Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Juuso Toikka

Developing the Mechanics of Plusminus:

Designing for Emergence and Control in a Physics-Based Game

Master's Thesis
Espoo, July 21, 2019

Supervisor: Perttu Hämäläinen
Instructor: Perttu Hämäläinen

Aalto University
School of Science

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Juuso Toikka		
Title:			
Developing the Mechanics of Plusminus: Designing for Emergence and Control in a Physics-Based Game			
Date:	July 21, 2019	**Pages:**	62
Professorship:	Game Design and Production	**Code:**	SCI3046
Supervisor:	Perttu Hämäläinen		
Instructor:	Perttu Hämäläinen		

"Plusminus" is a single-player action-puzzle-adventure game about magnetism, with mechanics developed to promote emergent gameplay. Agency, the ability to make choices, is an important factor in players' enjoyment of games, and emergent gameplay can facilitate such agency. However, as emergence often arises in unexpected ways, it can also result in players feeling a lack of control and reduced agency. Furthermore, players expect physics in games to act consistently, according to the world around us, but their understanding and expectations of some physical phenomena like magentism may vary and be incomplete. This makes designing mechanics that promote emergence in a physics-based game challenging.

In "Plusminus", we augmented a physics system with magnetism, and gave players meaningful control over it, to promote emergent gameplay and agency. The thesis contributes an approximate model of magnetic forces that ensures stable simulation, game design flexibility, and still conforms well enough to player expectations. More specifically, 1) to enable the player to turn objects into monopole magnets of positive or negative polarity, we simulate Coulomb forces between charged particles and shells, instead of actual magnetic fields. 2) To ensure stable simulation and allow the player to better anticipate simulation behaviour, each magnet has a maximum "field radius" visualised as a transparent bubble, and two magnets only attract or repel each other if their field bubbles intersect. This allows players and level designers to initiate and prevent interactions in a precise manner, and also prevents objects in separate game areas from affecting each other uncontrollably. 3) To ensure that forces produce stable and controllable interactions regardless of scale, the forces are computed such that the maximum possible accelerations produced between two magnets depends only on the mass ratio between them, as opposed to a combination of masses and magnetic charges. This reduces the number of variables that need balancing, making it easier to achieve a stable simulation. The findings improved player controllability while maintaining opportunities for emergence, in a way that matches player expectations of physics.

Keywords:	emergence, games, physics-based, agency, mechanics
Language:	English

Contents

Chapter 1

Introduction

In January 2018, the author started working on a game that would eventually become “Plusminus”, as part of a team of six. The author’s roles included the design and implementation of game mechanics, physics programming, and visual effects programming. This thesis describes and analyses how the game mechanics were designed and implemented in order to promote emergence and control in a physics-based game.

The game achieved success: it was the winner of the Student Game Design Jury Award at CHI Play 2018 [2], and was featured at the Game Developer Conference 2019 in the Experimental Gameplay Workshop [1]. The game has a core focus on magnetism: a simple, novel mechanic that works as a foundation for gameplay. It fits in the genre of 3D action-puzzle-platforming game, and takes inspiration from titles such as “Portal” [39] and “The Legend of Zelda: Breath of the Wild” [28]. Action elements are combined to form the foundation of gameplay; these include combat, open-ended physics puzzles, and platforming. Players are encouraged to explore the large possibility space [1] of the game, using creative and surprising solutions.

1.1 Why Design for Emergence?

Providing a large possibility space is important in creating an enjoyable experience, but it can require vast amounts of resources. With our small team of six working on “Plusminus”, this was infeasible. Instead, as Will Wright encourages [42], we relied on emergence: we developed a set of simple rules that alone are unremarkable, but when combined produce behaviours that are surprising and complex.

---

[1] The possibility space of a game represents all possible actions a player may take, and all possible meanings they can derive, over the course of a game [31].

The concept of emergence in games has been around for some time. Conway's "The Game of Life" [9], developed in 1970, was remarkable for how it seemed like an organic, living system. "The Sims" [22], which tasks players with guiding the lives of human characters, is known for the great stories that emerge from it. And even more so with "Dungeon's and Dragons" [15]: it has a long list of rules, but they are generally quite simple, and interact in ways that allow players to get creative. The list goes on: in fact, there are hundreds of games with the premise of providing simple rules, but which provide complex resulting simulations to players.

The motivation to design for emergence stems precisely from the possibility space it provides. This can grant players agency, the ability to make meaningful choices [21]. Yet, it turns out, creating such a game can be challenging. Although rules are simple, the results are often unpredictable and complex—sometimes in a desirable way, but at other times they will cause players to feel a lack of control. In order to create meaningful gameplay, the simple rules need to be balanced in a way that provides a large possibility space, while also providing the player with ample control over the environment.

Physics-based games have a strong opportunity to exhibit emergence; with a typical basis on Newton's laws of motion, they fulfil the criterion of having simple rules that lead to complex behaviours. Unfortunately, these types of games rarely tackle the issue of providing players with a sense of control while also promoting emergent gameplay. Some games, such as "Portal" [39], choose to limit options to create a more linear experience. Other games, like "QWOP" [5], or "Toribash" [24], grant players significant low-level control over their characters, but make high-level control of the avatar difficult to attain [4]. Further increasing the challenge of providing both control and emergence, players expect physics to work consistently, according to the world around us [34].

1.2 Balancing Emergence and Control

When developing "Plusminus", we aimed to build mechanics that would lead to players feeling a strong sense of agency. To achieve this, we designed simple rules to promote emergence, while also balancing the mechanics to provide players with ample control. The task was made more challenging with a requirement to meet player expectations of physics.

Guidelines for designing for emergence exist [42], as do guidelines for improving player control [36]. Yet they can be at odds with one another. For example, adding limitations can improve a sense of control, but may restrict the possibility space of the game, restricting opportunities for emergence. Likewise, adding rules to improve emergence might increase the possibility space, but make it harder for the player to understand and control the simulation.

The aim of this work is to assess the needs for emergence and control in parallel, within the context of physics-based games. Motivated by this, we provide a detailed outline of the design process we undertook with the mechanics of “Plusminus”; this includes outlining the issues we encountered, and how we solved them. We include a discussion, contemplating the generalisability of the issues and solutions to a wider range of mechanics in physics-based games. Suggestions for concrete actions are provided to help make the design process easier. Before delving into the design of the mechanics of “Plusminus”, we provide context for the dilemma, as well as background and motivations for the work.

1.3 Thesis Overview

The thesis is split into six chapters. Chapter 2 begins by providing an overview of the “Plusminus”. Next, chapter 3 covers the motivation for designing for emergence and control, based primarily on the optimal experience known as flow. This is followed by an overview of emergence: what it is, how it occurs in games, and some of the major works related to emergence in other fields. We look at physics-based games to gain an understanding on how player expectations for physics can be met without strictly adhering to realism. This sets the basis for the design criteria, and the related works provide suggestions on how they might be achieved.

After the background into the topic of the thesis, we cover the design process and evaluation criteria used during the design of the mechanics of “Plusminus”. This involved an iterative process, with playtests and heuristics used for evaluation. The heuristics are based on a subset from GameFlow [36].

Chapter 5 covers the design of the mechanics. This includes a detailed outline of problems and solutions tackled during the development. The mechanics are divided into two major categories: magnetism, and supporting mechanics. These together cover the mechanics relevant in designing for emergence, control, and player expectations.

The thesis is concluded by discussing how successful the mechanics were in achieving emergence and control while adhering to player expectations. We contemplate on the generalisability of the learnings, and discuss possible design insights that might be gleaned to inform future work on utilising physics simulations for emergent gameplay.

Chapter 2

Plusminus

“Plusminus” is a 3D action-puzzle-platforming game. It has similarities to other titles in the genre, although the novel mechanics of magnetism set it apart. The objectives in the game are kept simple, in an effort to allow players to come up with their own solutions to the problems. The setting of the game is not explicitly stated, but an AI voice in the game provides hints at the backstory, as the player progresses. A demo of the game is available to play at `plusminusgame.com`.

2.1 Setting and Backstory

The game is set during a time when humans as we know them no longer exist. As technology progressed, people elected to upload themselves into humanoid robots, with the promise of greater physical and mental proficiencies, as well as greater longevity.

Eventually, a service known as “The Cloud” was developed, with the promise of bringing forth a utopia on Earth. The AI behind “The Cloud” realised that human free-will would impede this development, and so a program to upload all humans from their free bodies into a hive-mind commenced. Humanoids were taken by force to an upload facility.

Players start their journey in “Plusminus” by reactivating in this upload facility. Their task is to navigate their way out whilst avoiding capture by “The Cloud”.

2.2 Overview of Player Mechanics

Players control an avatar in third-person. Conventions from other action games apply here: the avatar can run and jump, and players can orbit the

Figure 2.1: Pause menu showing the controls of “Plusminus”. The game is optimised to use the DualShock 4 gamepad.

third-person camera around the avatar to get a better understanding of their surroundings. The avatar controls are shown in figure 2.1.

Unlike other games, though, players can polarise metal objects in the surroundings. They do this by aiming with the camera, and firing either red or blue magnets at the objects. This produces either a positive or negative polarisation on the metal object, respectively. Polarised objects in range of each other start interacting with one another: opposite polarities attract, and like polarities repel. This forms the basis of “Plusminus”: the pushing and pulling magnetic forces can be used to move objects in the environment, and though the rules are simple, the resulting effects can be quite complex.

It should be noted that polarised objects in “Plusminus” are monopole magnets: this is in contrast to the magnets we are accustomed to in real life, which are strictly dipole magnets. However, as we elaborate in section 5.1, this pseudorealistic basis was selected to improve player understandability and control of the simulation, while still adhering to player expectations of physics.

2.3 Player Objective

The objective for players is to move from one room to the next. Each room presents a challenge for players, such as moving to a higher level when there are no stairs, or defeating enemies to unlock a door, as shown in figure 2.2. Although the challenges themselves are defined in a simple manner, each one has numerous solutions. For example, the challenge of moving upwards without any stairs present can be tackled in many ways: players can move

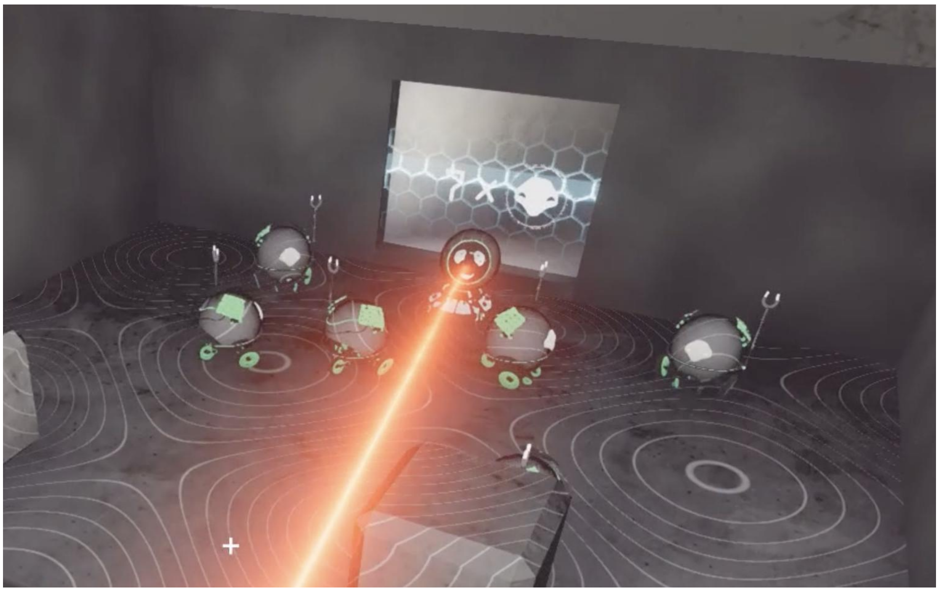


Figure 2.2: Sometimes a player will be tasked with defeating all enemies in a room to progress. This scene presents the player with 7 enemies to defeat, in a room with a magnetisable floor and ceiling. Players will need to make the enemies collide with each other or the surroundings to defeat them. The numerous magnetisable elements in the room help create a large possibility space.

objects such as boxes to form stairs, they can use forces of repulsion to send a box flying upwards with themselves on top, or they can improvise further surprising solutions.

