Bachelor’s thesis

Information and Communications Technology

2025

Frederik Aun

Replicating Immersion with Godot 4: Implementing an Interaction System Inspired by FrictionalGames

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2025 | 68 pages

Frederik Aun

Replicating Immersion with Godot 4

- Implementing an Interaction System Inspired by FrictionalGames

The video game studio FrictionalGames is recognized for its distinct physics-driven interactions, yet a significant lack of detailed documentation for its implementation techniques is noted. Hence, the objective of this thesis is to provide practical documentation and insights for developers interested in the replication of similar interaction mechanics.

As part of the methodology, key interaction mechanics (grabbing, doors, wheels, levers) within the FrictionalGames' system were observed and then systematically prototyped within Godot 4, utilizing the Jolt Physics engine. Extensive iterative testing and refinement were conducted, with a focus on parameter tuning and intuitive input mapping.

Based on the comparative analysis, high mechanical accuracy was confirmed for all interaction mechanics besides doors. The force application and input mapping were considered the primary deviations of the door mechanic.

Overall, the replication was achieved to a satisfactory degree, with Godot 4 shown to be a viable tool for the implementation of complex interaction systems aimed at enhancing player immersion.

Keywords:

game immersion, game physics, FrictionalGames, Godot engine, interaction design

Opinnäytetyö (AMK) | tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2025 | 68 sivua

Frederik Aun

Uppoutumisen jäljittely Godot 4:llä

- FrictionalGamesin innoittaman vuorovaikutusjärjestelmän toteuttaminen

FrictionalGames käyttää peleissään erottuvia fysiikan ohjaamaa vuorovaikutuksia. Niissä on kuitenkin merkittävä puute toteutustekniikoiden yksityiskohtaisessa dokumentaatiossa. Tämän opinnäytetyön tavoitteena oli laatia käytännönläheistä dokumentaatiota ja tietoa kehittäjille, jotka ovat kiinnostuneita vastaavien vuorovaikutusmekaniikkojen jäljittelystä.

Osana menetelmää tarkkailtiin ja analysoitiin FrictionalGamesin-järjestelmän keskeisiä vuorovaikutusmekaniikkoja (tarttuminen, ovet, pyörät, vivut). Siten ne prototyypitettiin järjestelmällisesti Godot 4:n sisällä Jolt Physics -moottorin hyödyntäen. Laajaa iteratiivista testausta ja hienosäätöä suoritettiin; keskityttiin parametrien virittämiseen sekä intuitiiviseen syötteidenkartoitukseen.

Vertailuanalyysin perusteella korkea mekaaninen tarkkuus vahvistettiin kaikille vuorovaikutusmekaniikoille paitsi oville. Voimankäyttö ja syötteidenkartoitus katsottiin ovimekaniikan ensisijaisiksi poikkeamiksi.

Kokonaisuutena jäljittely onnistui tyydyttävästi, ja Godot 4 osoittautui käyttökelpoiseksi välineeksi monimutkaisten vuorovaikutusjärjestelmien toteuttamiseen, joiden avulla pyritään tehostamaan pelaajan uppoutumista.

Asiasanat:

peliin uppoutuminen, pelifysiikka, FrictionalGames, Godot-pelimoottori, vuorovaikutussuunnittelu

Contents

Figures

1 Introduction

Video games in 2025 stand as one of the most influential forces in global entertainment. Having grown exponentially from a niche hobby into a mainstream phenomenon (Hadzinsky 2014). The gaming industry was estimated in 2024 to be around $241.1 billion, with an estimation of it reaching $535.29 billion by 2033 (IMARC Group 2024). A large portion of this can be attributed to the smartphone penetration of society, which enables billions of mobile users to experience video games (Hadzinsky 2014; IMARC Group 2024). Furthermore, video gaming is known to engage with players from all age groups, genders, and backgrounds (ESA 2024). Hence, enabling anybody with a compatible platform, such as consoles, personal computers, or the aforementioned mobile devices, to enjoy video games. Because of the massive scale and constant growth of the industry, technological innovation remains a primary driver in the evolution of video games (Hadzinsky 2014), primarily with a focus to improve development practices and player experiences.

A recurring central goal for innovation in player experiences is achieving deep immersion through a combination of visual fidelity and interactivity (Sweetser et al. 2005; McMahan et al. 2011). While visual fidelity is often prioritized in modern game development, it is the compelling sense of presence induced by interactivity that truly defines player immersion (Slater 2009). One method used to achieve this sense of presence is through physics-driven engagement with virtual objects and environments (Swink 2008). This stands as an efficient technique to foster a powerful feeling of agency and belief within the player (Slater 2009; Calleja 2011). It is a distinct style of engagement popularized and pioneered over the years by FrictionalGames.

In current day, advanced physics engines have become widely available and built into all major game engines, giving any developer the opportunity to experiment with advanced physics. The increasing accessibility and power of these physics engines have played a major role in improving game immersion. Physics engines significantly amplify immersion by delivering a realistic and grounded experience that players can believe in. Academic research supports this, identifying the 'kinesthetic involvement' offered by physics-based interaction as a key factor in fostering player presence and engagement within the virtual world (Cairns et al. 2014).

This thesis explores the technical implementation of physics-driven interaction mechanics inspired by those seen in titles by FrictionalGames. The aim is to replicate these mechanics within the game engine Godot 4; with the expected result being practical documentation, which would offer guidance and useful insights for developers interested in creating more interactive and immersive experiences using accessible and open-source tools.

To accurately replicate Frictional Games’ specific ‘feel’ and intuitive control of the physics interactions requires meticulous parameter tuning (mass, friction, damping) and object-specific constraint configuration (Grip 2008). The precise technical details of their implementation, such as joint configurations, constraint parameterization, and input-to-force mapping, remain largely undocumented publicly. This lack of specific documentation, coupled with a lack of detailed public methodology, creates a specific knowledge gap.

The gap addressed by this thesis aims to not only replicate the nuanced physics-driven interactions by FrictionalGames, but also to serve as an exploration of physics-driven interactions within Godot 4. Alongside the additional goal to identify the effective techniques for creating believable virtual object manipulation that feels intuitively connected to the player's actions.

With FrictionalGames' work serving as the main reference and benchmark, the thesis will explore how Godot 4, coupled with the Jolt Physics engine can be effectively used to replicate the physics-driven interaction system characteristic of FrictionalGames' immersion design. Though the potential scope of physics interactions is broad, this research will focus on replicating the key mechanics often present in FrictionalGames' titles: direct object manipulation, interactions with hinged objects (e.g., doors, levers), and interactions with objects featuring rotational mechanics (e.g., valves) (Grip 2008; Grip 2010).

The core methodology involves:

1. Analyzing and deconstructing the key interaction types found in FrictionalGames titles through observation.
2. Systematically building prototypes within Godot 4 (utilizing the benefits of the Jolt Physics engine) aimed at replicating the key interactions types such as grabbing objects, opening doors, and turning levers / wheels.
3. Testing and refining the prototypes iteratively, focusing on parameter tuning, constraint application, and accurate input mapping that feels intuitive and believable.
4. Analyzing results to identify the most effective techniques and strategies, with the goal of generating practical knowledge on potential reusable patterns or guidelines on creating Frictional-style physics interactions within Godot 4.

To accomplish these objectives, the remainder of this thesis will be structured as follows. Chapter 2 establishes the foundational concepts, also known as the Theoretical

Framework. This covers the theory of immersion, interaction design, and basic game physics, which will provide an understanding for the implementation. This is followed by Chapter 3 which provides an in-depth analysis of the Example System from FrictionalGames. The main focus is on their interaction model and its functionality as the benchmark. Chapter 4 is a summary and review of the author's Previous Attempt (Aun 2023), which discusses the lessons learned from earlier development phases. This provides the necessary context to the techniques presented in the practical work. The core of the implementation within Godot 4 is discussed in Chapter 5. This is where the approaches to the different interaction types are explained, as well as the challenges encountered. Subsequently, Chapter 6 presents the Comparative Analysis, which is conducted across all implemented features. The thesis is then concluded in Chapter 7 with a reflection of the journey and propositions for Future Work.

2 Theoretical Framework

This chapter sets out to provide the foundation of knowledge necessary to understanding the thesis' objective and its practical implementation. The core aim of this thesis is to replicate a Frictional-style interaction system within Godot 4 with the Jolt Physics backend, with the focus being on its capabilities at encouraging player immersion. Thus, the discussion of theory stands critical as it will define immersion, its related properties, and clarify why FrictionalGames' approach works. Additionally, it establishes the evaluation criteria for use in analysis within subsequent chapters.

This chapter will delve into three theoretical areas. Beginning with an exploration of theories on player immersion, what causes this state and how it is induced within a video game. This is followed by an examination of the relevant principles from interaction design, with the focus being primarily on intuitive, physics-based controls. The final section provides the relevant foundation to game physics, explaining the functionality of objects, their constraints, and their applied forces during real-time physics simulations. These areas provide the necessary foundation to understanding the framework of the system, and how it was implemented.

2.1 Theories of Player Immersion

Immersion in video games usually refers to the state of dissociation from the physical environment that a player experiences during gameplay (Cairns et al. 2014; Domingues et al. 2023). In its essence, immersion is the extent to which a system, like video games, is able to provide a believable illusion of reality to the player's senses (Slater et al. 2013; Carpio et al. 2023) and is frequently considered as a key goal in game design, especially in creating engaging and compelling video games (Sweetser et al. 2005). Hence, its relevance for the thesis topic lies within FrictionalGames' design philosophy and physics-driven interaction model present in their titles. The main argument proposes that allowing players to manipulate virtual objects with simulated physics stands as a powerful technique to foster immersion within video games. This is especially true when combined with visual fidelity (McMahan et al. 2011; Belga et al. 2025).

To achieve a better understanding of immersion and its complex layers, researchers have conceptualized various models. For instance, Mel Slater's framework proposes concepts of Place Illusion (PI), and Plausibility Illusion (Psi) (Slater 2009). These concepts are the cornerstones to a convincing, and deeply immersive experience. In practice, PI refers to the sense of presence the player feels within a virtual environment, while being entirely aware that in reality, they are not (Slater 2009). Psi stands for the believability of occurring events in an experience, such as virtual objects exhibiting realistic physics (Slater 2009). When both PI and Psi are strong, players tend to respond realistically to the experience. Displaying genuine emotional connection to the virtual events that are transpiring (Slater 2009; Slater et al 2013).

These concepts are foundational when discussing the effectiveness of FrictionalGames' physics-driven interaction model. Their design philosophy seems to be aimed at heightening these concepts in their titles (Grip 2013; Grip 2017a; Grip 2017b). As implementing believable physics interactions with intuitive and responsive controls, increases PI (Slater 2009; Norman 2013). This is due to it engaging with the player's natural sensorimotor systems, also referred to as 'kinesthetic involvement' (Slater 2009; Calleja 2011; Cairns et al. 2014; Abtahi et al. 2022). Kinesthetic involvement is the sense of connection that the player feels when their physical movements (e.g., mouse gestures) match their virtual body's movements (e.g., direct object manipulation) (Norman 2013)

Frictional's narratives also tend to impact the experience, as these strenghten Psi (Domingues et al. 2023). This occurs due to what is referred to as, 'narrative immersion' or 'narrative involvement' (Calleja 2011; Domingues et al. 2023). It is the state of immersion one feels for a story, its setting and characters. This is further enchanced by the realistic physics which ground the experience in the story's context (Swink 2008; Millington 2007; Walter 2023). Psi is additionally increased by their design decisions, such as the choice to exclude combat from Amnesia (Grip 2011). This allowed tensions to stay heightened, which further pulled the player into the experience (Grip 2011; Brown et al. 2004).

