39] Ryan EB Mruczek, Christopher D Blair, Lars Strother, and Gideon P Caplovitz. 2015. The Dynamic Ebbinghaus: motion dynamics greatly enhance the classic contextual size illusion. *Frontiers in human neuroscience* 9 (2015), 77.
[40] Ryan E.B. Mruczek and Gideon Paul Caplovitz. 2019. The Rotating Circles Illusion. http://illusionoftheyear.com/2019/12/the-rotating-circles-illusion/
[41] Juergen Musil, Angelika Schweda, Dietmar Winkler, and Stefan Biffl. 2010. Improving video game development: Facilitating heterogeneous team collaboration through flexible software processes. In *European conference on software process improvement.* Springer, 83–94.
[42] Kizashi Nakano, Daichi Horita, Nobuchika Sakata, Kiyoshi Kiyokawa, Keiji Yanai, and Takuji Narumi. 2019. Deeptaste: Augmented reality gustatory manipulation with gan-based real-time food-to-food translation. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).* IEEE, 212–223.
[43] Jacques Ninio. 2014. Geometrical illusions are not always where you think they are: a review of some classical and less classical illusions, and ways to describe them. *Frontiers in human neuroscience* 8 (2014), 856.
[44] Tomohiro Nishikado. 1978. SPACE INVADERS. https://spaceinvaders.jp/
[45] Tadasu Oyama. 1975. Determinants of the Zöllner illusion. *Psychological research* 37, 3 (1975), 261–280.
[46] Lionel S Penrose and Roger Penrose. 1958. Impossible objects: a special type of visual illusion. *British Journal of Psychology* (1958).
[47] Cambridge University Press. 1995. Cambridge Dictionaries Online. https://dictionary.cambridge.org/us/about.html
[48] Andreas Pusch and Anatole Lécuyer. 2011. Pseudo-haptics: from the theoretical foundations to practical system design guidelines. In *Proceedings of the 13th international conference on multimodal interfaces.* 57–64.
[49] Sivalogeswaran Ratnasingam and Barton L Anderson. 2017. What predicts the strength of simultaneous color contrast? *Journal of vision* 17, 2 (2017), 13–13.
[50] RealityBending. 2020. Pyllusion: A Parametric Framework to Generate Visual Illusions using Python. https://github.com/RealityBending/Pyllusion
[51] Yushi Sato, Naruki Tanabe, Kohei Morita, Takefumi Hiraki, Parinya Punpongsanon, Haruka Matsukura, Daisuke Iwai, and Kosuke Sato. 2019. Pseudo-haptic feedback in a projected virtual hand for tactile perception of textures. In *Proc. IEEE World Haptics Conf.* 1–2.
[52] Joseph Saulter. 2007. *Introduction to video game design and development.* McGraw-Hill New York.
[53] Neural Correlate Society. 2005. Best Illusion of the Year Contest. http://illusionoftheyear.com

[54] Bandai Namco Studios. 2001. Taiko no Tatsujin. https://taiko-ch.net/index.php
[55] Lykke Studios. 2017. AMON. https://www.lykkestudios.com/
[56] Nan-Ching Tai. 2016. Investigation on light effect on spatial illusion resulting from forced perspective. *Computer-Aided Design and Applications* 13, 2 (2016), 221–227.
[57] Raiten Taya, Walter H Ehrenstein, and C Richard Cavonius. 1995. Varying the strength of the Munker—White effect by stereoscopic viewing. *Perception* 24, 6 (1995), 685–694.
[58] Unity Technologies. 2005. Unity. Retrieved Augest 20, 2020 from https://unity.com/
[59] Ubisoft. 2020. Assassin's Creed:Valhalla. https://www.ubisoft.com/zh-tw/game/assassins-creed/valhalla
[60] usabiliTEST. [n.d.]. System Usability Scale online with analytics. https://www.usabilitest.com/system-usability-scale
[61] Nicholas Wade. 2015. *Art and illusionists*. Vol. 1. Springer.
[62] Daniel J Weintraub and David H Krantz. 1971. The Poggendorff illusion: Amputations, rotations, and other perturbations. *Perception & Psychophysics* 10, 4 (1971), 257–264.
[63] Peter Zackariasson, Alexander Styhre, and Timothy L Wilson. 2006. Phronesis and creativity: Knowledge work in video game development. *Creativity and Innovation Management* 15, 4 (2006), 419–429.
[64] Friedrich Zöllner. 1860. Ueber eine neue Art von Pseudoskopie und ihre Beziehungen zu den von Plateau und Oppel beschriebenen Bewegungsphänomenen. *Annalen der Physik* 186, 7 (1860), 500–523.