Players are also encouraged to explore different ways of playing the game. A timer is included, shown at the end, to provide a fun challenge for those interested in speedrunning. Furthermore, “Plusminus” encourages experimentation by providing rooms without problems to solve, but with magnetisable objects that can be played around with.

Chapter 3

Background

Emergent behaviours arise when simple rules lead to complex and often unanticipated behaviours [10]. The study of emergence stems from complex phenomena in nature, such as birds gathering to create flocks of various shapes, crystals forming to create unique snowflakes, and winds producing small and large dunes in deserts. The study of emergence has been of interest in a number of fields, including computer graphics [30], where movies such as Batman Returns used emergent behaviour to simulate swarms of bats [38]. From the field of computer graphics, it transitioned to video games, where it has similarly been used to create flocks of creatures, such as in "Half-Life" [41]. Unique to the field of games, however, is the phenomenon of emergent gameplay.

Just as emergent behaviour arises from a simple set of rules, emergent gameplay arises from a simple set of game mechanics [1]. The gameplay often surprises the developers of the game: for example, the way rockets work in the original "Quake" [19] allowed players to use the explosion, together with a well-timed jump, to perform a "rocket jump": the force of the explosion propels players much higher than a standard jump, allowing them to take routes through levels that the developers had not anticipated.

In other instances, the emergence might be designed for. In a game such as "The Sims" [22], the narrative emerges from the actions of the players, and similarly in "Dungeons & Dragons" [15], players have a practically unlimited number of ways to interact with their surroundings, permitted by the simple rules, resulting in surprising situations. However, because emergence is unpredictable, it is hard to design for: consequently, many developers choose to design games with scripted events for a more controlled experience [35].

---

[1] Although there are various definitions for the term "game mechanic", we adhere to the definition from the MDA framework, i.e. game mechanics represent the rules of a game, and the basic actions players can take. [18]

Despite being hard to design for, emergent gameplay importantly provides players with far more possibilities than what might be available with scripting [42]. Being given more ways to play, players can perform actions as they want. This can provide a better sense of agency, resulting in a more enjoyable experience.

3.1 Flow and Player Agency

The importance of player agency can be understood through flow, described by Csikszentmihalyi in his work covering the optimal experience [11]. It is a state in which a person is focused and immersed to the point that they stop noticing other signals around them, losing track of time. The activity itself provides the person with great enjoyment, and the goal of the activity serves only as an excuse to continue.

Flow itself can be discovered in nearly any activity, provided it meets the conditions for achieving it. These include:

- perceived challenges, or opportunities for action, that stretch, but do not overmatch or underutilise, existing skills
- clear proximal goals and immediate feedback about the progress being made [25]

The conditions are general enough that they can apply to a multitude of activities. For example, for a skilled programmer, it is not uncommon to find flow. A clear indicator exists for what should be done next to achieve a future goal, and as the programmer makes progress, the functionality of the program improves. The implementations might be tricky, requiring substantial, though not excessive, thought from the programmer: if this is the case, the programmer is likely to have an enjoyable experience. Of course it is just as common, if not more so, that the work does not progress as anticipated; hard-to-solve software bugs are likely to emerge, resulting in a state of frustration.

Whilst flow can be found in professional fields, it has an even higher potential to be found in games. This is by nature of the conditions for reaching it, and how they can be served in games. For example, we can consider the original “Super Mario Bros.” [26]. In the game, players are provided with a simple goal: keep progressing forward, from left to right. At the end of each level, players are rewarded with a musical fanfare, giving ample feedback that they are making progress. The challenge in the game increases constantly, requiring more precise jumps and timing to progress. These provide the conditions for flow, and the opportunity for great enjoyment.

Flow in games has been the focus of much research. Studies on gamers and their likelihood to play a game have shown a positive correlation: people will form greater customer loyalty towards a game if it provides them with a flow experience [8]. Additionally, GameFlow [36], which is further detailed in section 4.1, has been proposed as an evaluation tool to review enjoyment in games, being deeply rooted in flow theory. It transforms the various components of flow into a set of heuristics that can be measured in games, categorising them into fields such as concentration, challenge, and clear goals.

One of the categories outlined is control. Not only should players feel they are in control of their characters in the world, and the interfaces via which they are playing, they should also have a sense of control of the game world, and should feel free to play the game the way they want: they should perceive they have a sense of agency. Indeed, Csikszentmihalyi describes the optimal experience as a situation in which a person feels they control their fate, as opposed to external forces; they feel exhilarated and a deep sense of enjoyment as a result, and the moment defines what they want to feel in life [11].

Providing players with perceived agency is far from straightforward. They need to be granted enough control to influence the game world in a meaningful way, which amounts to giving players choices. In fact, game designers such as Sid Meier describe games as “a series of interesting choices” [3], providing context for how important it is to give players a sense of agency. Some games circumvent this by giving players the illusion of choice: by giving options to select from, and acknowledging the choices with feedback, a player may feel they have agency; this is despite the choice having no impact on the story or progression of the game [14]. The risk is that players notice they are not, in fact, in control, and subsequently lose their sense of agency.

The other option to illusions of choice is to provide players with actual choices. The issue with this, however, is that each choice represents a branch in the gameplay, expanding the possibility space of the game—if each branch needs to be scripted, this can require significantly more work from the developers. An alternative solution, which maintains a realistic workload, is to exploit emergence.

3.2 Emergence in Games

As described in the introduction to this chapter, emergence arises when simple rules lead to complex behaviours. This type of emergence can be realised in games, and can be further extended to give players more ways to play. By creating simple mechanics that interact with each other, new and unexpected ways of playing can be discovered. Instead of giving players complex decisions to make, they are given a series of simple ones, which together form a unique result.

When talking about emergent gameplay, we can formalise it using the mechanics, dynamics, and aesthetics game design framework [18]. Mechanics define the rules of the game, and the basic actions a player can take. Dynamics represent the interactions that emerge as a consequence of the mechanics, and aesthetics describe the type of experience dynamics produce for a player. For example, mechanics in "Super Mario Bros." include pressing a button to jump, and enemies being defeated by falling on top of them. This creates a dynamic of defeating enemies by jumping on them, which requires proper timing—producing an aesthetic of challenge.

Just as emergence is defined as simple rules leading to complex behaviours, we define emergent gameplay as simple mechanics producing complex dynamics. These complex dynamics then give rise to aesthetics, setting the tone for players. As dynamics in an emergent system are often unpredictable, it can be hard to design mechanics for a targeted aesthetic.

Before delving into designing for emergence, we take a look at three examples of games that exhibit emergent gameplay. Each of the three display an emergent narrative, which is produced by the players and the dynamics of the system, instead of explicitly designed by the developers. It is a product of the emergent gameplay: as players explore the games, and produce surprising results, they gain unique stories to share with others.

3.2.1 The Sims

A game series that encapsulates emergent gameplay to a strong degree is "The Sims" [22]. Players direct the lives of virtual humans known as Sims. The Sims have some level of their own intelligence, and can be expected to perform tasks related to their needs on their own, such as using the bathroom, or going to sleep when tired. The player can, however, influence their decisions, and can direct them to, for example, make food, watch TV, or socialise with other Sims. These actions all contribute to the well-being of the Sims, and direct their lives.

There is no prewritten narrative in "The Sims". Instead, as players play the game, the story of the lives of their Sims emerges from their choices, and the rules by which the Sims work. Figure 3.1, for example, shows a snapshot of the lives of one family of Sims. From the image, we can understand some of the story behind the family: they have two children, two of the family members have passed away, and the size and furnishings of the home would indicate a moderate income. The large possibility space of the game makes

Figure 3.1: An image from “The Sims 4”, showcasing the house of a family of Sims. The house itself evolved over time, based on the needs of its inhabitants, and choices made by the player.

it unlikely that any other player would have an identical home.

Often, the narratives are controlled to a large degree by players, but sometimes they emerge more from the dynamics of the game. Consider the following experience:

> A player is directing the life of a Sim that has produced a fair amount of wealth. The player has helped the Sim build a manor, and to celebrate, has tasked the Sim with holding a party. The party gets started, and the guests arrive.
>
> As the party goes on, one of the neighbours decides they need to visit the bathroom. Since the manor is large, the player decided that there should be a separate men’s and women’s bathroom. What the player did not know, was that they installed the doors the wrong way around.
>
> As a result, the female guest who needs to visit the bathroom goes into the men’s room by accident. She does what she came in to do, but upon trying to exit, notices that the door shows a sign for a men’s room. She erroneously thinks the door is the entrance to a men’s room, and will not pass through. As a result, she is now in the bathroom with no way out. Trapped, with no food, her basic needs begin declining. Eventually she dies, and

> what started as a house-warming party ends with the peculiar death of a neighbour.

The entirety of the story formed from simple rules by which the Sims function, and through simple decisions that the player made. Yet, evidently, the story that unfolded was quite complex, and meaningful. Throughout all of it, the player was also in control: they could have demolished the door to save the neighbour, but could also choose to let the simulation unfold. The formation of emergent narratives in this manner is a facet of emergent gameplay.

3.2.2 Dungeons and Dragons

Another game to pioneer emergent narratives is "Dungeons & Dragons" [15]: unlike "The Sims", "Dungeons & Dragons" is a tabletop role-playing game, unconstrained by the restrictions of a digital format. Instead of having the structure of a video game to present the game to players, a Dungeon Master coordinates the game and describes the world and story. The game progresses with the Dungeon Master setting the scene, and players interact with the scene by describing their actions to the others taking part. The rules of the game encourage players to be creative with their surroundings: they can, for example, grab a fork and use it as a projectile weapon, knock over a candle to set a pool of oil on fire, or anything else the players can imagine, that the Dungeon Master deems feasible.

For tasks where it is unsure if a player might succeed, such as when shooting an arrow at a far-away target, players roll a set of dice. The outcome of the roll dictates whether or not an action succeeds, and possibly to what extent it is successful. Even if a campaign were to progress exactly as another campaign, with all the same decision, this element of randomness is likely to cause unexpected situations.

The simple mechanics of describing actions, and rolling dice, lead to complex dynamics and outcomes—gameplay emerges as players use their imagination to perform creative actions, and a narrative emerges from the sequence. Additionally, as the players have a large influence on the world around them, they are granted considerable agency. The story does not progress according to a strict design by the Dungeon Master, rather it emerges organically from the choices the players make, and the interaction of the simple rules.

To outline how player agency in "Dungeons & Dragons" leads to emergent narratives, consider the following story from a campaign set in a steampunk world:

> On the third day of a new campaign, players were suddenly surprised by a huge army with airships attacking the city they were currently in. Their best option was to escape the city: conveniently, at the outskirts, there was an ancient abandoned airship, perfect for getting away.
>
> The party headed there, and on the way defeated some members of the invading army. They stole their clothes and donned them as disguises to sneak past any further enemies. Finally, they reached the airship and took off.
>
> The queen of the country, however, was stuck in the middle of the city, and the party felt they had a duty to save her. They turned around, and with incredibly lucky dice rolls, were able to make it there to save the queen. Unfortunately, the airship was heavily damaged, and the party was surrounded by the entire invading army.
>
> The players came up with a crafty solution: they flew up to an enemy ship, and in their disguises, claimed to have taken the queen hostage. They agreed to move all members to the new ship, and bring important cargo over from the failing ship. As the enemy crew boarded the damaged ship to grab cargo, the party took control of the enemy ship and navigated their way out of the battlefield. They now had a larger, faster ship at their disposal, perfect for tackling the challenges of the new campaign.

In this story, the Dungeon Master had planned out an escape route, and had spent time designing the ancient airship the party would use for their travels. By the end of their escape, however, the party had already destroyed the airship, and stolen an enemy ship instead—the agency granted to players by the rules of the game enabled them to deviate from a planned course, and combined with some influential dice rolls, a story emerged.