The active participation that the player engages in through interactions fosters player agency, which is referred to as the subjective experience of controlling one's actions and their consequences (Calleja 2011; Cairns et al 2014). Agency is reinforced through actions like the manipulation of an object via simulated physics, as this ensures a connection between the player's input and the perceived outcome (Swink 2008; Calleja 2011; Norman 2013; Cairns et al 2014). Thus, intuitive, and accurate controls are crucial for creating a feeling of ownership for the virtual body (Swink 2008; Calleja 2011; Norman 2013; Norrman 2020). Incorporating proper controls, alongside a proper 'feel' for the interactions strengthens the player's sense of presence and agency (Swink 2008; Calleja 2011; Cairns et al 2014). This contrasts with abstract controls (individual key presses for complex actions), which commonly form a greater psychological distance with the player, and overall have a decreased sense agency (Shneiderman 1997; Swink 2008; Calleja 2011; Cairns et al 2014). Hence, mimicking elements from the real-world is essential to grounding the player in the experience (Swink 2008; Calleja 2011; Cairns et al 2014).

While immersion is complex and with only a few, questionable methods to measure its effectiveness (Kukshinov et al. 2025), proposed models like Mel Slater's, allow us to discuss and analyze its many components (Slater 2009). Based on Slater's model, it can be deemed that Frictional's physics-driven interaction system embodies a strong sense of immersion due to the strong existence of PI and Psi (Slater 2009). Through direct manipulation of the virtual objects, the player becomes connected and present in the virtual environment through 'kinesthetic involvement' (Calleja 2011; Cairns et al 2014). The realistic physics interactions that occur enable the player to be further grounded in the experience by strengthening Plausibility Illusion (Psi) (Swink 2008; Slater 2009; Calleja 2011; Cairns et al 2014)

This section should now provide a better understanding of immersion and factors that encourage it in video games. The concepts discussed in this section will serve as a foundation for the implementation presented in Chapter 5, and the following analysis summary presented in Chapter 6.

2.2 Principles of Interaction Design for Games

Interaction design is the bridge that connects the player to the game systems (Ishii 1998; Calleja 2011; Cairns et al. 2014). Its aim is to create a meaningful and intuitive engagement with the game. This is usually achieved by designing interfaces and game mechanics which are easy to understand and feature intuitive controls (Shneiderman 1997; Pagulayan 2003; Calleja 2011). The more seamless the experience is for the player, the more they are to be immersed by it. This is done by utilizing a user-central approach to game design, which aims to reduce friction and confusion for the player (Pagulayan 2003; Cairns et al. 2014). A successful implementation means the player can naturally immerse themselves into the game world and focus on its gameplay without any challenges (Pagulayan 2003; Brown et al. 2004; Calleja 2011).

To achieve a seamless experience for the player requires certain key principles to be considered when designing interactions. Particularly those that concern player perception and expectation (Swink 2008; Calleja 2011). Firstly, the player has to receive enough feedback on their interactions. These can be in the form of objects reacting appropriately to forces, sounds being emitted, or even haptic responses like vibrations on a controller or phone (Swink 2008; Cairns et al. 2014). These let the player know that their actions have impact on the game world. Secondly, consistency has be be maintained across the entire experience (Swink 2008). For example, if pushing a heavy crate requires significant effort, this should apply to all similar crates. Any inconsistencies in interactions would cause the player reevaluate and relearn the interaction, thus impacting immersion and potentially cognitive load (Sweetser et al. 2005; Swink 2008). Lastly, the principle to compliment the previous is intuitiveness. Intuitiveness stands to ensure that the player is able to learn the interactions without any cognitive effort (Shneiderman 1997; Pagulayan 2003; Swink 2008). This is achieved by designing controls around potential mental models and expectations that the player would already possess (Swink 2008; Norman 2013). The goal is to have controls that the player can use without a conscious thought, ensuring they focus on the experience itself (Pagulayan 2003; Sweetser et al. 2005; Swink 2008).

To achieve intuitiveness, developers can utilize a technique known as ‘natural mapping’ (Swink 2008; Norman 2013). This refers to controls that feature a logical connection to the action being performed within the game. For example, turning a virtual steering wheel by turning a physical wheel controller. For a more thesis related example, pushing/pulling a door with mouse gestures. Natural mapping stands to mimic actions from reality, that the player would already be aware of, in the virtual world (Swink 2008; Norman 2013). Therefore, it serves to strengthen immersion within an experience. This is backed by research which has displayed that through the use of such techniques, players feel more present and autonomous (McEwan et al. 2014; McEwan 2017). Leading to more immersive, enjoyable, and flow-inducing experiences (McEwan et al. 2014). This contrasts greatly with abstract controls which, although being easier to master for complex scenarios, lack the ‘natural’ feel of an action that players might be accustomed with (Swink 2008; Calleja 2011; McEwan et al. 2014; Shafer et al. 2014).

FrictionalGames’ physics-driven interaction model is deemed effective because their implementation of direct object manipulation lends itself heavily to natural mapping. Players have to enact interacting with doors, drawers, or valves for example, through mouse gestures which replicate these actions from reality (Swink 2008; Calleja 2011).

However, to truly absorb the player in the experience, the ‘feel’ of these interactions is crucial. When the occurring interactions feel tangible and weighty, it greatly enhances the player’s sense of presence (Swink 2008; Slater 2009). The interactions coupled with the lack of heads-up display (HUD) elements, further directs the player’s focus onto the environment (Calleja 2011; Cairns et al. 2014; Iacovides 2015; Norrman 2020). It enables players to experiment and truly get to know their surroundings, further strengthening their sense of presence. Atmosphere is another aspect very prevalent in FrictionalGames’ design. The sense of vulnerability that is induced from a slow and clumsy interaction is perfect for their horror titles (Boonen 2018; Smith 2023). It heightens tension, which is known to increase immersion (Cairns et al. 2014), as well as enforce the player avatar’s vulnerable setting (Smith 2023). Directly making the player embody the avatar they are controlling, which further connects them to the game world (Walter 2023).

Although physics controls sound like a superior approach, there are inherent trade-offs to using them when compared to traditional button prompts. To begin with, implementing a proper physics-driven interaction system requires serious technical effort (Swink 2008), as the physics have to be believable and modeling them based on existing human intuition is challenging (Tang 2024; Li 2025). Difficulty arises from the technical issues associated with simulated physics. They can be unpredictable or contain latency issues (e.g., input lag), which can undermine the player's sense of control (Swink 2008). Furthermore, if the physics are working properly, each interaction requires rigorous tuning of physics body parameters and PID Controller values, to achieve the appropriate ‘feel’ (Swink 2008). However, if successfully implemented, they provide an immense boost to immersion, as detailed in prior paragraphs.

Traditional button prompts on the other hand, are reliable, precise, and easier to master (Shneiderman 1997). Additionally, they are simpler to implement as they are the standard in the industry and thus included in most game engines by default. Based on the presented benefits, physics controls might seem like the inferior choice. However, the drawbacks to abstract button prompts primarily reside within their intuitiveness and impact on immersion (Shneiderman 1997; Iacovides 2015; Norrman 2020). When a player is prompted to open a door with an abstract prompt like ‘press E to open’, it makes the game world feel less tangibly interactive (Shneiderman 1997; Ishii 1998; Swink 2008). Thus, resulting an impact on immersion, as well as the player’s sense of agency (Sweetser et al. 2005; Swink 2008).

Ultimately, neither choice for controls is inherently superior, therefore it comes down to purely design intent that has to be done by designers; are they looking to deeply immerse the player in the game world, or do they need reliable and fast controls which beginners can easily pick up on? This all depends on their specific game, and what is the most appropriate fit for its gameplay.

In conclusion, this section highlights that successful interaction design builds upon principles like affordance, feedback, and intuitiveness (Norman 2013). Natural mapping standing out as key technique to enhance the player’s connection with the virtual environment (Swink 2008; Norman 2013). Especially in the context of physics-based interactions. FrictionalGames’ approach clearly shows an application of these principles, with the focus of the implementations being not only on believable physics interactions, but also on capturing a specific atmosphere (Boonen 2018). Understanding these principles is crucial in understanding why FrictionalGames’ interaction model produces its distinct immersive effects. The presented principles will be used within Chapters 5 and 6, to evaluate the implemented interaction system, and to determine its success at capturing said principles.

2.3 Fundamentals of Game Physics Simulation

Physics engines serve to simulate physical mechanics such as gravity, forces, and collisions within video games. They aim to provide believable real-time physics simulations, by favoring performance over strict accuracy (Johnson 2005; Millington 2007). They turn game worlds dynamic through engaging physical interactions, such as those created by FrictionalGames which immensely boosts the player's sense of presence by making virtual environments feel tangible (Swink 2008). This section serves to introduce the core physics concepts, and the necessary vocabulary to provide a better understanding of the implementation discussed in Chapter 5.

Within physical simulations, the foundational building block is often the 'rigid body' (Johnson 2005; Millington 2007). It represents the physical properties of a solid object, by considering its mass (resistance to linear motion), inertia (resistance to angular motion), and velocity, which dictate the behavior of the rigid body during the simulation. These bodies are usually grouped by types: Dynamic (entirely physics simulated), Kinematic (controlled through code), Static (not simulated at all, immovable) (Millington 2007).

For rigid bodies to interact with each other, the physics engine has to process what is known as 'collision detection' (Ericson 2004; Millington 2007). This identifies when and where two, or more objects made contact, and enables the physics engine to calculate appopriate responses. This process is often done with simplified, invisible 'collision shapes' (aka 'colliders') instead of complex visual meshes, which require immense computational power to simulate in real-time (Ericson 2004; Millington 2007). A visual comparison between simplified and complex collision shapes is presented in Figure 1.

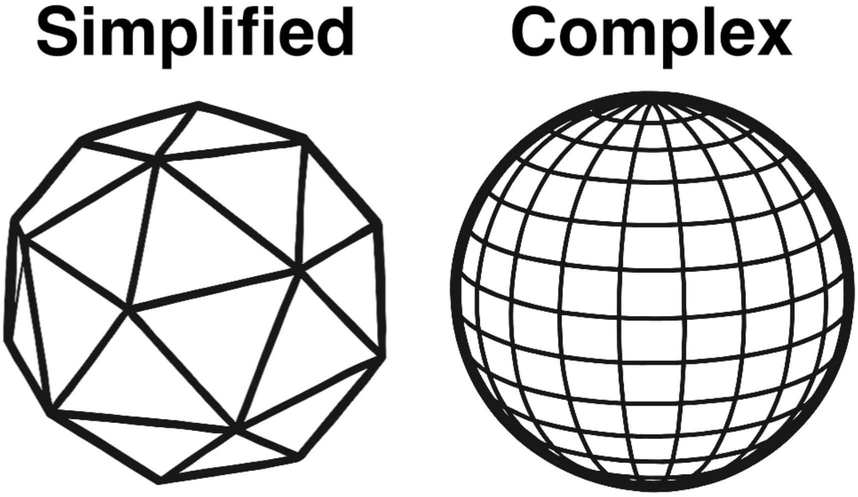


Figure 1. Simplifed vs. Complex collision shape.

The choice of collider for nuanced interactions like those present in FrictionalGames' titles, is paramount for believability. This is due to collisions forming the foundation of physical object behaviors (Baraff 1997b; Millington 2007).

To build more complex mechanisms with rigid bodies, 'joints' exist to constrain two rigid bodies together and restricts their relative motion (Seabra et al. 2023). Joints function by limiting the 'degrees of freedom' (DoF) which are available to the connected bodies (Baraff 1997a; Millington 2007). The degrees in this context refer to all the possible ways the rigid bodies can move, or rotate relative to each other (Baraff 1997a; Millington 2007).

Several types of joints exist, most commonly used is the 'hinge joint' (aka revolute joint), which limits rotations to only a single axis, such as a door hinge or lever pivot (Unity 2025). The 'slider joint' (aka prismatic joint) is quite similar, however it enables linear motion only along a single axis (Rapier 2025). Ideal for drawers or sliding doors for example. These joints limit DoF to simulate real-world constraints. Fundamentally, joints function by applying corrective forces or impulses to maintain the defined constraint during simulation (Baraff 1997b; Seabra et al. 2023). Joints are essential to creating interactive objects like doors, drawers, levers, or wheels that function similarly to their real-world counterparts. Especially when attempting to replicate physically interactive environments like those seen in FrictionalGames' titles.

As mentioned briefly before, forces and torque are used to manipulate rigid bodies (Baraff 1997a; Millington 2007). These forces abide by real-world physical laws, such as Newton's law. Specifically the principle which discusses the changes in motion that occur from an object being subjected to forces (Millington 2007). Physics engines simulate this using ’forces’ which alter an object’s linear velocity (object’s speed and direction) (Millington 2007). The magnitude of the motion is influenced by the mass of the object (Millington 2007). ’Torque’ works similarly but affects angular velocity (object’s rotation speed and axis) instead (Baraff 1997a; Millington 2007).

In the context of direct object manipulation systems like FrictionalGames’, mouse gestures are converted into forces and torques, which are applied to the object (Grip 2008; Grip 2010a). This method is preferred as it enables the physics engine to create natural motion, which appears more believable and tangible to the player (Millington 2007). Unlike setting object position or velocity each frame, which would result in inconsistent behavior and potential collision issues (Song 2022). The way input is translated into forces and torques, and applied to the object determines the overall ’feel’ of the interaction. Such as their percieved weight, responsiveness, and controllability (Swink 2008).

Establishing the setup of the aforementioned components (rigid bodies, colliders, joints) is only the beginning in implementing believable physics interactions, as each object has a certain ’feel’ associated to it (Swink 2008). This ’feel’ can be achieved through the careful ’tuning’ of various parameters, which is especially critical for nuanced interactions like those seen in titles by FrictionalGames (Swink 2008). The default parameter values rarely suffice, as believable game physics are often based on player perception and expectation rather than strict real-world accuracy (Johnson 2005; Millington 2007). Hence, they have to be tuned to behave believably. Tuning is an iterative design task, which involves adjusting parameters, testing them to see the results, and refining them until the desired effect is achieved (Swink 2008; Zook 2019). The key parameters adjusted during the tuning process are: ’mass’ (perceived weight and resistance to inertia) (Johnson 2005; Millington 2007), ’damping’ (controls how quickly motion settles) (Millington 2007), ’friction’ (resistance between sliding surfaces) (Millington 2007). Additional parameters such as ’restitution’ can also be adjusted within ’physics materials’ (Millington 2007). All these parameters are often interdependent, meaning if one is changed, others might need to be changed also (Millington 2007).

To achieve precise, stable control over physics-driven objects instead of applying basic forces, control system techniques, such as the Proportional-Integral-Derivative (PID) controller, are used (Åström 1995). This borrowed technique from engineering and robotics calculates corrective forces necessary to smoothly guide a rigid body towards a desired state (e.g., position, orientation, velocity) (Åström 1995). The core of a PID controller functions within a feedback loop; it measures the current state, compares it to the target point (the desired state), and calculates the error (the difference between them), then outputs a force or torque based on that error (Åström 1995). This feedback loop is illustrated in Figure 2. This allows a well-tuned PID controller to achieve significantly smoother and more stable control than applying just simple forces (Åström 1995), enabling the reliable ’feel’ for nuanced interactions that are present in FrictionalGames’ titles. However, its effectiveness relies heavily on the tuning of its ’gains’ (Kp, Ki, Kd), which dictate the influence of the corrective forces. Improper tuning will result in instability or poor responsiveness, requiring an iterative tuning process similar to core physics parameters (Åström 1995).

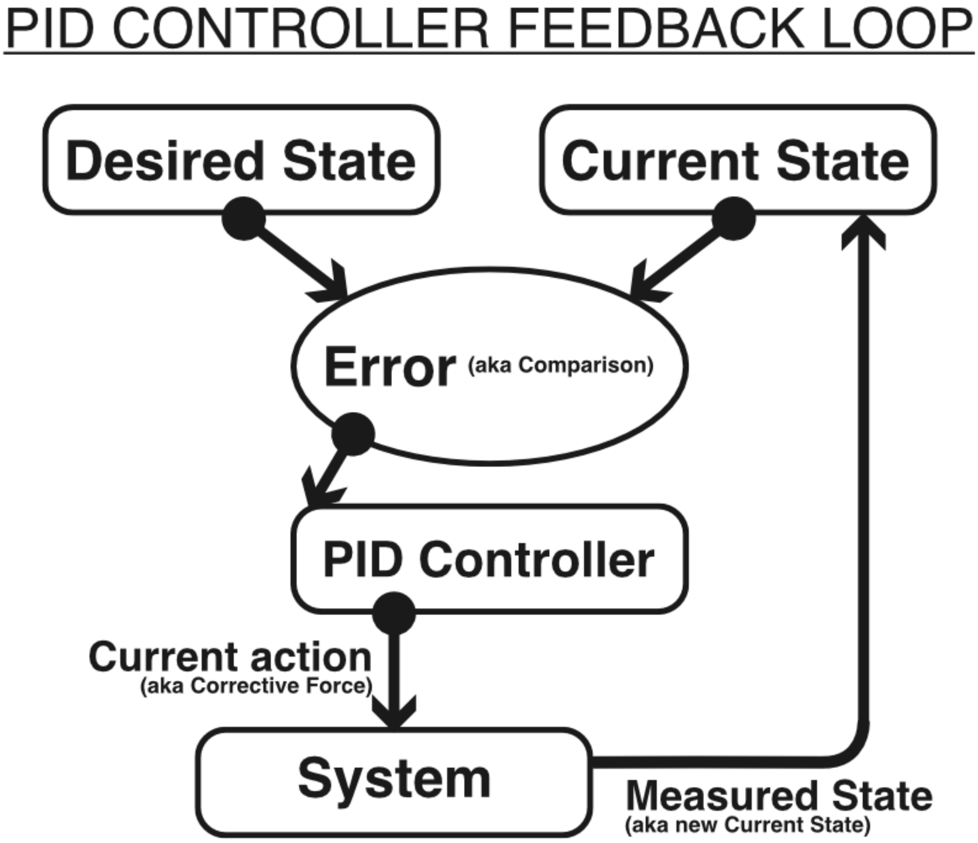


Figure 2. PID Controller feedback loop.

To summarize, physics engines provide the necessary foundation to simulate real-time physics within video games, utilizing core concepts such as rigid bodies, colliders, and joints (Johnson 2005; Millington 2007; Ericson 2004; Seabra et al. 2023). The motion of an object is simulated with forces and torques (Baraff 1997a; Millington 2007). The characteristic ’feel’ of the nuanced interactions is achieved through tuning physics parameters (mass, damping, friction), and potentially PID controllers (Åström 1995).

This section established the theoretical foundation of physics engines, and the necessary vocabulary to understand the implementation techniques used within Chapter 5, by utilizing these concepts within Godot 4 with the Jolt Physics engine (Hermansson 2023; Godot 2025).

2.4 Connecting Theory to Practice

This final section serves to bridge the gap between the established theories and the practical implementation detailed in Chapter 5. It connects the concepts of immersion, interaction design, and physics fundamentals to the thesis' objective of replicating the nuanced, physics-driven interaction system present in FrictionalGames' titles. This connection will clarify the reasoning behind the choices made, and approaches utilized during implementation. Additionally, it provides the foundation for the evaluation criteria in Chapter 6, which will analyze the effectiveness of the practical work based on the presented theories.

As established prior, player immersion stems from design decisions that enhance feelings such as presence, agency, and believability (Sweetser et al. 2005; Slater 2009; Norman 2013). The direct physical manipulation of a virtual environment, alongside naturally mapped controls, is a powerful pathway to foster these feelings within players (Swink 2008; Calleja 2011; Norman 2013). These principles frequently appear in titles by FrictionalGames, where they are used to create engaging and immersive atmospheres. Hence, the thesis' goal of implementing a Frictional-style physics-driven interaction system, serves to directly explore the link between theory and practice through a practical investigation into theory-backed design principles that help create immersive experiences.

Achieving the goal of physics-driven immersion within a working model, involves the use of the presented concepts as they form the basis for the object's behavior in the physics simulation. This requires setting up rigid bodies such that they reflect the attributes of the desired object, joints to constrain their behavior, and eventually to allow the player's mouse input to be converted into applicable forces or torques. Then, this output can adjusted to capture the specific 'feel' of the interaction through the careful tuning of physics parameters, potentially even the use of a PID controller for stabilization and smoothness (Swink 2008). Hence, why these concepts are foundational to understanding the implementation done within Godot, as there they appear as building blocks for physics objects.

Beyond just practical use, these theories stand to provide the framework for evaluation criteria for the analysis conducted later. The comparisons for each implemented mechanic being assessed against concepts like Place Illusion and Plausibility Illusion (Slater 2009), interaction design principles like natural mapping (Norman 2013), and the ‘feel’ of an interaction in regards to weight and responsiveness (Swink 2008). This helps connect practical results back to the theoretical goals of an immersive interaction model.

In conclusion, this chapter provided the theoretical framework for the thesis’ objective, as well as its implementation. It explored the components of immersion, different interaction design principles, and defined the necessary physics fundamentals to understand the implementation within Chapter 5. The presented concepts will be put to further use within the comparative analyses presented in subsections of Chapter 5, and ending with the summary of all results in Chapter 6. Additionally, they will aid in the discussion of the example system in the next chapter.

3 Example System

This chapter serves to provide the foundational understanding of the example system discussed within this thesis. Specifically, the discussion will be on the interaction model by FrictionalGames, and its ability to foster player immersion. The objective is to dissect their interaction model and understand the design philosophy, the key components, and technical aspects behind it. It is crucial to establish the foundation of this system as it will become the benchmark for the practical implementation work presented in the subsequent chapters.

FrictionalGames’ model was chosen primarily due to them being recognized as the pioneers of physics-driven interactions in video games. They have developed a distinct ‘feel’ for their physics interactions that significantly amplify the immersion within their horror experiences. This ‘feel’ and the physics interactions themselves are at the core of this thesis's objective, which is to replicate them within Godot 4.

3.1 Overview of FrictionalGames' Design Philosophy and System

This section transitions the discussion away from the specifics of the interaction types, and what they are, towards an exploration of FrictionalGames’ distinct application of interaction design concepts. Their use of direct, physics-driven interactions serves deliberate goals of enhancing immersion, fostering agency, evoking specific emotions, and maintaining environmental focus (Grip 2013; Grip 2016). These deliberate choices do not stem from an accidental technical feature, but from a thought-out design philosophy that is deeply integrated into each game they develop. Understanding this philosophical foundation is crucial in the upcoming discussions.

The central goal of FrictionalGames’ interaction model is to deepen player immersion (Grip 2013). They achieve it by creating intuitive controls, which allow the player to smoothly play without actively thinking about them (Shneiderman 1997; Grip 2017b; Norman 2013). These intuitive controls enable players to manipulate the objects within a virtual environment, through physics-driven controls (Millington 2007; Grip 2008; Grip 2010a). This methodology resonates heavily with Mel Slater’s Place Illusion (PI) and Plausibility Illusion (Psi) framework (Slater 2009). The believable physics interactions reinforce both elements: direct manipulation strengthens the feeling of presence (PI), while the physical simulation enhances the authenticity of events (Psi) (Slater 2009).