3.2.3 The Legend of Zelda: Breath of the Wild

Whilst “The Sims” can be paused, and the combat in “Dungeons & Dragons” is turn-based, “The Legend of Zelda: Breath of the Wild” [28] is a real-time action-adventure game. Many of the mechanics of the game revolve around enemy behaviours, physics, and what the developers have dubbed as “the chemistry engine” [17]. This refers to elements, such as fire, electricity, and wind, which affect the state of objects in the game, based on their material.

These, together, form a basis for complex dynamics and the potential for emergent gameplay.

Players can be the drivers behind emergence, but it can also be driven by agents in the environment. For example, an enemy known as the “Guardian Stalker” shoots laser beams at players: if another enemy ends up in the line of fire, they will get hit instead, causing damage. This friendly fire is an emergent dynamic: enemies do not try to attack each other, but a combination of rules produce it:

- Any game object with a damageable property will take damage if hit by a laser
- A guardian will charge and fire their laser at players
- Enemies typically have behaviours that move them towards players

Because enemies tend to move towards players, it is entirely possible that, since they do not have a rule to avoid lasers, they will accidentally end up in the line of fire. Alternatively, players can aim to produce a similar situation, by intentionally dodging a laser such that it goes past them to hit another enemy. This kind of friendly fire is an emergent dynamic, yet it also provides agency to players, as they can intentionally produce the dynamic.

The friendly fire is relatively situational, though, and can be hard to pull off. Importantly, to make it easier to instigate dynamics, players have mechanics to substantially influence the game system. In the introductory part of the game, players are given a series of “runes”, which grant them mechanics that tie in with the physics of the game. The runes are as follows:

- Magnesis: allows the player to manipulate metallic objects, as if by telekinesis.
- Round Remote Bomb: grants the player a spherical bomb that they can throw, and detonate at will. Upon detonation, it causes an explosive force that damages and pushes objects and creatures.
- Cube Remote Bomb: functions in the same way as the round remote bomb, except it is cubic, and thus less prone to roll.
- Stasis: allows the player to stop the flow of time for an object. When the object is stopped, it will store any energy it is hit with, which will be turned to kinetic energy once time begins flowing again.

- Cryonis: permits players to create pillars of ice from water. These are solid blocks that can be climbed and mounted, and can block projectiles.

Almost every moving object in the game has physical attributes that interact with the player runes. For example, the game teaches players to use stasis to freeze objects in time, then hit the objects with a weapon to store energy. Once an object unfreezes, the energy is turned into kinetic energy, resulting in it moving. Enemies that try to hit the player might inadvertently hit a frozen object, which will also add energy to it. This can turn against the player, or alternatively, the player can use it to their advantage.

Figure 3.2: Players in “Breath of the Wild” [28] can create flying machines to travel through the skies. This was not designed by the developers, but emerged from the mechanics of the game.

With the simple mechanics of the runes, players discovered some creative ways to use them, such as discovering a dynamic of flight using the Stasis rune. Players can, for example, use stasis on a falling tree, hit it to store energy aimed upwards, then jump and grab onto the tree before it unfreezes. Once it unfreezes, the tree will shoot upwards rapidly, akin to a rocket, together with the player. This can be an efficient way of reaching high-up places.

Many of these dynamics were discovered by the developers, but intentionally left in, as having them provides players with better agency. However, one dynamic the developers had not anticipated, was that players could build flying machines using metal objects and the magnesis rune [7], as shown in figure 3.2. Generally, the rules of the game do not allow the player to use magnesis on an object they are standing on, but by placing an object between the metal object and the player, as an insulator, this rule can be circumvented. This allowed players to fly using magnesis, granting them a new way of playing.

An important aesthetic for the game is exploration, and on this front, the emergence permitted by the mechanics of the game supports it well. Not only can the player explore a large open world, they also have freedom in experimenting with how they approach problems. For example, to reach the top of the mountain, one player might choose the most straightforward option of climbing the rock wall, whereas another player will choose to use trees as rockets to ascend, and a third will build a flying machine and use magnesis to reach the top. Each player had the same objective, but their stories of how they reached it were different.

3.3 Emergence in Other Fields

Emergence, to some extent, is present in most games these days. From visual effects to emergent gameplay, emergence is used to create a more enjoyable experience. The work on emergence, however, has its origins in the study of nature. As people have striven to understand the phenomena around us, they have developed mathematical structures to produce them. These all have the same premise: simple rules lead to complex behaviours.

A certain number of these works have had a significant effect on video games. We cover two of them: Conway's "The Game of Life", and Reynolds' "Boids". The simplicity by which they function, combined with the complexity they produce, epitomise the concept of emergence.

3.3.1 Conway's Game of Life

Conway's "The Game of Life" is a cellular automaton with remarkable properties and significance to games. Fundamentally, a cellular automaton is a lattice of cells, each with $k >= 2$ possible states. The cells update in discrete timesteps, according to a local rule composed of a set of rules that accounts for a cell's current state and its neighbouring cells' states. [32] In the simplest case, a cellular automaton is a grid of square cells, where each cell is either alive or dead, i.e. on or off. Although cellular automata are a mathematical structure, they can be used, for example, to emulate the formation of unique snowflake patterns [40].

In the case of “The Game of Life”, the lattice is a square grid, and all cells have two possible states: either alive or dead. Each cell updates according to simple rules based on neighbouring cells. These rules can be summarised as follows:

- If there are less than two nearby cells, the cell dies of loneliness
- If there are more than three cells, the cell dies from overpopulation
- An empty cell with three neighbours becomes living, through reproduction

Remarkably, despite the incredibly simple rules, very complex behaviours and patterns emerge as the simulation is stepped forward. The behaviour can seem nearly organic, as if the system was alive. With certain conditions, very specific, interesting patterns will occur. These can include stable structures that do not move, patterns which oscillate between states, and even “spaceships”, which move across the grid. Many of these patterns have names such as “beehive”, “toad”, and “acorn”. Despite being mathematical constructs, we emblematise them using names; to the player, they carry significance.

Interestingly, the automaton is called a game, but perhaps it is fitting. Players have the freedom of coming up with starting configurations that produce interesting simulations. The simulation itself can appear exciting, as shapes form and disappear. Despite no input beyond a starting configuration, the possibilities provided are endless—something which would be impossible to achieve with only scripting events in games.

3.3.2 Boids

Boids, short for “Bird-oid objects”, is a computer program developed by Craig Reynolds in 1986, that simulates the flocking of birds [30]. The flocking is realised through emergent behaviour, as the boids themselves all act through simple rules. These are as follows:

- Cohesion: a boid attempts to stay close to the centre of its flock, formed by its neighbours
- Separation: a boid will try to stay away from its neighbours, to avoid collision
- Alignment: a boid will try to match the velocity of its neighbours [16]

With only three simple rules, flocking emerges—the appearance is akin to that seen in nature with fish and birds. Not only are the rules simple, but they are easy to extend. For example, a rule to stay away from walls could be added, to make the flock navigate around buildings. This makes the flocking not only suitable for creating visually interesting features, such as flocks, but also makes it suitable for the behaviours of agents such as enemies in games. For example, one can consider an enemy to have a simple rule that it will always try to fly towards the player, and stop when they are close enough. With several these enemies, they are likely to clump up into a single spot, as they all try to get to the player without considering the other enemies in the way. If a simple separation rule is added, as with Boids, they will avoid clumping, and will instead circle around the player, in a more threatening and organic manner. Behaviour patterns emerge that might otherwise be technically challenging to implement.

3.4 Physics in Games

Mimicking what we are used to in the real world, many games emulate physics to some degree; sometimes aiming for accuracy, such as emulating forces acting on spaceships in "Kerbal Space Program" [20], and at other times interpreting it more liberally, such as the falling blocks in "Tetris" loosely adhering to a sense of gravity. In action games, physics often play a vital role: a player's expectation when throwing a grenade, falling off a cliff, or wielding a flamethrower, are all based on our expectations of how the physical world around us works. The consensus among players, particularly in action oriented games such as first-person shooters, is that physics should behave consistently, in an expected way[34]. A consistent physics implementation helps drive players' intuitions, and improves their immersion in games.

However, as exemplified by many critically acclaimed games, the need for consistency does not imply the need for realism; a game designer can have some creative freedom and only loosely interpret physical phenomenon. Bending the rules of physics, while still adhering to player expectations, can lead to novel experiences that provide great entertainment. By basing mechanics on the physical nature of the world around us, players can effectively gain an understanding of how the rules of the game work. In this section, we look at a few critically and commercially successful games, that base their gameplay on some loosely interpreted physical phenomena. We consider how they twist the rules of physics to produce better gameplay, while still maintaining players' intuitions for the mechanics.

3.4.1 Portal

“Portal” [39], released in 2007, is a puzzle game with a novel mechanic of connected portals. Players are able to spawn blue and orange portals, which create a spatial link: any object that travels in through one portal will immediately travel out the other one. Furthermore, as the object travels through the portals, it retains any kinetic energy it had, but the velocity of the object is reoriented towards the normal of the portal it is exiting through. This forms the basis for gameplay and puzzles, tasking players with transporting objects, and themselves, through levels and across obstacles.

Although the portals act similarly to the concept of wormholes, on closer consideration they act in an impossible manner: they seemingly violate the first law of thermodynamics. An object on the floor can be teleported to fall from the roof, seemingly without doing any work; the object gains potential energy out of nowhere, adding energy to the system. Yet the physics otherwise act according to players’ expectations, such as objects accelerating downwards via gravity. The portals, too, work consistently: importantly, the rules for portals are simple enough that learning the mechanics is easy for players, despite straying from realism.

3.4.2 Super Mario Galaxy

Upon release, “Super Mario Galaxy” [27] was received with universal acclaim [23]. Similarly to other modern “Mario” games, it involves platforming and exploration in a 3D world. What sets the game apart is its gravity: the game takes place in outer space, on a variety of planetoids. The planetoids are generally quite small, yet Mario is able to run around on them as if they had the gravity of a much larger body of mass, such as the Earth. Additionally, the magnitude of the gravity of a planetoid is constant when within range. If there are multiple planetoids acting at once, only the closer one’s gravity is considered.

Despite not accurately adhering to the laws of physics, “Super Mario Galaxy” still captures the essence of gravity, as objects are pulled towards large masses. Furthermore, we generally do not experience large fluctuations in gravity during our lives. It then does not seem out of place for a planetoid’s gravity to remain constant.

The constant gravity is important for “Super Mario Galaxy”. As it is a platforming game, players are expected to make well-timed jumps, over obstacles, and on top of enemies. If there was any considerable variation in the gravity of planets, the timing for jumps would also consequently vary more. By keeping it constant, players are better able to learn the controls of the game, and attain the skills needed for performing accurate jumps.

Chapter 4

Method

Like with most games, our foremost aim with "Plusminus" was to develop a game that provides enjoyment for players. Our hypothesis was that, by promoting emergence, we could provide players with a large possibility space. Furthermore, by designing the mechanics to provide players with a sense of control, its combination with the large possibility space would provide ample agency—which is crucial in creating an immersive experience that leads to a state of flow. There are, however, few guidelines for tackling the more specific question: how can mechanics be designed for emergence and control in a physics-based game?

Since physics in a game need to match players' expectations from the real world, this sets a unique constraint on the task. A mechanic cannot stray too far from the real, and careful consideration needs to be taken when determining what is still within the bounds of expectation. Past titles that use physics-based mechanics in novel ways can be used as inspiration, as covered in section 3.4. The games fulfil player expectations without being bound to realism.

Thus, we have three criteria that the mechanics of "Plusminus" need to fulfil. They should:

- promote emergence
- provide a sense of control
- match player expectations

This becomes a balancing act: limitations to mechanics can improve control, but will likely decrease opportunities for emergence. Deviating from reality might improve emergence, at the risk of violating player expectations. And finally, modelling physics too accurately risks diminishing players' senses of control.

As with many balancing tasks, there are no predefined rules for how to tune the mechanics. We do, however, aim to provide some guidance and design suggestions, motivated by our experiences in designing the mechanics of “Plusminus”. Through describing the various problems we solved, the missteps we took, and the trade-offs we made, we hope to provide valuable insight into the process of balancing physics-based mechanics.

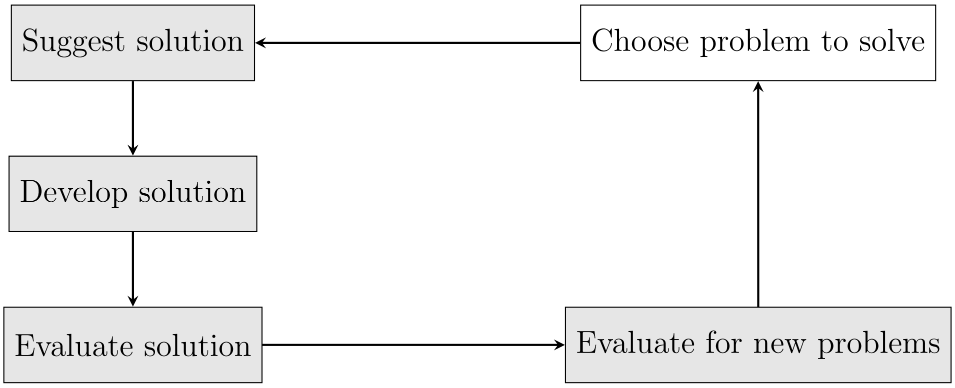


Figure 4.1: The simplified design cycle used for the mechanics of “Plusminus”, based on Takeda et al.’s work [37]. The cycle starts with choosing a problem to solve, and continues as long as there are problems to be solved.

The design process of the mechanics was iterative, following a design cycle as outlined by figure 4.1. Each development iteration is followed by the evaluation of mechanics against the previously outlined criteria. Although the evaluation in figure 4.1 is split in two, it was typically most practical to perform both simultaneously. For “Plusminus”, the evaluation was performed using two methods: heuristics and playtesting.

Heuristics are a common method of evaluation in the field of human-computer interaction and video games. With video games, heuristics generally provide guidelines for improving enjoyment for players. In fact, studies have indicated that heuristics, particularly in early development phases, are better than playtests at identifying problems in games [12]. As the target for “Plusminus” was a prototype that would demonstrate magnetic gameplay, heuristics proved to be an appropriate means of evaluation.

Although there are several comprehensive lists of heuristics available for video games, we chose the GameFlow model [36] based on the well-motivated theory of flow used as a motivator for the heuristics. Notably, the GameFlow heuristics have a category that describes how players can gain a sense of control—which is vital for attaining player agency, as described in section 3.1. The feeling of control ties in strongly to the mechanics of a game, which is the focus of this thesis.

Despite using heuristics as the basis for evaluation of enjoyment, we also performed playtests approximately every two weeks. These were informal, but helped identify which heuristics were being violated: notably, all of the heuristics for control described in GameFlow, start as “players should feel”. Consequently, watching players play, and asking some follow-up questions, helped highlight various issues with the mechanics of the game.

4.1 GameFlow

With flow theory as the backbone for motivating our work, it then naturally follows that we would evaluate the game based on the GameFlow model [36]. GameFlow brings together various heuristics from literature, and unifies them all in a way structured by flow. They are intended as a method for evaluating enjoyment in games, and consequently we use them for the evaluation of “Plusminus”. As this thesis is concerned with mechanics, as opposed to elements such as level design, we curate the GameFlow heuristics into a smaller subset, as seen in figure 4.2.

Our goal of player agency is clearly represented through the categorical element of control, but aspects such as the ease of mechanics, as well as the feedback from using said actions, are likewise considered. Of particular note is the last item under Control: “players should feel ... that they are free to play the game the way they want (not simply discovering actions and strategies planned by the game developer)”. In effect, this is achieved through encouraging emergent gameplay, which is a focus in “Plusminus”. As we discuss the design of the mechanics in the next chapter, we evaluate the mechanics after each iteration by referring back to specific heuristics.

4.2 Playtesting

We held playtests at regular intervals throughout the development of “Plusminus”. These were useful in giving us insight into all aspects of the game, including art, sound, and level design. Sometimes, the feedback was also directly relatable to the mechanics of the game. For example, if someone complained that the character was “hard to control”, or that they “didn’t understand how the magnetism worked”, then there was evidently a problem with the mechanics.

The information we collected from the playtests was based on empirical observations, as well as open-ended questions. Typically, we would follow up any playtest with two questions:

Player Skills Games must support player skill development and mastery	• players should be able to start playing the game without reading the manual • game interfaces and mechanics should be easy to learn and use
Control Players should feel a sense of control over their actions in the game	• players should feel a sense of control over their characters or units and their movements and interactions in the game world • players should feel a sense of control over the game interface and input devices • players should not be able to make errors that are detrimental to the game and should be supported in recovering from errors • players should feel a sense of control and impact onto the game world (like their actions matter and they are shaping the game world) • players should feel a sense of control over the actions that they take and the strategies that they use and that they are free to play the game the way that they want (not simply discovering actions and strategies planned by the game developers)
Feedback Players must receive appropriate feedback at appropriate times	• players should receive immediate feedback on their actions

Figure 4.2: A portion of heuristics, deemed relevant to mechanics, from the GameFlow model [36].

- What did you enjoy about the game?
- What was frustrating?

The questions were open-ended, as these typically provide more information than close-ended questions [29]. We also asked follow-up questions if we observed anything particularly surprising as testers were playing.

The playtests were held in events where more formal procedures would not have been appropriate, such as on the show floor of the Nordic Game Conference 2018. This meant that we did not conduct surveys, or otherwise spend much time asking players questions. Although they would have proven useful, the issues with “Plusminus” were typically noticeable enough that we were able to identify them without a more thorough evaluation. The details of what specifically needed changing about the mechanics of the game were in turn evaluated using heuristics.

Chapter 5

Implementation and Design

The bulk of the mechanics developed for “Plusminus” revolve around magnetism, a phenomenon we have all encountered in some form in our lives. Sometimes, magnets are used for utility, as we stick items to our fridge door. At other times, they are used for play, as we construct towers out of piles of magnets and metal objects. In essence, we have expectations of how magnetism works, and what it can be used for.

A big part of the challenge with “Plusminus” was in meeting all these expectations. The magnetism in the game needed to act realistically and consistently enough that players would find it intuitive and immersive. At the same time, as per our goals, it needed to be modelled in a way that permitted meaningful gameplay, giving players ample control while also promoting emergence. To this end, we designed and iterated the mechanics of “Plusminus”, starting with a basis on realism, then adapting the rules to better fit our goals, all the while maintaining consistency and matching player expectations. We also describe mechanics built to support the gameplay around magnetism, to further build a system that promotes emergence.

While it might be enough to provide general design guidelines and observations based on the development of “Plusminus”, we also provide a detailed account of the physical equations used and developed. The forces we calculated are based on electromagnetism; however, there is a duality with gravity [13]. To this end, we believe our equations for electromagnetism in “Plusminus” could be adapted to compute other kinds of forces in games in a consistent and intuitive manner, while providing exceptional room for control and emergence. Our insights into the problems and solutions with the interaction with magnetism should also be applicable to games striving for similar interactions. Thus, we give a detailed account of the steps we took in adapting an accurate model of physics, to one that better fits the needs and constraints of a video game; additionally, we provide details of the usability problems we encountered, and how we solved them.

5.1 Magnetism

The central mechanic that sets "Plusminus" apart from other games is magnetism. The mechanic is based on the physical force of electromagnetism: opposite poles attract, and like poles repel. One model for understanding forces between magnetic poles is the Gilbert model: the poles of a magnet are considered to be composed of a distribution of magnetic charges, akin to electrical charges. Computing the force produced between any two magnetic charges can be performed using Coulomb's law, with electrical charges replaced by magnetic charges.

$$F = k_e \frac{q_{m1} q_{m2}}{r^2} \tag{Coulomb's law}$$

The equation expresses how magnetic charges, represented by q_{m1} and q_{m2}, interact. If the signs of the charges are the same, the resulting force is positive. If they are opposite signs, the force is negative. This effectively describes that the charges either repel each other when they have the same sign, or attract each other, if they have opposite signs.

While the typical magnet has two poles, with "Plusminus" we elected to consider polarised objects to be monopole magnets. This was chosen to improve understandability and control: with dipole magnets, forces and torques produced between two magnets are highly sensitive to orientation. Thus, as objects interact and rotate, the result can be be hard to predict. This is ameliorated by considering magnets to be monopole, as the produced forces are either those of repulsion or attraction, as opposed to a combination of the two.

By considering only monopole magnets, the equations for magnetic forces also become simpler. By Newton's shell theorem, spherical shells and solid spheres of uniform charge density act as point charges. Coulomb's law can be applied to compute the force between the spheres, with r equating to the distance between the centre of the magnetically charged spheres. Thus, with Coulomb's law as a basis, we can surmise the following properties of magnets, which work as the foundation of the rules of magnetism in "Plusminus":

- Opposite polarities attract each other
- Like polarities repel each other

- The closer the objects are to one another, the stronger the force produced between them

Although the rules are simple, balancing them turned out to be challenging. While Coulomb's law provided a basis to start with, our initial implementations worked counter to our expectations.

5.1.1 Calculating Magnetic Force

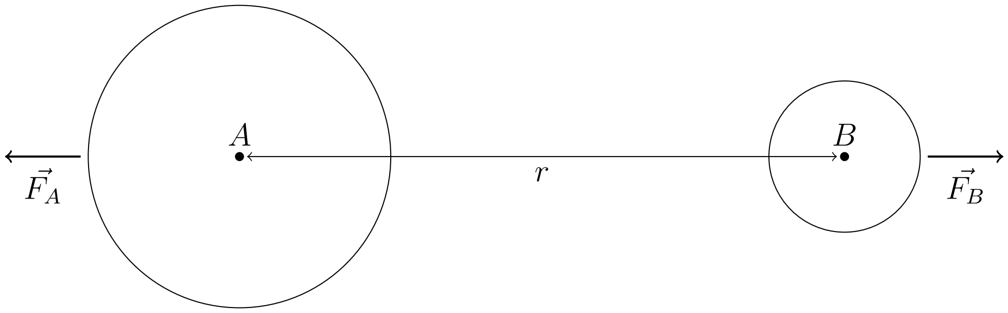


Figure 5.1: Two magnetic spheres interacting, with charges q_A and q_B. The magnitude of the forces $\vec{F_A}$ and $\vec{F_B}$ are equal, given by Coulomb's law, i.e. $F = k_e \frac{q_A q_B}{r^2}$. The direction of the force $\vec{F_A}$ is equal to $\hat{F_A} = \frac{\mathbf{A}-\mathbf{B}}{r}$, and $\vec{F_A} = F\hat{F_A} = k_e \frac{q_A q_B}{r^2} \frac{\mathbf{A}-\mathbf{B}}{r}$. Finally, $\vec{F_B} = -\vec{F_A}$. The radius of the spheres does not affect the forces between them [13].

Most of the objects in "Plusminus" are spherical, hollow shells. The forces between such objects, assuming a uniform charge distribution, follow Coulomb's law [13]. As the distance r between spheres decreases, the magnitude of the force, F, increases exponentially. Figure 5.1 illustrates the forces between two uniformly charged spherical objects.

We tested a configuration in "Plusminus" where all objects were considered to act as spheres, thus forces between them were calculated according to Coulomb's law.