Mechanically, the intuitive controls are naturally mapped based on the player’s pre-existing mental models (Grip 2008; Grip 2010a; Grip 2017b; Norman 2013). These are primarily mouse movements, which are converted into raw physical forces, or torques, that are then applied to the interacted objects (Grip 2008; Grip 2010a). The naturally mapped controls enhance the player’s ‘kinesthetic involvement’ with the game, which strengthens the player’s sense of agency and sense of presence within the virtual environment (Calleja 2011; Cairns et al 2014; Grip 2017a; Grip 2017b).

This emphasis on kinesthetic involvement and agency weaves directly into FrictionalGames' wider design philosophy (Grip 2010a; Grip 2017b). Instead of tutorials, the interaction system encourages agency and discovery through physics manipulation, thereby avoiding excessive hand holding (Grip 2008; Grip 2013; Horreur 2023). This way the player connects to the virtual environment through exploration (Grip 2017a). Additionally, the interactions that the player engages in throughout their gameplay, creates a feedback loop between the protagonist and the player. (Grip 2017a). This can lead to the player making clumsy mistakes under stress, as they foster the similar vulnerable mindset as the protagonist they are controlling (Grip 2011; Brown et al. 2004). For instance, when escaping a monster and end up struggling with opening/closing a door. In this way, the interaction system is crucial to establishing the game's overall atmosphere and delivering the desired emotional impact, alongside its primary role in player immersion and agency (Grip 2008; Grip 2013; Grip 2016; Grip 2017a).

However, for the atmospheric and emotional effects to be apparent, there is a heavy reliance on the precise ‘feel’ of the interaction, which is a factor that FrictionalGames values greatly (Swink 2008; Grip 2010a). The interaction must seem tangible and weighty, behave believably, and be appropriately responsive within the game’s context (Swink 2008; Slater 2009; Grip 2010a). FrictionalGames’ developer blogs highlight the extensive ‘polish’ and tuning required to attain the appropriate ‘feel’ for an interaction (Grip 2010a). This commitment, coupled with the nature of the interactions, and the use of minimal UI elements ensures that the player stays focused on the game world, and its believable interactions (Grip 2008; Grip 2010a; Iacovides 2015; Norrman 2020).

3.2 Key Interaction Types

This section will cover the key interaction types present in FrictionalGames’ interaction model. These types, such as directly grabbing objects and applying forces based on mouse gestures, form the foundation of their interaction model (Grip 2008; Grip

2009a; Grip 2010a). Additionally, these interaction types are central to the implementation, and the following analysis discussed later within this thesis.

The first key interaction type to be discussed is the direct object manipulation, also referred to as ‘grabbing’. This interaction type allows the player to virtually pick up, hold, throw, and rotate objects (Grip 2008; Grip 2010a). It functions by translating the player’s mouse movements into corresponding forces or torques on the grabbed object (Grip 2010a). This method aims to simulate an intuitive physical grasp, which enables the player to have a tangible connection with the virtual environment (Slater 2009; Davies 2009; Grip 2017b).

The second key interaction type is the manipulation of objects constrained by physics joints (Grip 2009a). For example, objects constrained by a hinge joint (doors, lids, levers), as well as slider joints (sliding doors, drawers). For both of these, the mouse input is presumably interpreted either as torque for hinge joints, or forces for slider joints. Achieving an intuitive ‘feel’ for these interactions requires extensive and careful tuning (Swink 2008; Leadwerks 2013; Zook 2019), a challenge noted by FrictionalGames themselves (Grip 2010a).

The third interaction type involves objects that rotate around a central axis, such as wheels, or more specifically valves. Compared to simple hinged objects, these often require continuous input from the player to function. The wheel’s system looks for mouse circling gestures, which are then converted into torques that are applied to the object’s rotational axis (Grip 2010a; FrictionalGames 2020). The assumed of goal being to simulate the physical act of continuous rotation potentially with resistance or the effect of the action. Achieving their distinct ‘feel’ for this interaction type required refinement efforts from FrictionalGames, with the focus being on making the interaction intuitive and attempting to fix challenges players faced in earlier versions (Grip 2010a).

The fourth and final interaction type revolves around the manipulation of heavy objects. These are objects that are too heavy to be directly picked up via the first interaction type (Grip 2010b). For example, large metal crates or barrels. This interaction type, referred to as the ‘push’, involves the player applying forces that overcome the object’s inertia and friction, which allows it to be slid along the ground (Grip 2008; Grip 2010b). The goal of this type is to convey the object’s significant weight and the resistance that comes with it (Grip 2010b). Although being a key interaction type in FrictionalGames’ model, it won’t be discussed within this thesis as it falls out of the replication scope, and due to external time constraints which hindered implementation.

To summarize this section, the scope of this replication involves the interaction type for grabbing objects, interacting with hinged objects like doors, and interacting with objects featuring rotational mechanics such as valves. The implementation of these three key interaction types, with a focus on capturing their nuanced interaction ‘feel’ (Swink 2008; Grip 2017b), is key in the comparative analysis against FrictionalGames’ model.

3.3 Technical Considerations and Challenges

This section covers the technical requirements and difficulties involved in the implementation, and refinement, of the FrictionalGames’ physics-driven interaction model. Accurately achieving their distinct ‘feel’ for nuanced interactions requires more than just enabling basic physics simulations; it involves overcoming specific technical hurdles (Grip 2008).

To overcome these hurdles requires a capable enough physics engine to build it on. FrictionalGames utilizes the integrated physics library within their proprietary game engine, the HPL engine (IndieDB 2010; Amnesia Wiki 2025). It is called the Newton Game Dynamics and it provides the core physics capabilities in their titles, such as collision detection and rigid bodies (Grip 2008; Grip 2010a). This physics engine is critical for FrictionalGames to achieve their distinct ‘feel’ for interactions. It can be presumed that a big chunk of their development efforts includes the time-consuming, but crucial process of fine-tuning physics parameters. This can be noted by their blog, where they mention spending roughly a week tuning the door interaction (Grip 2010a). This likely involved adjusting parameters such as mass, damping, and friction for each interactable object until they felt believable and intuitive. Then the same was done with joint constraints and PID Controller gains (FrictionalGames 2020; Åström 1995).

In addition to the extensive tuning, developers faced challenges with mapping 2D mouse input into intuitive 3D forces. This can be noted by the improvements made to the interaction system for Amnesia: The Dark Descent, where they fixed the issues players faced in Penumbra’s interaction system (Grip 2010a; Leadwerks 2013). Based on their blog, the issues seemed to stem from custom, unintuitive algorithms which applied forces/torques differently (Grip 2010a). In addition to improving the custom algorithms, they enhanced the responsiveness of the interactions by minimizing input lag as much as possible (Grip 2010a).

During their earlier implementations of the interaction system, they noted particular issues with the real-time physics simulations. Primarily in the form of occasional unstable behavior, such as the ‘chaos effect’ (Grip 2009b; Grip 2010a). The Chaos effect refers to small variations in systems (e.g., parameters values or different algortihms) creating massive, unpredictable results. FrictionalGames noted a puzzle within Penumbra was plagued by this issue; even after setting up additional corrective forces and spending ‘numerous hours’ on getting it stable, the issue would still persist (Grip 2009b).

Despite FrictionalGames overcoming these technical hurdles through iteration (Leadwerks 2013; Grip 2010a), details about the implementation techniques involved in creating their interaction system remain mostly undocumented. Their developers often discuss high-level problems, but generally do not publish the specific low-level implementation details (Grip 2008; Grip 2010a). This includes information on their ‘input-to-force’ algorithms , tuned parameter values , or other tuning strategies they utilized. Although FrictionalGames has released the source codes for their game Amnesia: The Dark Descent, it is presented in a high-level programming language known as C++ (FrictionalGames 2020). C++ is a language that notoriously requires significant programming expertise to navigate effectively. Thus, it is often difficult for designers or novice developers, who are interested in implementing a similar system in their own games, to decipher. Replicating a similar interaction system therefore requires independent analysis and practical prototyping. This is the same approach adopted in the subsequent chapters of this thesis.

3.4 Role as Benchmark

The previous sections provided a detailed look at FrictionalGames’ physics-driven interaction model. Section 3.1 discussed their design philosophy, which focused on immersion, agency, and vulnerability (Grip 2013; Grip 2017a; Grip 2017b). Their system’s key interaction types were covered in detail within Section 3.2, with Section 3.3 covering the technical considerations. Particularly focusing on certain challenges and the extensive tuning of parameters (Grip 2010a; Grip 2009b). This chapter’s analysis of FrictionalGames’ interaction system established the understanding of its intended functionality, its design goals, as well as an insight into its distinct ‘feel’.

With the functionality and considerations understood, this model will effectively act as the benchmark for the Godot engine implementation presented later in this thesis. The analysis conducted within this chapter provides the necessary criteria for evaluation in Chapter 6. Success of the implementation will be measured based on a detailed checklist of expected mechanical behaviors and specific anomalous behaviors, or their absence. This checklist will be crafted from a careful observation of the benchmark system, which will then be utilized to rigourously test the implementation and evaluate its accuracy to the benchmark. The checklist will cover the core interaction mechanics discussed within Section 2.3; particularly picking up objects, manipulating objects constrained by a physics joint (doors, sliding doors), and objects featuring rotational mechanics (wheels/valves). With additional checks to measure the characteristic ‘feel’ from FrictionalGames’ model based on its intuitiveness, responsiveness, and stability.

With the benchmark system and the evaluation criteria established, the discussion will now turn to the practical work conducted in this thesis. However, before discussing the core implementation, the insight gained from previous replication attempts for a similar interaction system will be explored. This will provide additional context for the methodology that will appear in Chapter 5.

4 Previous Attempts

This chapter serves to discuss the author's previous attempts at replicating a Frictional-style interaction system within Godot. It will detail the techniques used, their flaws, and the challenges that were encountered. The discussion will provide context to the choices made and the methodologies used for the thesis' practical work.

The information for this discussion will primarily stem from the author's blog (Aun 2023), which outlines the development process of the previous prototype, and the technical hurdles it experienced.

Additionally, it will provide clarity for the author's learning process and iterative development approach.

The chapter is structured as follows: section 4.1 will provide an overview of the initial prototype's implementation and key mechanics. Section 4.2 discusses the difficult challenges faced during development and playtesting, alongside their potential solutions. Section 4.3 outlines the key takeaways and learnings gained from the entire experience. Finally, Section 4.4 concludes the chapter by connecting the previous attempt to the latest, refined implementation presented in Chapter 5.

4.1 Overview of the Pre-thesis Prototype

The primary objective of the initial implementation was to replicate certain mechanics from FrictionalGames' interaction model, as well as gain additional development experience within Godot.

The process involved insvetigating the key components (interactions with doors, wheels, levers, etc.) involved in the interaction system in an attempt to understand their functionality. Then, through that understanding a prototype would be replicated with physics components within Godot. The aim of the prototype was to foster the sense of kinesthetic involvement and player agency familiar of the example system.

To begin with, a simple raycast system was built to detect interactable objects, similar to the one utilized within this thesis' implementation. A ray would be cast from the player's camera which would have a length comparable to a reaching-out hand. If the 'ray-intersected' object was detected as interactive, it could then be manipulated. For grabbing and holding objects, a physics joint was used. The physics joint was attached to the currently manipulated object, specifically at the point where the raycast hit. This allowed the PID controller to handle the movement of the joint's object towards the ‘hand’ object’s position. This method was used to portray perceivable weight on the objects, as they would fall behind the ‘hand’ position when being held. A key aspect to achieving the ‘feel’ (Swink 2008).

For interactions involving doors, wheels, and levers, a much inferior approach was used. Essentially, each interaction involved the same logic as grabbing. In addition to the camera also not being locked, the resulting interactions were unintuitive and inconsistent; meaning that it went against the principles of kinesthetic involvement, and natural mapping (Shneiderman 1997; Swink 2008).