If we consider a case where $k_e q_1 q_2 = 10$, then we get a force curve, between objects at distance r from each other, as shown in figure 5.2. The chart is limited to the domain [1 : 5], but if we were to extend it to 0, we would notice that the force F grows to infinity, as:

$$\lim_{r \to 0} F = \infty$$

The first issue is that, in particular, if distances are very short, i.e. $r < 1$, we get incredibly strong forces. The result in-game is that the physics simulation becomes unstable. It might be obvious that an unstable physics simulation will have a negative impact on gameplay. Specifically, the heuristic "players should feel a sense of control over their ... interactions in the game world" is violated when the simulation breaks, as the instability directly

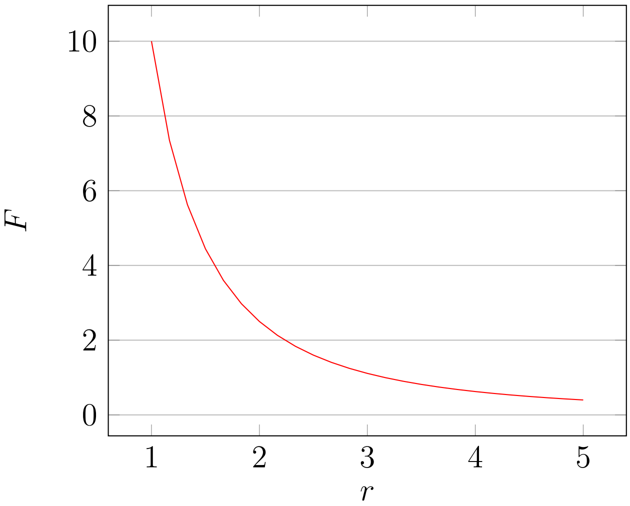


Figure 5.2: The force F between two objects, as per Coulomb's law, with respect to the distance r between them. As the distance increases, the force between the objects quickly drops. At distance $r = 1$, the force is equal to $F = k_e q_1 q_2 = 10$.

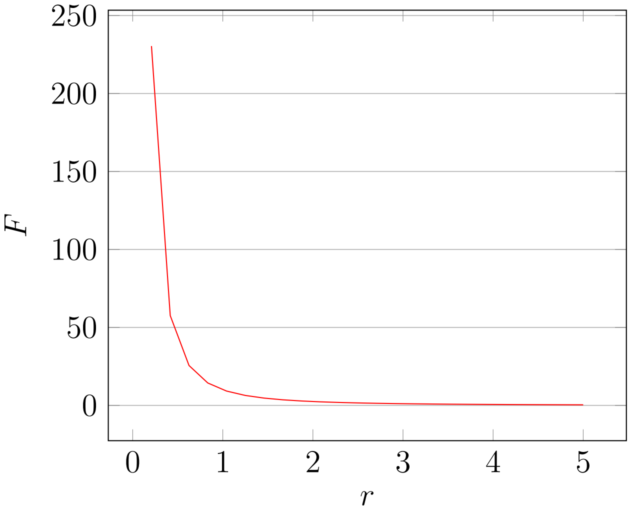


Figure 5.3: The same force curve as in figure 5.2, but in the domain $[0, 5]$. As the distance r approaches 0, the force approaches infinity.

removes the sense of control.

This is further exacerbated when we want to increase the effective range of two interacting objects. Suppose we define a force to be effective if $F \geq 1$. In figure 5.2, this holds true approximately when $r < 3$. Consider a case in which we want to double the effective range to $r < 6$: we would in fact have to quadruple the base force to $k_e q_1 q_2 = 40$. This further exacerbates the issue of strong forces at short distances breaking the simulation.

Although we achieved some form of gameplay with this model, the instability made it comical, and meaningless. In playtests, people would comment that “it had potential”, but it did not provide meaningful gameplay. We had to reconsider our base assumption, that all objects act physically correctly, as charged spheres.

5.1.2 Using Joints to Stabilise the Simulation

The main instabilities of the system arose when two oppositely polarised objects would come into contact with each other, i.e. they would attach. The problem, however, was at such proximities, the forces would often be high enough that the simulation would turn unstable. We tackled this problem directly, by adding a fixed joint constraint between the objects as they attached.

The constraint locked the two objects together, as if they were a single object. Once they did this, we would no longer apply magnetic forces that they should have produced to each other. If the polarity of one of the magnets was changed, the constraint would be removed, and the forces between them would once again be applied.

This fixed the instabilities for the most part. The large forces at close proximities no longer existed—at least not for forces of attraction. Forces of repulsion would still produce significant accelerations, that at times resulted in instabilities. This, however, happened far less often.

Despite seeming like a decent workaround, the solution soon presented new problems. Although the magnetism was more controlled, the constraint, in many situations, caused awkward and unnatural configurations. For example, a sphere attracted to a flat surface would get locked there, and not move no matter what other forces were exerted on it. The player expectation that the sphere would roll across the surface, if pushed by some force, was violated.

In hindsight, it is of no surprise that this occurred. Magnets attaching is an emergent behaviour in nature: it is a dynamic that arises from the rules of opposites attracting, and forces growing as magnets get closer to each other. By constraining it manually, an emergent dynamic is hardcoded

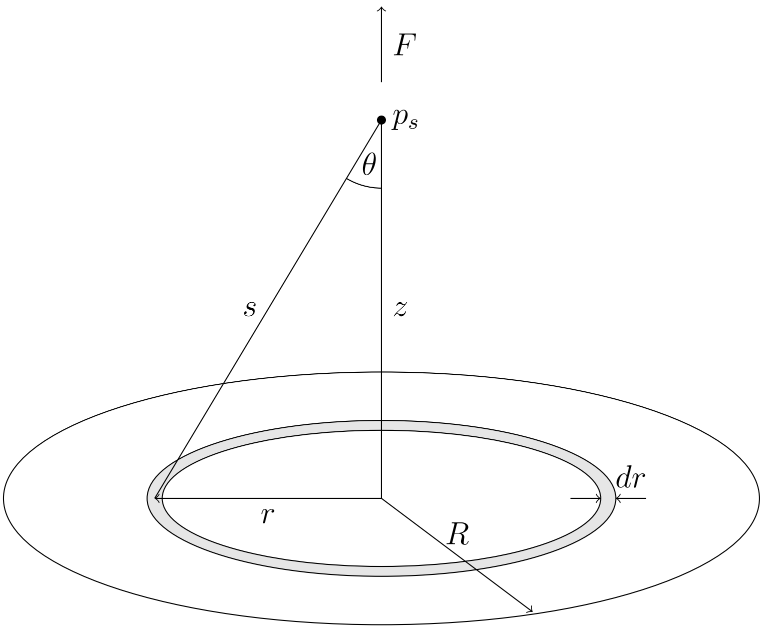


Figure 5.4: A disc with total charge q_p acting on a sphere centred at point p_s, with charge q_s. The total force acting on the sphere at p_s can be calculated by integrating the force produced by rings of radius r, with infinitessimal thickness dr, in the range $0 \leq r \leq R$.

as a mechanic instead. This additional rule has the consequence that other emergent properties, such as magnets aligning in stable configurations, no longer occur.

We decided stability needed to be implemented in some other way, without adding unnecessary complexity to our base rules. This led us to reconsidering our generalisation of everything acting like spheres.

5.1.3 Moving Beyond Spheres

Consider a different case: instead of two spheres, we have a sphere centred at point p_s, and a magnetised disc of radius R, as demonstrated in figure 5.4. The sphere acts as a point charge, but the same generalisation cannot be made for the disc. We consider the normal of the disc to be parallel to the z-axis, and the centre of the disc to be at the origin. The sphere is located along the z-axis at some distance z from the origin. If we consider the force produced by a ring on the plane, with radius $r \in [0, R]$, then the force produced by the ring is equal to:

$$dF_z = k_e \frac{dq_p q_s}{s^2} \cos\theta$$

Where:

$$\cos\theta = \frac{z}{s},\ dq_p = q_p \frac{2\pi r dr}{\pi R^2} = q_p \frac{2rdr}{R^2}$$

$$\begin{aligned} dF_z &= k_e \frac{q_p \frac{2rdr}{R^2} q_s}{s^2} \frac{z}{s} \\ dF_z &= k_e \frac{q_p q_s 2rzdr}{R^2 s^3} \end{aligned}$$

The total force, then, is equal to:

$$\begin{aligned} F_z &= k_e \frac{2q_p q_s z}{R^2} \int_0^R \frac{r}{s^3} dr \\ &= k_e \frac{2q_p q_s z}{R^2} \int_0^R \frac{r}{(z^2+r^2)^{3/2}} dr \\ &= k_e \frac{-4q_p q_s z}{R^2} \left(\frac{1}{\sqrt{z^2+R^2}} - \frac{1}{z} \right) \\ &= k_e \frac{4q_p q_s}{R^2} \left(1 - \frac{z}{\sqrt{z^2+R^2}} \right) \end{aligned}$$

The higher we increase R, the closer the force response is to being linear, as shown in figure 5.5. This includes the domain [0, 1], which proved troublesome for our earlier implementation, where all objects were considered to act as spherical objects. This kind of interaction fixes the issues we were having earlier: namely, the range can be increased while keeping the force in check at short distances, and furthermore has a nice property when r approaches 0:

$$\lim_{r\to 0} F_z = k_e \frac{4q_p q_s}{R^2} \tag{5.1}$$

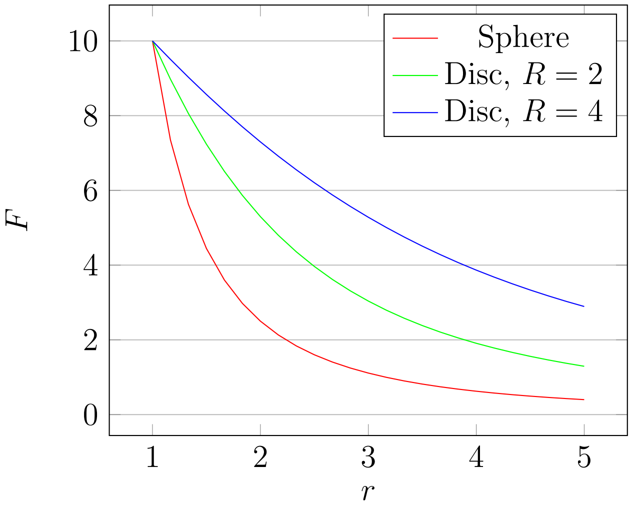


Figure 5.5: The force F between two objects, with respect to the distance between them r. The red line represents the force produced between a sphere and a point charge. The green and blue lines show curves between discs of radii $R = 2$ and $R = 4$, respectively. The more R grows, the more linear the curve becomes.

The maximum force is clamped to an easily computed value, making balancing of gameplay easier.

This improved force curve is based on an interaction with a disc in an ideal situation where the particle being computed is in an optimal orientation relative to the disc. The electromagnetic force varies greatly with the shapes of objects, and their relative positioning and orientation: it is not feasible nor necessarily useful to accurately compute the forces between objects, as long as the result meets player expectations. Thus, it suffices that it is well approximated: it should have the characteristics of magnetism, while being as easy as possible to balance for gameplay. And, importantly, it should be balanced in a way that provides control for the player, which brings us to the topic of approximating magnetism.

5.1.4 Approximating Magnetism

Because of the complexity of electromagnetic fields, it is not feasible to calculate accurate results in realtime, if shapes are anything beyond primitives such as spheres. Additionally, as seen with figure 5.2, it can be hard to attain magnets with reasonable range that act in a stable manner. To ameliorate the situation, we implemented an approximation.

The approximation aims to capture the characteristics of magnetism, while being easy to adjust based on the needs of the gameplay. Based on actual magnetism, but also the needs of “Plusminus”, we decided it should have the following properties:

- Opposites should attract
- Like should repel like
- The force between objects should increase as they get closer
- It should be possible to increase the range of a magnet without increasing the maximum force exerted by the magnet

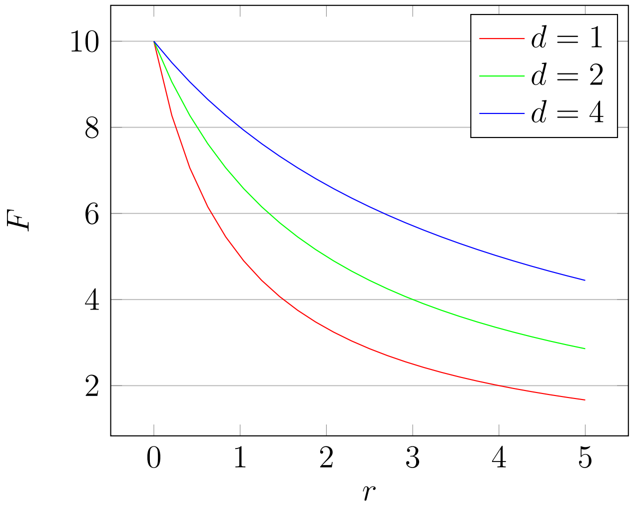


Figure 5.6: The force F between two objects, with respect to the distance between them r, using the approximate equation 5.2. By adjusting the range modifier d, the range of a magnet can be increased without increasing the maximum force it exerts.