4.2 Challenges Encountered and Solutions Explored

One of the initial hurdles to be encountered was the grabbing system's functionality. The tutorial-based system worked flawlessly, however it did not provide the desired, ‘anchor point’ result. The first solution involved applying the PID force at the specific grab point, which would be calculated relative to the object's center. This method was abandoned due to physics instabilities, such as the objects appearing to "glitch and flick around”. This significantly undermined the plausibility of the simulation (Slater 2009). This unstable behavior was reminiscent of the ‘chaos effect’ discussed by Thomas Grip (2009b), which FrictionalGames also attempted to mitigate.

This challenge necessitated an investigation into FrictionalGames' methodology, which lead to the discovery of PID controllers. These were coupled with physics joints to achieve the desired effect. The PID controller allow for a smooth and precise control over the manipulation of the grabbed object, while the physics joint provided the ‘anchor point’ result.

The implementation of door (hinged) interactions also involved a tutorial-based approach to develop a ‘precision drag’ system. Simply, it was a system that combined mouse delta with the player’s rotation to manipulate the door. The player's rotation allowed interactions with the door to stay consistent and intuitive from any angle. However, the system was modified to function in a similar manner to grabbing. Essentially, the mouse's relative motion was translated into global space, which was then used to manipulate the door’s angle. The angle was calculated based on mouse movements, and the intersection point of the raycast which was relative to the door’s hinge position. The rotation was made smoother through the use of linear interpolation methods. The end result demonstrated effective natural mapping, however the interaction itself was unintuitive. Its functionality is illustrated in Figure 3.

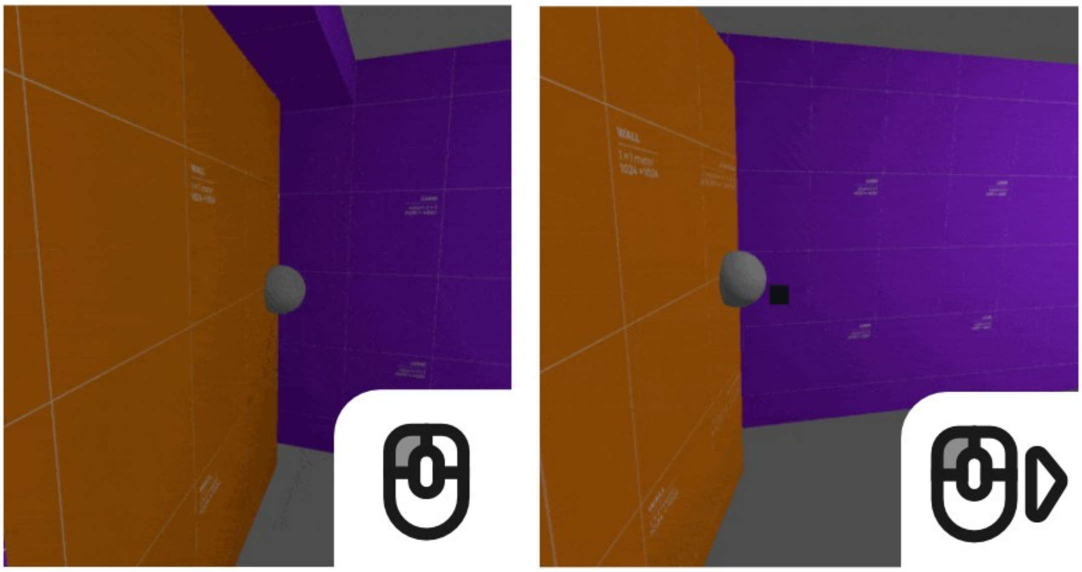


Figure 3. Door interaction in previous prototype.

The implementation of wheels involved multiple approaches, many which were unsuccessful. These approaches include attempts with PID controllers, joints and the reuse of existing scripts such as grabbing and doors.

The approach which was deemed successful, also happened to be the initially attempted approach. This involved translating the wheel script from Amnesia's source code to use within Godot (FrictionalGames 2020). Essentially, the mouse's 2D viewport positions are converted into 3D coordinates via raycast. If the raycast hits the wheel, an angle is calculated from the center of the wheel to the raycast's intersection point. This calculated angle is then used to rotate the wheel. The reason this method was deemed unsuccessful in the beginning was presumably, due to flaws during the transfer of logic. This resulted in the wheel not rotating. After some adjustments, the logic worked and the wheel was able to be rotated. However, the result still was not accurate due to improper setup of physics joints and existing, unknown flaws within the logic. This caused the wheel to tilt during interaction, as well as not follow the player's mouse position accurately. Subsequently, the logic of the wheel script was used in implementing the lever, which required some additional rotational constraints and a change in its rotational axis.

Additional issues appeared for these mechanics during playtesting, such as a 'jumping' bug, which occurred when grabbing would be re-initiated after holding a heavy object. The cause was determined to be the hand-object, which would stay at the position where the previous object was held. This ‘drifting’ resulted in the hand-object being misplaced from its intended location, which caused the PID controller to compensate for the drift, and apply a massive, rapid force to the newly grabbed object; resulting it being flown around. Attempts at resetting its position were unsuccessful, and thus it was left untreated.

Issues regarding the instability of physics were a common occurrence. On some occasions the issues were so massive that the feature had to be removed entierly. For instance, door interactions were left out of the playtest. Their issue was suspected to stem from their implemented logic, which resulted them to ‘fly around’. As this issue was encountered late into development, it was left untreated.

During the implementation of a puzzle for the playtest level, a flaw in the lever logic was encountered. When the lever would be pulled an entire 90-degrees, it would ‘break’ and couldn’t be manipulated again. To compensate, the puzzle in question was abandoned, and a new puzzle designed in its place. This new puzzle featured a 45-degree constraint on the levers to prevent the bug entirely.

The previously mentioned puzzle originally utilized wheels, instead of levers. However, this change stemmed from a suspected bug in Godot. Essentially, during interaction with the wheel it would occasionally make sudden rotations, which made it rotate past its defined constraints. This was deemed an issue with Godot due to constraints being applied within the wheel script, as well as the hinge joint simultaneously.

Due to the severe bugs being encountered, the final playtest build of the prototype featured the option to skip levels. This allowed playtesters to still experience all of the implemented mechanics without getting stuck due to an untreated bug.

4.3 Key Learnings and Takeaways

The use of PID controllers and physics joints, provided a viable method to implementing a basic ‘anchor point’ grabbing system, similar to that of FrictionalGames' titles. Allowing the grabbing mechanic to create a sense of weight on the virtual objects. While the setup is sufficient for a simple grabbing mechanic, the PID controllers and physics joints presented numerous challenges, such as the ‘jumping’ bug discussed prior. The challenges encountered during the playtesting of this grabbing mechanic, highlighted the need for careful state management, as well as a better understanding for the physics engine’s behavior when combining both of these elements. As applying raw PID forces without any precise control on the joint's anchor led to instability issues.

While directly calculating 3D rotations from 2D mouse positions for wheel interactions achieved a form of natural mapping, it did not utilize forces from the physics engine to drive the rotations. Instead, the rotations were set directly to the desired rotational axis. This highlighted the trade-off for these rotational interactions as setting the rotations directly was a simpler technique, and avoided any issues with physics. However, this meant that the interactions would lack the tangible 'feel' and any emergent behaviors, which are central to FrictionalGames' model (Swink 2008).

All of the encountered issues made the knowledge gap even more apparent. As developing such a physics-driven interaction system in Godot 4 required extensive iteration, research and a willingness to try and discard multiple approaches. All due to the lack of tutorials, and related material. This led to perseverance being the key in the implementation. As after numerous iterative steps, solutions would often emerge when revisiting an earlier problematic idea with new insights. This proved that replicating existing systems is not a linear process, and demands significant experimentation. Especially, when dealing with undocumented features or complex physics interactions.

While, Godot 4 proved itself to be a capable development environment for such an interaction system with its tools, achieving stable and predictable physics proved to be a challenge. Primarily, due to Godot being a game engine, which recently garnered a community. Leaving the discussion on such niche interaction systems lacking. This meant that the encountered issues and perceived quirks in the physics simulations during implementation, and playtesting were difficult to diagnose. Thus, leaving many of them unresolved. This coupled with the lack of documentation, significantly impacted development speed and reliability. This meant that while Godot is capable, a deeper understanding of its tools and backends, in addition to community support, might be needed to overcome such challenges.

Even with functional mechanics, achieving the distinct 'feel' of FrictionalGames' interactions is a significant additional challenge beyond just basic implementation (Swink 2008; Grip 2010a); it being highlighted by this initial prototyping phase. Proving that to achieve it the implementation requires more refined techniques, improved tuning of physics and PID controllers, and better systems for debugging. As the encountered bugs directly impacted the perceived quality and believability of the interactions. Therefore, providing the reasoning for why further development would be necessary.

4.4 How Previous Attempts Shaped the Current Work

The challenges faced and the lessons learned during this early implementation directly shaped the techniques and the technical approaches that were adopted in the subsequent practical work. It highlighted the crucial areas of improvement for future implementations. In addition to bringing awareness to the weaknesses of Godot, it also provided a better understanding of the available tools, and their functionality. For example, joints and various physics bodies. It was a crucial exploratory step that provided useful insight into the needs of a stable, and believable Frictional-style interaction system.

The issues that emerged with the grabbing mechanic like the jumping bug or general instability, provided the direct focus areas for this thesis' implementation. They motivated the careful exploration into stabilizing and simplifying the overall functionality of the system.

The unsuccessful and convoluted raycast-driven approach being omitted in favor of a more physics-driven approach, which would utilize appropriately calculated forces/torques. This would provide far favorable results due to closely resembling FrictionalGames' approach to their model.

Overall, this initial exploratory step proved that to implement a physics-driven interaction system within Godot, regardless if it is based on FrictionalGames' model, requires extensive iteration, consideration of the challenges that might be faced, as well as the significant time that is required to capture the appropriate 'feel' (Swink 2008; Zook 2019). The insights gathered through this implementation lead to a more conscious effort to develop with an aim to enhance overall quality and plausibility of the player's experience (Slater 2009).

While the early prototype was ridden with numerous experience-breaking bugs, it provided invaluable knowledge for the development of such a complex interaction system. This knowledge will be seen through the methodology used and discussed within the subsequent chapter, which discusses the practical work of this thesis.

5 Thesis Work Implementation

This chapter details the implementation of the key interaction types of the FrictionalGames' interaction model within Godot 4; with a consideration for the techniques, tools, and key learnings from the previous attempt.

Mainly, it will cover the following interaction types: picking up items, manipulating hinged objects, turning wheels, and pulling levers.

The aim was to accurately recreate the nuanced *'feel'* of the example system, as well as replicate its naturally mapped interactions.

5.1 Implementing Object Pick Up

The object pickup mechanic is arguably the most used mechanic in the interaction system during gameplay, as it allows the player to directly interact with their virtual surroundings. Players are able to grasp, move, and throw virtual objects via simulated physics. Indicative of the 'kinesthetic involvement' discussed prior in the Theoretical Framework (Calleja 2011; Cairns et al. 2014).

The following paragraphs will detail the implementation of the pickup mechanic, its prerequisites, and additional components. Figure 4 shows the overall node structure within Godot that supports the entire interaction system.