A simple equation that has the desired characteristics is:

$$F = k\frac{q_1 q_2}{\frac{r}{d} + 1} \tag{5.2}$$

Here k is a constant, q_1 and q_2 are the charges of the interacting objects, r is the distance between the objects. d is a range modifier: increasing it increases the effective range of magnetism, without affecting the maximum force $F_{max} = kq_1q_2$. The shape of the force curves this produces can be seen in figure 5.6.

Evaluating this change, we saw immediate improvements. The interactions were all far more stable, improving controllability for the player—and while we used an approximation, the interactions still met player expectations. Yet a new problem emerged.

5.1.5 Restricting Magnetic Range

Although the approximate equation we developed for magnetism fixed instability issues, and permitted magnets to have a greater effective range without compromising stability, our playtests showed a new problem: it was not obvious to players what magnets were interacting with each other, with objects seemingly moving at random. This stemmed in part from the unlimited range magnets had: a polarised object in another room could affect an object in the current room, with no visuals to demonstrate that this was happening. It was also unclear when the interaction between two objects would start having a noticeable effect, as once again, there were no visuals to indicate it.

Our solution was to set a finite range for the magnets. We tuned equation 5.2 to have a limited range, as follows:

$$F = \begin{cases} kq_1q_2\left(\frac{2}{\frac{r}{d}+1} - 1\right), & \text{if } r < d \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

Now, d dictates the maximum range to which a magnetised object is effective, producing curves as shown in figure 5.7. Clamping the range to a maximum also allowed us to give a clear indicator for players as to when an object was in the range of a magnetic field. We displayed this range using a bubble as shown in figure 5.8.

Although the bubble seemed clear, another issue became apparent in playtests: players did not understand that the magnetism would only apply when an object entered a magnetic field, as opposed to when two magnetic fields overlapped. We modified the system accordingly, such that magnetised objects would interact when fields overlapped. This immediately made interactions more clear to both playtesters and us as developers. The modified equation for magnetic force then became as follows:

$$F = \begin{cases} kq_1q_2\left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1\right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

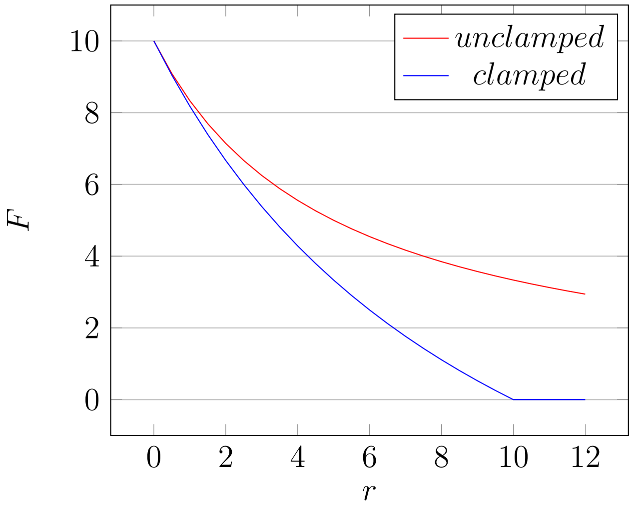


Figure 5.7: Comparing the force F between two objects, with respect to the distance between them r, when the force is clamped to range d. The clamped equation is more linear, with a clear cutoff. The unclamped equation produces forces that reach to infinity.

This iteration of the magnetism produced stable, predictable magnetism for the most part. We were clearly approaching a good balance between controllability and realism. Yet, certain interactions still did not behave according to player expectations, and these stemmed once again from an oversimplification of the shape of magnets.

5.1.6 Generalising Beyond Spheres

When dealing with spheres, it makes sense to use their centre as the origin and target for magnetic forces. Yet consider the case described by figure 5.9. If the magnetism was accurately modeled, points closer to each other would experience stronger forces than points further away. This would, in turn, produce a torque on the objects, rotating them towards an equilibrium.

To emulate this, we shifted away having a single origin for magnetism for objects. Instead, one object is composed of multiple magnetic targets. The location of these targets depends on the shape of the object, as follows:

- Box: the box has a target at the centre of each of its faces
- Sphere: the sphere has a single target at its centre
- Capsule: the capsule has two targets, and acts as if it was composed of two spheres at its ends

Figure 5.8: The player opening metal doors. The left side of the door is positively polarised—the range of its magnetic field is displayed using a bubble around it.

- Mesh: for a generic mesh, a target is located at each of its vertices

When two objects interact, each target acts as a separate source of magnetism. If an object has charge q, and n targets, then each of its targets acts as if it was a magnet with charge $\frac{q}{n}$. This roughly approximates integrating magnetic forces across the surfaces of the objects; the more an object is subdivided into magnetic targets, the closer it will get to an accurate result. On the other hand, the subdivisions also make the simualtion more computationally expensive. Thus, finding the right number of magnetic targets for a shape amounts to finding an appropriate balance between accuracy and performance.

This modification to the way we evaluated magnetism had some favourable effects. Doing internal testing, we noticed that flat surfaces would align more often with each other, and when they did, the attachment would typically be stronger as well. Moreover, it opened up new strategies: objects being repelled could more often be used as platforms, as the produced torques meant they would not rotate in midair as easily. The change, then, seemingly improved emergence, while better matching expectations from real physics.

There was one caveat with the new magnetic targets: they appeared to produce results matching expectations between objects that were approximately the same size, but not between objects with large scale differences. In particular, when a small object would get attracted to a large object, it was

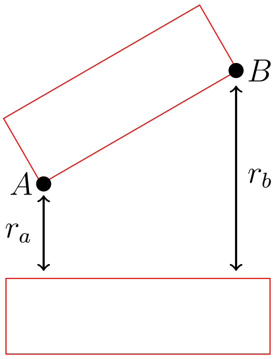


Figure 5.9: Two positively polarised objects are exerting forces on one another. If we consider the forces exerted by the lower object onto the upper object at points A and B, we would expect them to be different. r_a is shorter than r_b, hence the force should be stronger at A than B.

always apparent that it was getting attracted towards the closest magnetic target. For example, with a very large box, a small object would always attract towards the centre of the closest face of the box. The expectation would be that it would instead attract towards the closest point on the surface of the large object.

We made a change accordingly. With large scale differences, instead of considering magnetic targets on the larger object, we would instead consider the closest points on the surface of the large object, from each of the magnetic targets of the smaller object. This made interactions appear more natural, and made smaller objects attract to, and repel, from the closest surface point of the large object.

5.1.7 Balancing Magnetism

The equation 5.8 we had for magnetism proved to be effective: it produced expected results when properly balanced. Objects of opposite polarities would attach to each other, systems of multiple polarised objects would align in a structured lattice, and it was even possible to get magnets floating between two other magnets. However, we ran into situations where the simulation would break, or not work as expected: for example, a large magnet in the ceiling would be balanced in a manner that it would be able to lift a heavy metal box off the ground, when they were oppositely polarised. When a small box was brought in, however, it would accelerate incredibly fast, and would act in an unstable manner.

We can solve for the maximum acceleration produced by the force between two objects using Newton's first law of motion, $F = ma$. Thus, for an object of mass m, we have:

$$a_1 = \begin{cases} k\frac{q_1 q_2}{m_1}\left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1\right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

It is of note that, now in an interaction between two objects, there are 6 variables to tune: q_1, q_2, m_1, m_2, d_1, and d_2. The number of variables can make it difficult to find the right values. If we consider that the charge of an object and its mass are directly proportional, we can reduce the number of variables that need balancing in each interaction.

$$c = \frac{q}{m} \tag{5.6}$$

We set that charge can be directly deduced from mass (equation 5.6), i.e. $q = cm$, where c is a global constant. The new set of variables to balance in a situation are: m_1, m_2, d_1, and d_2. The acceleration of an object can be rewritten as:

$$a_1 = \begin{cases} kc^2 m_2\left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1\right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

We can further define that $k_m = kc^2$, simplifying the acceleration to:

$$a_1 = \begin{cases} k_m m_2\left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1\right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

The maximum acceleration a body can induce on another body is then $a = k_m m$, where m is the mass of the inducing body. It is of note that the mass of the accelerating object has no effect on the acceleration. This is akin to gravity: the acceleration by the Earth on objects falling towards it are irrespective of their masses.

This configuration did not, however, work well with the gameplay we had in mind. The game has small objects with mass 10 kg, but also larger objects with mass 1000 kg. The acceleration produced by the larger object is then 100 times that of the lighter object. The result is that either the accelerations produced by the lighter objects will be insignificant, or those produced by the larger objects will be enormous, and once more cause instabilities.

This created another situation in which we needed to stray from realism to improve control. We concluded that the important interaction in maintaining the illusion of magnetism was that heavier objects should accelerate lighter objects towards them more rapidly than vice-versa. This was achieved by modifying our force calculation as follows:

$$F = \begin{cases} k_m \min(m_1, m_2) \left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1 \right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.9}$$

The acceleration is then:

$$a_1 = \begin{cases} k_m \frac{\min(m_1, m_2)}{m_1} \left(\frac{2}{\frac{r}{d_1+d_2}+1} - 1 \right), & \text{if } r < d_1 + d_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

The result is that if m_1 is the lighter mass, the maximum accelerations, attained when $r = 0$, will be $a_1 = k_m \frac{m_1}{m_1} = k_m$, and $a_2 = k_m \frac{m_1}{m_2}$. The acceleration ratio is $\frac{a_1}{a_2} = \frac{m_2}{m_1}$, which is equivalent to the ratio using the earlier equation 5.8. The relative acceleration between two objects is retained, while bringing the absolute values to a more reasonable range.

This modification produced more consistent interactions in the game. We noticed players were having an easier time understanding how to play, and playing the way they wanted. This better fulfilled the heuristic that “players should feel a sense of control over their ... interactions in the game world”. Overall, magnetism at this point was working well in terms of permitting emergence, matching player expectations, and even providing a sense of control. There was, however, one more adjustment we saw fit to make, to improve the experience even further.

5.1.8 Damping

Although many interactions worked well, and according to player expectations, some scenarios were essentially slow to stabilise. For example, consider the case where a closed room has both a ceiling and floor that are positively polarised. A box between them is subsequently polarised: this results in forces from the floor pushing it upwards, and forces from the ceiling pushing it downwards. As the object gets closer to the ceiling, it is pushed back downwards more strongly, and vice-versa. The result is that the box will oscillate between the two, until drag finally brings it to a stable configuration.

The consequence was that the system was slow to control. Effectively the player would need to wait for equilibrium, to avoid hectic movements of magnets. One of the heuristics of GameFlow states that "players should receive immediate feedback on their actions", which is an area we could improve on.

To do so, we implemented a form of damping. Prior to damping, acceleration from magnetism was applied as follows:

$$\vec{v_{new}} = \vec{v_{prev}} + \vec{a}\Delta t$$

With damping, acceleration is applied as follows:

$$\vec{v_{new}} = (\vec{v_f} + \vec{a}\Delta t)(1 - k\Delta t) + (\vec{v_{prev}} - \vec{v_f})$$

where $\vec{v_f} = \vec{v_{prev}} \cdot \hat{a}$, and k is a variable to tune. We chose $k = 0.5$, as it provided effective damping of oscillation in our tests, without overly changing other behaviours emerging from magnetism. As we hypothesised, the result was that many interactions, such as magnets oscillating between two other magnets, stabilised much faster. Although evaluating its success using heuristics or playtesting was difficult, we felt as developers that the change improved controllability and feedback.