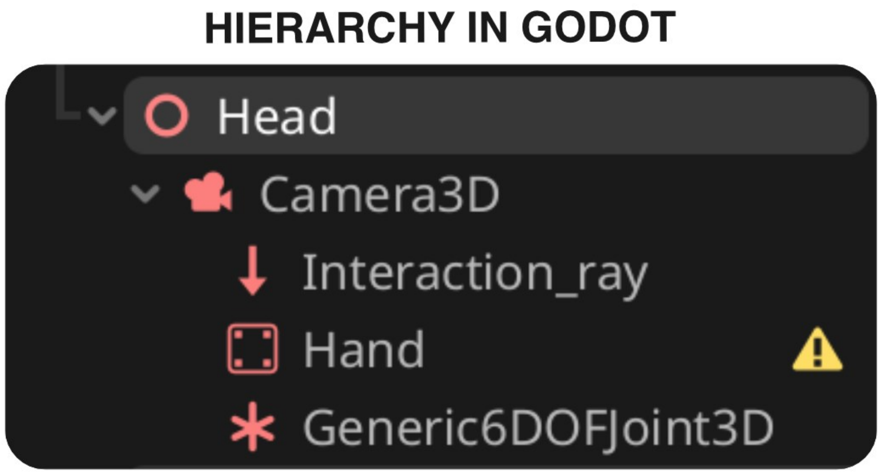
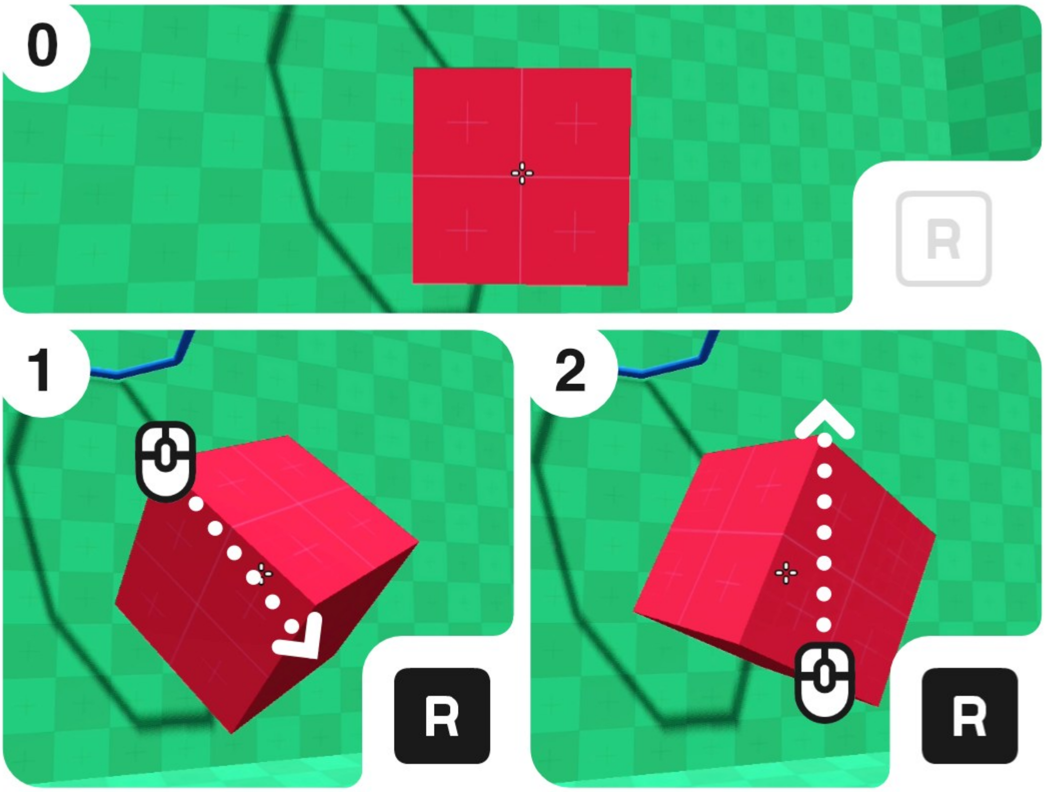


Figure 4. Grabbing mechanic hierarchy setup in Godot 4.

To begin with, the pickup mechanic is initiated by the player ‘grasping’ a ‘grabbable’ object by pressing and holding down their left mouse button. The object stays held until the mouse button is released. The object is then pushed by physics forces towards a specified target point, which is slightly in front of the player’s camera. While the object is being held, its rotation is constantly adjusted to face the player. The held object can be rotated manually by holding down the “R”-key on the keyboard, and moving the mouse in the desired direction. While this feature is active, the player’s camera is locked.

Within the rotation logic, the mouse’s relative movement, also known as mouse delta, is converted into radians. The resulting value is used within Godot’s built-in methods *rotate_y()* and *rotate_x()*. These methods are used to rotate objects around their local X and Y axes respectively. The result is a smooth, intuitive rotation that matches the feel of the entire mechanic. This mechanic is illustrated in Figure 5. The setup for this feature consists of a *Generic6DOFJoint* node, and the target point as a *StaticBody3D*. The *Generic6DOFJoint* has every aspect of it constrained besides rotations to prevent any unwanted manipulations.



NOTE: DEPICTED MOUSE MOVEMENTS REPRESENT PHYSICAL ACTIONS

Figure 5. Functionality of grabbed object's 'rotation' feature.

The identification of the desired object is handled within the interaction manager, which is also the player script. The method for object detection is a basic raycast system. A short ray is cast forward from the player's camera when the left mouse button is pressed. The object the raycast hits has its type and designated group verified. Each interactable object (door, wheel, lever, etc.) was assigned a group within Godot with the sole purpose of identifying interactable objects. The method of verifying both, the rigid body type and group, was to prevent interactions with other, non-interactable objects within the virtual environment. Once both conditions are met, the desired object's method would be called. The method specific for pickup, moves the held object towards the target point via physics forces adjusted by a PID Controller. The object would maintain its position until the interaction key was released, or alternatively, when the object is thrown. The throwing feature is initiated by pressing the right mouse button. This applies a force impulse in the player camera's facing direction to the held object. Once thrown, the object is disconnected from the target point and no longer held.

A feature yet to be discussed within the pickup mechanic is “inspecting”. “Inspecting” refers to moving the held object closer or further away. This is achieved by moving the target point instead of the held object, as the held object constantly maintains its position at the center of the target point. It is clamped to a specific range to prevent excessive manipulation, and can be controlled via the scroll wheel on the mouse. This is visualized in Figure 6, where the object is shown at both minimum and maximum distances.

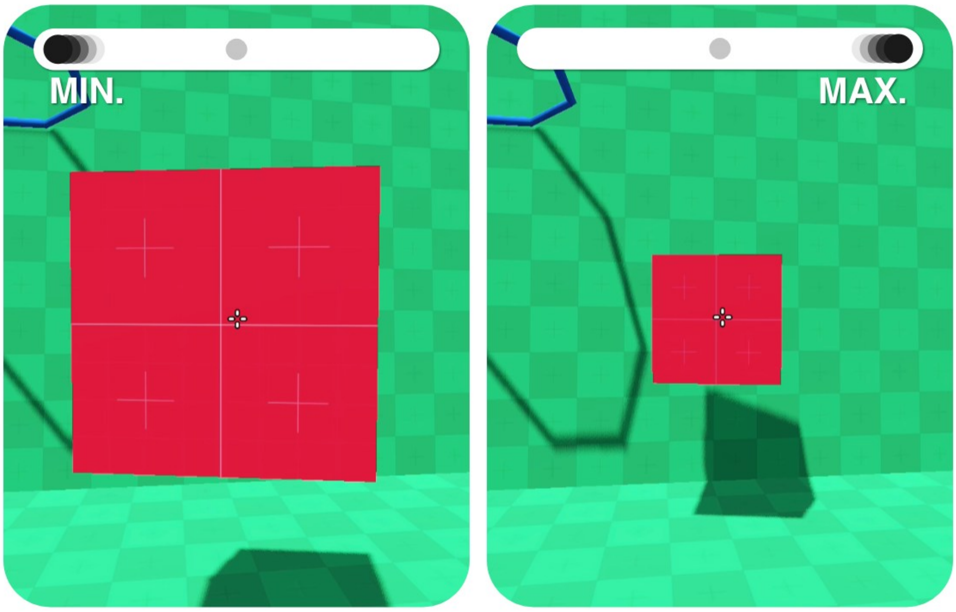


Figure 6. Functionality of grabbed object’s ‘inspect feature.

To achieve the nuanced ‘feel’ of the example system, PID controllers were utilized. These controllers are utilized within the example system as well (FrictionalGames 2020). The gains of the PID controller were tuned through rigorous trial-and-error until they felt appropriate and intuitive.

Although, Frictional has released the source code for their hit game “Amnesia: The Dark Descent” (FrictionalGames 2020), its tuning was not applicable to this implementation. This is due to difficulties, which stemmed from the differences in approaches, development environment, and physics backends.

5.2 Implementing Doors (Hinged Objects)

Hinged objects such as doors are arguably the next most used interaction type within FrictionalGames' model. Its replication in Godot features the following setup: the door object is utilizing a RigidBody3D node with its shape and collision being provided by the MeshInstance3D and CollisionShape3D nodes respectively. This rigid body is then attached to a hinge joint, which is also attached to a door frame. This door frame utilizes the StaticBody3D node, as it shouldn't move during the interactions nor be affected by any physics during gameplay. These two bodies are necessary, as each joint type within Godot requires two rigid bodies to be attached to it. As discussed within the theoretical framework, this is how physics joints fundamentally work; the primary rigid body is constrained relative to the secondary rigid body. The door frame in this setup, is the secondary rigid body. The HingeJoint3D constrains the door object to a singular, defined axis. Then, the hinge joint's parameters are adjusted to constrain the door to a desired range of motion (e.g., opens only 90 degrees to either side). The setup and node hierarchy for this mechanic are shown in Figure 7.

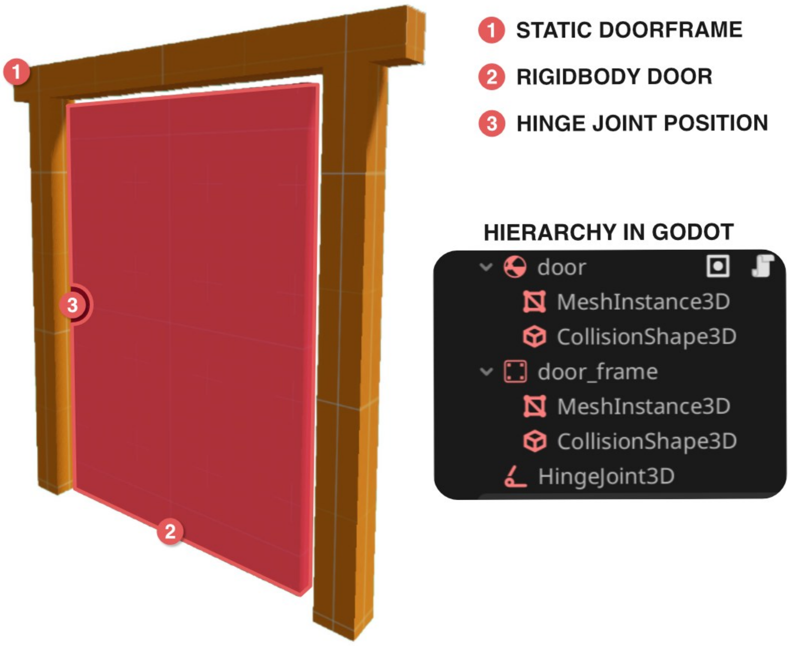


Figure 7. Door mechanic hierarchy setup in Godot 4.

The initial implementation of the door's logic involved capturing the 2D mouse movements (delta X and delta Y), while the interaction button was held. These raw input values were directly translated into 3D forces, which were applied to the door's center. This method of force application allowed the door to be opened in a realistic manner, as the player would need to do a curve with their mouse to open it; reminiscent of opening a real-world door. This result heavily aligns with the crucial concept of natural mapping discussed in the theoretical framework.

However, based on initial observations of the benchmark system, its functionality did not seem accurate. Thus, the logic was slightly reworked to resemble the example system's implementation. The calculations for translating mouse movements into 3D forces were kept, and forces were changed to torques. This was done due to issues encountered with consistency (e.g., forces not being applied uniformally), which subsequently impacted intuitiveness with the more realistic approach.