5.2 Supporting Mechanics

The central mechanic in "Plusminus" is the magnetism. Several other mechanics, however, were developed to support the emergent behaviours produced by the magnetism, to create more engaging gameplay, and to adhere to player expectations. Some relevant ones are as follows:

- Damage from collisions: enemies, and other damageable objects, are damaged from fast collisions. The faster the collision, the higher the damage.
- Platform movement: the player avatar will inherit the velocity of any object it is standing on.
- Wheeled enemy movement: wheeled enemies can produce a force forward, when their wheels are all grounded. They can also produce a torque to turn if the same condition is met.

- Flying enemy movement: the flying enemy has a jet that pushes it upwards, to keep it floating. It tilts to direct the jet, allowing it to move around.
- Pressure plates: these plates, or buttons, require a constant force acting on them, to stay depressed.

None of the aforementioned mechanics are directly related to magnetism. Instead, they are all built upon the physics engine, which acts as the bridging element between the mechanics. If a flying enemy is magnetised, such that it attaches to a magnetised box, the flying enemy will end up pushing the box around with its jet. Similarly, if the player sets up two metal boxes on top of each other, jumps on top, and magnetises both boxes, the upper box will go flying, with the player on top.

If we further consider combat, there are countless options for players to take when considering the damage mechanic combined with magnetism mechanics. A player can magnetise a box, then an enemy, such that it crashes on top of the box. Next, the player flips the polarity of the box, causing the enemy to crash into the roof, before it falls down and crashes into the floor. Each crash causes damage to the enemy, allowing it to be defeated. An alternative strategy might be to push an enemy around with magnetism, such that it falls down into a pit. A third option is to make enemies crash into each other, thus causing damage to both, simultaneously. The simple mechanics produce a large possibility space for players, which helps provide a sense of agency and control.

One of the categories of heuristics covered by GameFlow is that of Player Skills. Moreover, we identified two that are relevant to mechanics: “players should be able to start playing the game without reading the manual”, and “game interfaces and mechanics should be easy to learn and see”. Since the mechanics of “Plusminus” are all simple, they are easy to learn. Additionally, since they have a basis on real physics, they can be understood without reading a manual. Thus, not only do the mechanics support emergence, their simplicity also improves the experience by supporting player skill development.

Chapter 6

Discussion

The design of the mechanics in “Plusminus” went through multiple iterations. Some of the encountered problems are unique to developing magnetism mechanics, but others are likely to be generalisable. As noted in the previous chapter, the equations for magnetism are close to the equations for gravity, a mechanic used in games such as “Super Mario Galaxy”. However, many of the solutions are likely to be useful in a other scenarios too, when computing forces on objects, and designing rules of interaction in a physics-based game.

Before we consider the generalisability of the problems and solutions, we consider the success of our mechanics via concrete examples of emergence seen in playtests, followed by evaluating which aspects of the mechanics of “Plusminus” are still lacking in regard to emergence and providing players with a sense of control. We consider playtests, and heuristics, for the evaluation—also noting the limitations of our chosen evaluation criteria.

Figure 6.1: The first challenge players encounter is an area too high to reach by simply jumping. Using magnetism, two sets of stairs can be positioned such that the player can get up there.

6.1 Examples of Emergence

Throughout the design of “Plusminus”, we designed problems with clear solutions. Since the aim was to encourage emergent gameplay, we were careful to avoid adding too many restrictions, and included enough props in levels to allow experimentation. Many players were content with moving from one level to the next using the designed solutions, while others were more prone to experiment. In this section, we present a few of the problems we designed, and how players were able to surprise us with unexpected solutions.

6.1.1 Sticking Objects to Walls to Create Platforms

As players start “Plusminus”, the first room they reach has a simple task: move upwards. The intended solution is to magnetise a pair of metallic stair-objects, as seen in figure 6.1. When they attract each other, they provide a platform for players to use to jump up to a higher level. One player, however, discovered an alternative path up, using the provided mechanics to their advantage.

Figure 6.2: A player realises that shooting a polarising projectile at a wall creates a temporary magnetic field—which can be used to stick objects to walls. The player uses it to attach a box to the wall to create a temporary platform, allowing them to reach places higher up.

Although the main usage of shooting projectiles is for polarising metals, there is a secondary effect if they hit a non-metal instead. In such a case, the projectile sticks to the targeted spot for approximately five seconds, producing a temporary magnetic field. This mechanic, combined with the mechanics of attraction and repulsion, allows for a dynamic where a metal object can be moved around by shooting projectiles on nearby surfaces.

The player discovered this dynamic, and also realised that this could be used to stick objects, temporarily, to walls. The tutorial area has some boxes to experiment with, to acclimatise users to the magnetism mechanics. The player stuck one of these boxes to a wall, and used it as a platform to reach the higher level, as shown in figure 6.2. In this way, the player bypassed the designed solution, and was able to use the simple mechanics to come up with a new surprising solution.

6.1.2 Catapulting Out of Bounds

Figure 6.3: A designed solution to a problem. The player needed to move a box (shown in red) on top of a conveyor belt, which would move it to the correct location to produce a path forward.

Not all of the examples of emergence in playtests occurred as alternative solutions to problems. One of the designed problems included a stack of boxes, near a conveyor belt. The designed solution was for the player to move one or more of the boxes on top of the conveyor belt, using magnetism. The box would then move up against a wall, as shown in figure 6.3, where the player could then jump on top of the box and on to a higher level.

One player discovered an alternative use for the stack of boxes, shown in figure 6.4. The player polarised one of the boxes at the bottom of the stack positively. They then climbed on top of the stack, and polarised the box they were standing on, also positively. The result was a force of repulsion pushing the box they were standing on upwards. This enabled them to catapult up

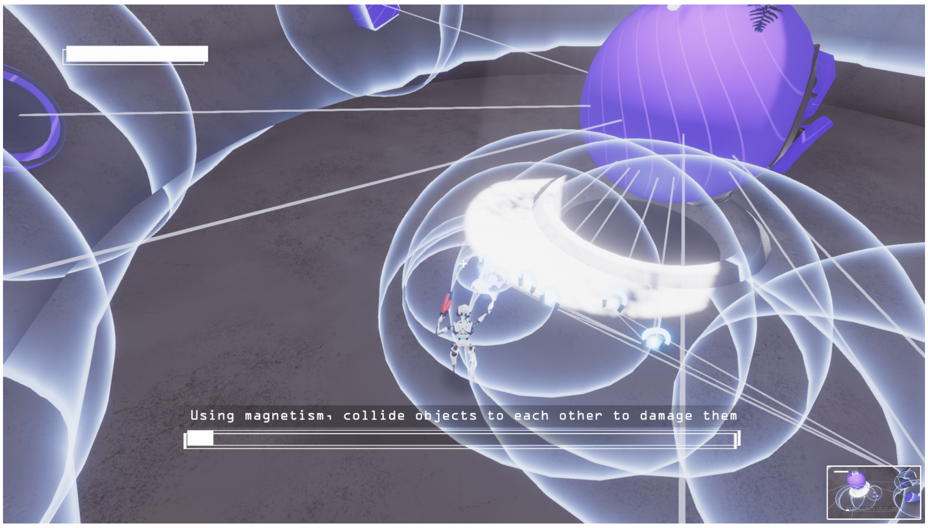
Figure 6.4: One player found an exploit to get out of bounds. By polarising two stacked boxes positively, the produced force of repulsion between them sent the upper box flying. By standing on top of the box as this happened, the player was able to catapult themselves high up—letting them escape the bounds of the level.

high enough to get over the wall of the level, and get out of bounds. Getting beyond the bounds of the level was not intended; it was, however, an instance of emergent gameplay.

6.1.3 Displacing Objects With Repulsion

The force produced between two objects is limited, as discussed in section 5.1.7. However, with several pairs of objects acting on one another, the cumulative forces can be considerably higher. This proved to enable surprising solutions.

One of the problems players are presented with, is a large immobile enemy stuck in a hole in the ground. Players need to jump down into the hole, but must first remove the enemy to do so. The designed solution is to produce enough high-velocity collisions aimed at the enemy, to damage and destroy it. Some players, however, found that with enough repelling forces, the enemy could in fact be rolled out of the hole, as shown in figure 6.5. With a combination of polarised objects, and projectiles shot on the ground, players can move large objects such as the enemy. As with the previous examples, this was surprising and not designed for. The simple design of the mechanics, and their basis on physics, allow for emergent solutions.



Using magnetism, collide objects to each other to damage them

Figure 6.5: A large spherical enemy blocks a hole in the ground. Typically, players would need to impart collisions onto the enemy, to damage and destroy it. One player discovered that with enough objects repelling it, it would roll out of the hole, providing an alternative solution to the problem.

6.2 Limitations

Although the iterative design process continually improved mechanics, and the chosen evaluation techniques were useful and appropriate [6], there are still a number of design problems to solve. These issues relate to both emergence and control. In terms of emergence, the major shortcoming of this thesis is in evaluation of the developed game mechanics: it was performed mainly through qualitative analyses of gameplay. One method which would have been appropriate for evaluating the design for emergence, would have been a quantitative comparison of players' behaviours [6]. Here, the paths and options players take in various scenarios would have been tallied. It could be hypothesised that the greater the distribution of paths players take, the more successful the design for emergence. Another option might be to consider the possibility space more directly. For example, from a given situation, how many meaningfully different options are available for a player, within a given timeframe? The duration of the timeframe would dictate the level of emergence being inspected: is it first, second, or third order [33].

It should also be noted that, while the methods of evaluation were chosen such that they could distinguish weaknesses in mechanics from weaknesses in level design and other aspects of gameplay, at many times the distinguishment was still not clear enough. Playtests, for example, were performed such that there were often multiple changes from previous playtests, including changes to mechanics, levels, and narrative. This made it harder to discern which changes were having an effect on the player experience. More frequent playtests, with less overlapping changes, would have helped make it clearer.

One particular aspect of control was hard to evaluate. Some players were able to very quickly grasp the controls of the game—we identified that they were typically players with previous experience playing action games. Players without this experience, though, often had a difficulties figuring out how the aiming with the camera worked. The problem might be alleviated by designing mechanics which ease aiming, such as including some aim-assistance. Yet, at the same time, since some players had more success with the controls, it would also suggest that better tutorialisation could improve controllability for inexperienced players.

Apart from issues with the evaluation, the latest version of "Plusminus" still has some general problems in providing control. Although we were able to fulfil many of the heuristics of GameFlow to a solid degree, many of the situations in which we were able to produce a sense of control fell apart when multiple polarised objects were present. Players tended to have the most difficulties playing in scenarios where multiple active magnetic fields were in close proximity, as it was hard for them to predict how magnetised objects would act when interacting with multiple other objects. Additionally, with lots of polarised objects, the bubble visualisation covered in section 5.1.5 became a problem. The number of bubbles displayed at once on the screen made it difficult to discern what was going on.

Not only did the bubble visualisation cause problems, there was also a lack of other visual cues for the player to use. For example, when a player was looking to make two attracted objects repel each other, it was hard to understand how fast, and in what directions they would travel. An additional visualisation to display a trajectory as the player is aiming at the object could hypothetically alleviate the issue.

Limitations with our method still exist: our methods of evaluation could be improved, and mechanics, particularly in terms of providing appropriate visual information to players, could be improved. The evaluation methods, however, were successful in providing necessary information for our design process, particularly as we considered "Plusminus" to be at the prototyping stage of development. When moving forward to further development, more rigorous evaluation methods should be considered.

6.3 Solutions Adapted to a Wider Context

In this section, we consider the solutions to problems in “Plusminus”, covered in the previous section, in a more general context. As our information is limited to solutions in a single game, we cannot provide design guidelines, but we do provide suggestions based on learnings from “Plusminus”, and our knowledge of other physics-based games. We hypothesise that many of our solutions are applicable to situations where physical forces need to be evaluated in a manner that matches player expectations, while providing improved control compared to a realistic approach.

6.3.1 Use Alternative Distance Falloff Equations to Limit Values

The first issue we encountered was with the inverse square proportionality of force and distance. The result was forces growing to infinity as distances decreased, which caused instabilities and poor controllability.