While the changed logic fixed the consistency issues, it introduced new issues with intuitiveness. Primarily, regarding the player's position. When the player interacts with the door at an angle, the torque has to be flipped. The logic responsible for this happened to be faulty. Thus, rotations were not flipped appropriately, which impacted the effectiveness of natural mapping. This issue is illustrated in Figure 8.

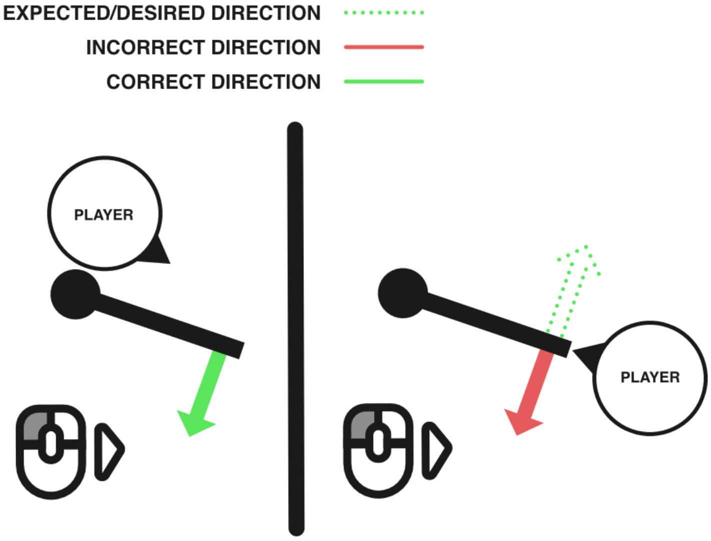


Figure 8. Door torque flipping issue.

Although, interactions were still mostly functional and resembled the implementation by FrictionalGames.

A PID controller was used to smooth and stabilize the result, while also enabling precise control over the interaction so the appropriate ‘feel’ could be achieved. Additional parameters were adjusted to improve the ‘feel’, such as mass and damping. Mass was used to provide a general sense of weight to the interaction, while damping was used to prevent the door from swinging during interactions. The swinging stemmed from the momentum that was generated from the applied torque.

As already discussed within FrictionalGames’s own blog and within the theoretical framework, this tuning process was critical to creating the ‘feel’ for the interaction. While involving rigorous time-consuming trial-and-error, the end result could still be improved through further tuning.

5.3 Implementing Wheels (Rotational Objects)

The next mechanic to be covered is wheels, or objects with rotational mechanics. These objects require the player to continuously perform circular mouse gestures to make them rotate.

Its setup within Godot is similar to other interaction types discussed prior; the RigidBody3D node is used for physics, while the CollisionShape3D and the MeshInstance3D nodes provide the shape and collisions. The HingeJoint3D is utilized again as it is the most appropriate. This is to prevent the tilting issue encountered in the previous implementation when using a 6DOFJoint3D. A VehicleWheel3D node was also considered as a substitute to the joints, however it doesn't provide the necessary constraints to capture the wheel’s behavior accurately. The setup and node hierarchy for this mechanic are shown in Figure 9.

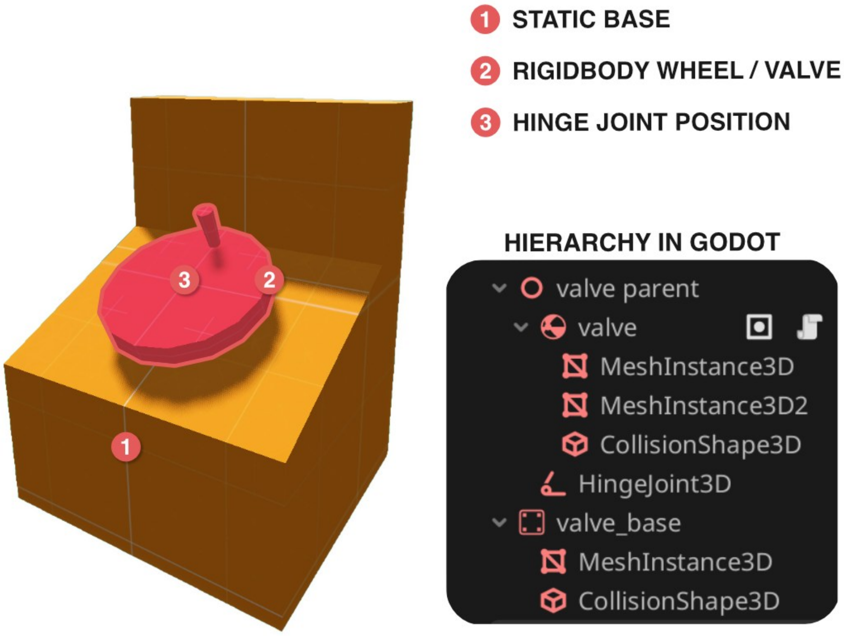


Figure 9. Wheel/Valve mechanic hierarchy setup in Godot 4.

The logic behind this mechanic involves the technique from the previous implementation. Essentially, the 2D mouse position relative to the center of the screen is used in determining the directional vector. This is then combined with the mouse's motion (mouse delta) to calculate both the direction and magnitude of the mouse gesture. The mechanic’s interaction is illustrated in Figure 10. The use of a PID controller was omitted from this implementation as it caused major oscillation issues, which were not able to be fixed through tuning. Hence, why applying the torque directly provided a more intuitive and smoother interaction.

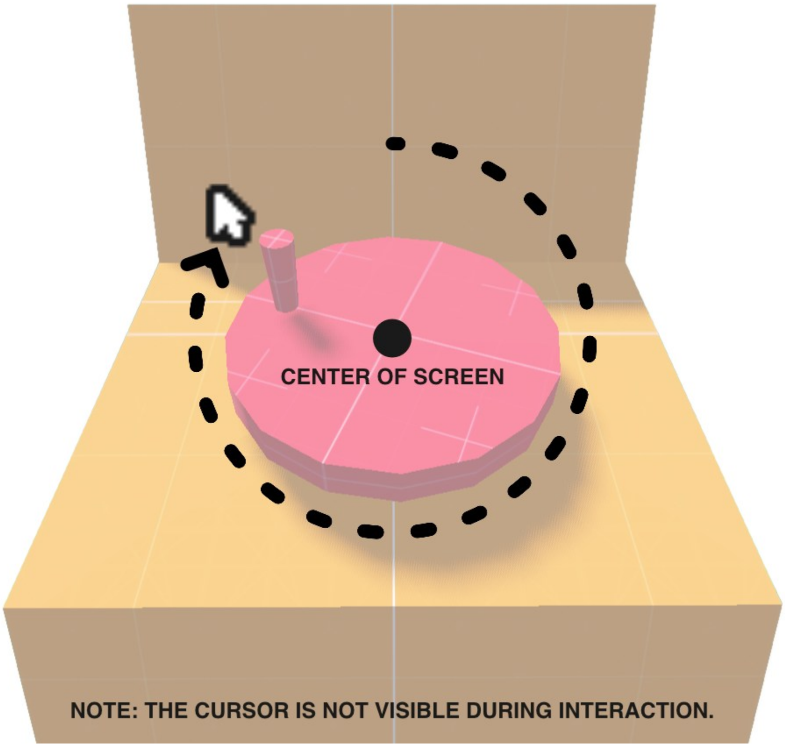


Figure 10. Wheel/Valve interaction.

Although missing the precise control offered by a PID controller, this method still directly improved upon the initial unsuccessful implementation presented in the previous attempt. This is due to the physics engine driving the rotations for the wheel through the application of torque. Additionally, the usage of the hinge joint prevented any tilting or additional transformations entirely.

To compensate for the lack of a PID controller, angular damping was used to achieve the desired 'feel'.

5.4 Implementing Levers

The last interaction type to be covered is the lever. They offer an interesting way to craft puzzles by fostering a tangible connection through manipulation, and feature more flexibility than basic buttons. This new implementation for the lever ditched the technique used in the previous attempt. Primarily, the use of the wheel logic in favor of proprietary logic. This was due to the new system not being built around the raycast system like the previous. The setup and node hierarchy for this mechanic are shown in Figure 11.

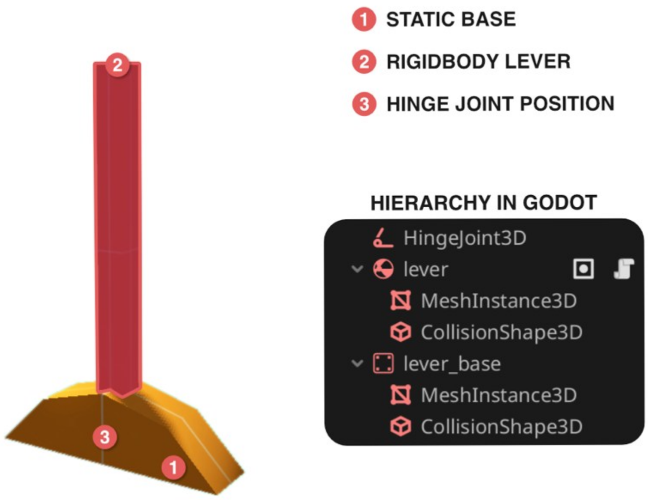


Figure 11. Lever mechanic hierarchy setup in Godot 4.

The initial challenge faced with the lever’s implementation, emerged due to gravity. Gravity being applied by the RigidBody3D node meant that during the interaction the lever would fall over to one side. The initial solution was to ditch the RigidBody3D node in favor of an AnimatableBody3D. Mechanically, this physics body is the same as the static body, however it can be moved through code. Similar to a kinematic body. Although, this method proved successful, the interaction was not driven by physics. This necessitated further observation into the example system’s implementation, which revealed the solution. To prevent the lever from falling over while still being affected by physics forces, is to simply disable gravity entirely during the interaction. This allowed the RigidBody3D node to be used, and subsequently also physics forces to affect the lever. Additionally to prevent the lever from falling over outside of interactions, was to apply constant forces to the lever that would keep it standing upright. This also makes sense mechanically to the player as they likely expect a lever to behave in this manner. Thus, strengthening plausibility (Slater 2009).

A PID controller was used to smooth both the interaction torque, as well as the return-to-center torque. This resulted in slightly springy movement when it was returning to the center.

Overall, the switch to proprietary logic proved as a significant improvement over the method used within the initial prototype. The implementation is simpler, and more stable than compared to the latter. Although, no function testing like the puzzle seen in the initial prototype was conducted, it can be confidently determined that due to the improved, cleaner codebase the implementation of new features will not be troublesome.

6 Comparative Analysis

This chapter serves to discuss the comparative analysis, which involved the implementation detailed prior and FrictionalGames' interaction model. The goal of this analysis was to measure the mechanical accuracy of the implementation done within Godot, which was motivated by the challenges encountered in the pre-thesis attempt.

The key interaction types that were analyzed are the following: object pick-up, interactions with hinged objects, manipulating objects with rotational mechanics, and pulling levers. The implementation of these mechanics was detailed in the previous chapter, and they were chosen due to being the fundamental interaction types that appear in multiple titles by FrictionalGames.

The methodology for the comparative analysis involved building a test scene, featuring all necessary interaction objects using the level editor that comes with every copy of Amnesia: The Dark Descent. It can be seen in Figure 13, while the test scene within Godot can be seen in Figure 12. This test scene was critical to the analysis as it was a time-efficient method allowing every interaction to be targeted directly. The alternative would have been to play the game's campaign, and potentially spending hours looking for the desired interaction object. Amnesia: The Dark Descent was chosen as the test title due it being critically acclaimed, as well as being the only title available to the author at the time of the analysis.