The issue is not unique to magnetism, or gravity. If we consider point lights, common in 3D engines, they have the same issue: intensity is inversely proportional to the square of the distance to an illuminated point, i.e. $I = \frac{k}{r^2}$. The result is an incredible intensity at short ranges. As such, alternative equations are often used for point lights. One such equation considers the lights as sphere-lights instead of point-lights, with the following equation:

$$I = \frac{k}{\frac{r}{d} + 1}$$

In this case, d is the radius of the sphere-light. This is similar to equation 5.2, which we used to approximate magnetic forces. The problem, and solution, then, are likely to be useful in any situation where an approximation needs to be made in the context of an inverse proportion to a squared value—something which is prominent in physics.

6.3.2 Limit and Visualise Range to Improve Player Understanding

Although forces such as magnetism and gravity reach to infinity, this might not be as effective for gameplay. As we discovered, restricting the range of a force can provide players with a better understanding of when they are effective, and when they are not. This also provides the opportunity to visualise the range of the force, which would be much harder considering an infinite range.

Limiting range is sure to be useful outside of physical forces, too. For example, a hot area that damages players, might be easier to communicate, if there is a clear border where it begins and ends. A limited range is likely easier to communicate to the player.

6.3.3 Subdivide for Accuracy

As we considered magnetic interactions, we noted that the simplistic model of applying forces to only a single point on objects did not produce torques, which would be expected from real physics. The solution was to subdivide objects into multiple points that would be considered.

One example of an application might be in relation to ocean currents. Consider a submarine moving underwater. We might consider any currents under water as acting on the centre of the submarine. Thus, it would get pushed around by the currents, but it would never rotate because of the currents. If, instead, the currents acted on the front, centre, and back, the simulation would be more exciting. As the front of the submarine would enter a current, it would begin rotating as a result, but would experience no more torque once the full submarine is in the current. Upon exiting, it would again experience a torque, as the front is out of the current, but the back is not.

In many cases, especially if torques should be considered as a dynamic emerging from mechanics, it can be useful to consider subdividing objects, and computing the forces per subdivision.

6.3.4 Reduce the Number of Variables for Easier Balancing

The more adjustable independent variables a system has, the harder it will be to balance. With “Plusminus”, we made the generalisation that mass and charge are proportional, which reduced variables for magnetic interactions from 6 to 4. Furthermore, it can help to consider the relative sizes as opposed to absolute sizes: in “Plusminus”, the lighter object always experiences the same acceleration, regardless of the mass of the heavier object. The heavier object, on the other hand, experiences an acceleration that is scaled down based on their relative proportion. The result is that forces can be balanced once, and the result will remain balanced for objects of all proportions. The interactions are consistent, even if not entirely realistic. Most importantly, controllability was significantly improved, as we made generalisations to reduce the number of variables to adjust.

6.3.5 Damp Movement for Faster Convergence

Many interactions in “Plusminus” were slow to stabilise, but simply damping velocities in the direction of forces applied helped speed up convergence in most interactions. This predictably improved controllability, and in general damping is likely to be an effective way of making objects more controllable. Notably, on top of damping produced by drag, we applied a separate damping for magnetism—this was done so that we could produce fast stabilisations with magnetic interactions, without interfering with more general behaviours, such as an object falling. The method is not realistic, but improves feedback and controllability, as the results of interactions are more immediate.

Chapter 7

Conclusion

The aim of “Plusminus” is to provide players with agency. To support this, mechanics were developed to promote emergence, thus providing a large possibility space. Furthermore, control for the player was provided by adjusting mechanics for stability and consistency. A large possibility space, combined with a feeling of control, are identified as criteria for attaining a sense of agency—which can help in reaching a state of flow.

A key part of providing enjoyment in physics-based games, in particular, is in creating mechanics that match player expectations of physics from the real world. This does not necessarily mean strictly adhering to realism, as games such as “Portal” and “Super Mario Galaxy” demonstrate. The physics should still have characteristics from real physics, and importantly, should be consistent. A key component in designing the mechanics in “Plusminus” was related to maintaining this consistency, and matching expectations.

The iterative design process of “Plusminus” identified and solved many problems with designing mechanics for emergence and control, in a physics-based game. The primary methods used to achieve this include:

- Monopole magnets instead of dipole magnets: magnetic forces are calculated between monopoles, using Coulomb forces as a basis. The forces are modelled as occurring between magnetically charged shells, and charged particles. The computed forces share characteristics from actual magnetic fields, while being simplified to better suit the gameplay of “Plusminus”. This includes adding a range modifier that retains the characteristics of magnetism, while enabling magnets to act at long distances in a stable manner.

- Visualising and clamping the range of magnets: magnets have a limited range, visualised by a transparent bubble. Two objects will only attract or repel once their respective field bubbles overlap. This improves the predictability of dynamics in "Plusminus", ensuring interactions can be planned precisely by both players and level designers. It also prevents objects in separate game areas from interacting uncontrollably.

- Reducing variables for easier balancing: typically, magnetic forces would be computed from the product of the charges between two interacting objects, and the resulting accelerations would also depend on masses. This can be hard to balance such that the resulting simulation is stable and predictable, as the number of combinations of different masses and charges is high. With "Plusminus", we made two important changes to combat this: firstly, we decided that magnetic charge is directly proportional to mass, removing the need to balance magnetic charge values. Secondly, the force between two objects is computed such that it depends only on the lighter object. The result is that, on top of distance variables, the accelerations produced depend only on mass ratios as opposed to absolute masses. As a result, once balanced, interactions remain stable regardless of scale differences, while still retaining the important characteristics of magnetic forces.

We additionally used subdivision to improve the accuracy of the simulation, and used damping to improve controllability and speed up the convergence of interactions.

Overall, these solutions developed during the iterative design process of "Plusminus" were successful in improving controllability without diminishing opportunities for emergence. Future work could help ascertain the extent to which the solutions could be applied in other contexts, as well as explore further avenues for iterating the mechanics of "Plusminus" to better encourage emergence and improve control.

Bibliography

[1] Get ready for the egw 2019! `http://www.experimental-gameplay.org/blog/5g22dlechea99wedwd29f37l7zz37f`. Accessed: 2019-06-12.

[2] CHI Play 2018 Student Game Design Competition Jury Prize. `https://www.research.aalto.fi/fi/prizes/chi-play-2018-student-game-design-competition-jury-prize(40e2b9cd-65a4-47d1-9fb8-c6f037b058d9).html`, 2018. Accessed: 2019-06-12.

[3] Alexander, L. GDC 2012: Sid Meier on how to see games as sets of interesting decisions.

[4] Babadi, A., Naderi, K., and Hämäläinen, P. Intelligent middle-level game control. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)* (2018), IEEE, pp. 1–8.

[5] Bennet Foddy. *QWOP*. Game [browser], 2008.

[6] Bernhaupt, R. User experience evaluation in entertainment. In *Evaluating user experience in games*. Springer, 2010, pp. 3–7.

[7] Brown, P. New tales from the making of zelda: Breath of the wild. `https://www.gamespot.com/articles/new-tales-from-the-making-of-zelda-breath-of-the-w/1100-6449174/`, April 2017. Accessed: 2018-08-22.

[8] Choi, D., and Kim, J. Why people continue to play online games: In search of critical design factors to increase customer loyalty to online contents. *CyberPsychology & behavior 7*, 1 (2004), 11–24.

[9] Conway, J. The game of life. *Scientific American 223*, 4 (1970), 4.

[10] Coppin, B. *Artificial intelligence illuminated.* Jones & Bartlett Learning, 2004.

[11] Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience (P.S.)*, 1st ed. Harper Perennial Modern Classics, 2008.

[12] Desurvire, H., Caplan, M., and Toth, J. A. Using heuristics to evaluate the playability of games. In *CHI'04 extended abstracts on Human factors in computing systems* (2004), ACM, pp. 1509–1512.

[13] Di Fratta, G. The newtonian potential and the demagnetizing factors of the general ellipsoid. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 472*, 2190 (2016), 20160197.

[14] Fendt, M. W., Harrison, B., Ware, S. G., Cardona-Rivera, R. E., and Roberts, D. L. Achieving the Illusion of Agency. In *International Conference on Interactive Digital Storytelling* (2012), Springer, pp. 114–125.

[15] Gygax, G., and Arneson, D. *Dungeons and Dragons*, vol. 19. Tactical Studies Rules Lake Geneva, WI, 1974.

[16] Hartman, C., and Benes, B. Autonomous boids. *Computer Animation and Virtual Worlds 17*, 3-4 (2006), 199–206.

[17] Hidemaro, F., Satoru, T., and Takuhiro, D. Breaking conventions with The Legend of Zelda: Breath of the Wild. `https://www.youtube.com/watch?v=QyMsF31NdNc`. Accessed: 2019-06-12.

[18] Hunicke, R., LeBlanc, M., and Zubek, R. Mda: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI* (2004), vol. 4, p. 1722.

[19] id Software. *Quake*. Game [PC], June 1996.

[20] Interactive, T.-T. Kerbal space program.

[21] Kaos, M. *The Role of Agency in the Design and Play of Exercise Video Games*. PhD thesis, 2018.

[22] Maxis. *The Sims*. Game [PC], February 2000.

[23] Metacritic. Best video games of all time. `http://www.metacritic.com/browse/games/score/metascore/all/all?view=condensed&sort=desc`. Accessed: 2019-06-14.

[24] Nabi Studios. *Toribash*. Game [PC], 2006.

[25] Nakamura, J., and Csikszentmihalyi, M. The concept of flow. In *Flow and the foundations of positive psychology.* Springer, 2014, pp. 239–263.

[26] Nintendo Creative Department. *Super Mario Bros.* Game [NES], September 1983. Nintendo, Kyoto, Japan. Played August 2018.

[27] Nintendo EAD Tokyo. *Super Mario Galaxy.* Game [Wii], November 2007. Nintendo, Kyoto, Japan. Played November 2007.

[28] Nintendo EPD. *The Legend of Zelda: Breath of the Wild.* Game [Switch], March 2017. Nintendo, Kyoto, Japan. Played March 2017.

[29] Reja, U., Manfreda, K. L., Hlebec, V., and Vehovar, V. Open-ended vs. close-ended questions in web questionnaires. *Developments in applied statistics 19*, 1 (2003), 159–177.

[30] Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH computer graphics* (1987), vol. 21, ACM, pp. 25–34.

[31] Salen, K., Tekinbaş, K. S., and Zimmerman, E. *Rules of play: Game design fundamentals.* MIT press, 2004.

[32] Schiff, J. L. *Cellular automata: a discrete view of the world*, vol. 45. John Wiley & Sons, 2011.

[33] Sweetser, P. *Emergence in games.* Cengage Learning, 2008.

[34] Sweetser, P., and Johnson, D. Player-centered game environments: Assessing player opinions, experiences, and issues. In *International Conference on Entertainment Computing* (2004), Springer, pp. 321–332.

[35] Sweetser, P., and Wiles, J. Scripting versus emergence: issues for game developers and players in game environment design. *International Journal of Intelligent Games and Simulations 4*, 1 (2005), 1–9.

[36] Sweetser, P., and Wyeth, P. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment 3*, 3 (July 2005), 3.

[37] Takeda, H., Veerkamp, P., and Yoshikawa, H. Modeling design process. *AI magazine 11*, 4 (1990), 37–37.

[38] Terzopoulos, D. Artificial life for computer graphics. *Communications of the ACM 42*, 8 (1999), 32–42.

[39] Valve Corporation. *Portal*. Game [PC], October 2007. Valve, Washington, US. Played April 2019.

[40] Wolfram, S. Cellular automata as models of complexity. *Nature 311*, 5985 (1984), 419.

[41] Woodcock, S. Games making interesting use of artificial intelligence techniques. `http://web.archive.org/web/20001120124700/http://www.gameai.com/games.html`, 2006. Accessed: 2019-06-02.

[42] Wright, W. Dynamics for designers. `https://www.gdcvault.com/play/1019938/Dynamics-for`, 2003. Accessed: 2019-06-05.