The benchmark was conducted based on a detailed checklist, which outlined the expected mechanical behaviors of all key interaction types. Additionally, it outlined anomalous behaviors that might occur if the system is flawed, such as instabilities. These checks were then rigorously tested against the implementation within Godot.

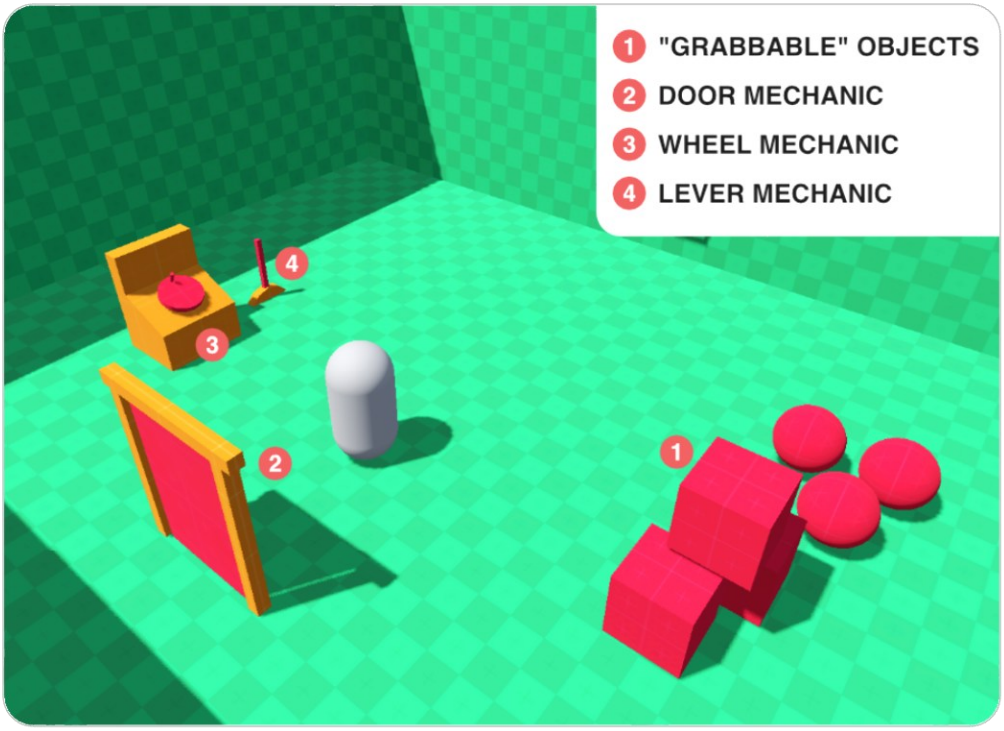


Figure 12. Mechanic testing environment in Godot 4.

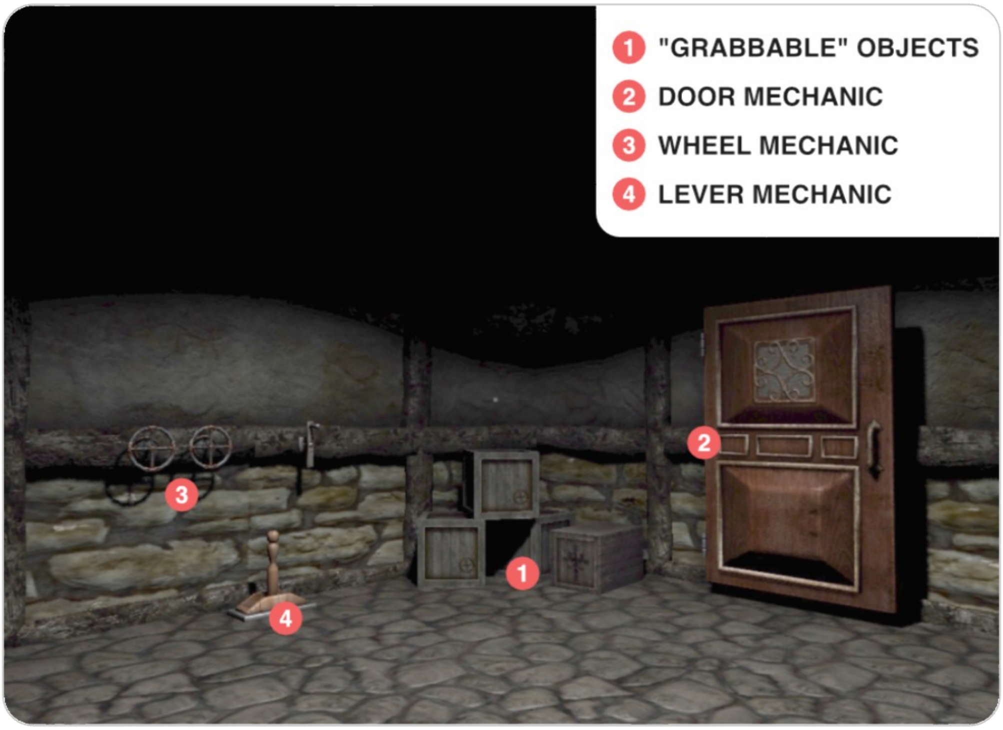


Figure 13. Mechanic testing environment in Amnesia: The Dark Descent.

The focus of the checklist was to assess the mechanical accuracy, as well as the proper implementation of natural mapping in the interactions. Although acknowledge thoroughly in Chapter 2, the subjective aspects of ‘feel’ or broader player immersion were excluded from checklist, due to being notoriously difficult to assess (Kukshinov et al. 2025). The use of the detailed checklist allowed for a direct comparison of mechanical accuracy, and help identify any specific areas of deviation.

The following section will present the findings for each interaction type discussed in Chapter 5. This will be followed by a discussion of overall evaluated mechanical accuracy, and potential limitations. The chapter is concluded by section 6.6, which synthesizes all findings across the analysis.

6.1 Comparative Evaluation of Implemented Mechanics

6.1.1 Object Pick Up

This subsection will focus on comparing the grabbing mechanic implemented within Godot against the benchmark system within Amnesia: The Dark Descent based on the checklist. This comparison will center on mechanical accuracy and the natural mapping of the interaction.

The first check involved observing how the mouse gestures are converted into 3D forces on the object. Beginning with the implementation within Amnesia. A slight delay was observable when moving the held object around; reminiscent of stiffly tuned gains of a PID controller. The implementation within Godot showcased similar behavior. However, it featured a larger delay due to the tuning of the PID gains. This means that the delay can be resolved for greater accuracy with additional tuning. This additional tuning would also resolve the perceivably slow but smooth settling of the object after movement, which was not observable within Amnesia’s implementation. In Amnesia, an object would settle almost instantaneously, with larger mouse movements displaying a slight overshoot of the target position, or a little ‘bounce’.

On the discussion of large mouse movements, the Godot implementation was able to handle them without issues. Especially the rotation feature of the grabbing mechanic, which was observed to be even more responsive than in Amnesia. The feature’s implementation within amnesia showcased a large delay when doing a multitude of large mouse movements, with the system taking a second to ‘catch up’.This large delay was not observable in the Godot implementation, where doing the same large movements displayed surprising responsiveness. This proved to be a major success over the example system.

Moving onto the throwing mechanic, which seemed to be identical to that of in Amnesia. Both systems displayed applying appropriate force impulses onto held objects, which subsequently collided believably with the environment. The only observed difference being the amount of applied force; with the Godot implementation applying a more realistic amount. Furthermore, the implementation within Amnesia applied enough force for the player to throw a wooden box an estimated 10 meters as if it was weightless. However, both throws felt believable, and thus did not impact plausibility (Slater 2009).

The inspection mechanic within Amnesia felt abrupt and lacked smoothness. Additionally, the controls felt unresponsive with functionality requiring multiple scrolls with the mouse wheel. The implementation within Godot addresses this directly, with each scroll from mouse wheel manipulating the object. Although not accurate to the example system, it felt more responsive and intuitive. The distance range which the objects were able to be manipulated on, was observed to be the same.

The only anomalous behavior that could be observed was concerning collisions, particularly involved with pushing objects through walls/floors. Although objects could not be pushed through walls or floors, sustained pushing caused the PID controller to 'break'. This meant that the PID controller would not accurately manipulate the object back to the target position, causing it to be significantly offset. If the object was pushed against a wall, and the player would forcefully walk into the held object, it would eventually clip through the surface. In Amnesia, this issue was resolved by disabling collision between the player and the held object. The reason this was not resolved within the Godot implementation was due to it being overlooked prior to the analysis.

Any additional anomalous behaviors were not observed, with even the 'jumping bug' from the previous implementation being resolved. Objects were now able to be grabbed repeatedly without issues, as well as be grabbed in mid-air and thrown with a mouse flick. These were features that were not possible in the initial prototype, highlighting a significant improvement.

To summarize, this replication was a success, which closely resembled the benchmark system's implementation. Especially, in core functionalities, such as holding, rotating, and throwing; all of which, resolved the issues encountered in the previous attempt. Additionally, the Godot implementation displayed an improvement in responsiveness over the initial prototype and the benchmark system. Areas for improvement consist of minor differences like PID stiffness and the major 'surface-clipping' issue.

To conclude this section, the Godot implementation of the grabbing mechanic was mechanically accurate to the example system according to the checklist.

6.1.2 Hinged Objects (Doors)

This subsection will focus on comparing the hinged object interaction mechanic within Godot against the benchmark system within Amnesia: The Dark Descent based on the checklist. This comparison will center on mechanical accuracy and the natural mapping of the interaction.

The implementation of the mechanic within Amnesia seems to display tuning characteristic of low damping with high force application. This tuning makes the interactions feel realistic and intuitive. This is to be contrasted with the Godot implementation, where interactions feel like applying torque to the hinge rather than physically opening the door. A major area of concern for these interactions is player position, which dictates when forces should be flipped based on where the interaction is happening from relative to the door. This is very noticeable within the Godot implementation, as forces are being incorrectly inverted causing the interactions to be clunky and awkward. This significantly impacts natural mapping, as incorrectly inverted forces do not accurately represent the player's mouse gestures; causing a disconnect with the virtual environment.

Regarding the range of motion and constraint handling of the interactions, Amnesia's implementation prevails again. in amnesia. In Amnesia, the test door was able to appropriately open and close without any tilting or additional axis rotations on its 90-degree range of motion. The door within Godot had a set range of motion of 120-degrees. While also not displaying any issues with tilting, closing or opening the door fully caused some additional rotations. Primarily, being observed as unintentional springiness, which is not present within Amnesia's implementation. Besides that issue, both systems appear to display stable joint contraints.

The mouse input during door interactions in Amnesia were observed to be very sensitive, making smaller mouse movements difficult to control. This is likely due to the low damping that was noted earlier, as well as the way forces are being applied. The implementation within Godot does not feature similar sensitivity due to the amount of torque that is applied as well as damping. The damping differences are inherently obvious when a door is let go after building up momentum. The door in Amnesia displays a slight swing in the direction that it was pushed in, while the door in Godot stops almost completely.

Both implementations did not display any anomalous behaviors. With the analysis focusing on any noticeable shakes or vibrations. The joints constraints on both doors keep them properly constrained.

Overall this implementation was unsuccessful. Although, the basic constraints and stability were replicated, there were major differences between the implementations, such as the perceived physics forces being applied (torque vs. forces), intuitiveness of the input mapping, as well as other subtle behaviors (sensitivity, and ’springiness’). To conclude this section, While the door can be interacted with, the way it differentiates from the example system in key aspects of natural mapping and behavior is a major area of concern. A potential solution would be to explore the previously attempted realistic approach, which was discussed in the hinged object subsection of Chapter 5.

6.1.3 Rotational Objects (Wheels)

This subsection will focus on comparing the rotational object mechanic within Godot against the benchmark system within Amnesia: The Dark Descent based on the checklist. This comparison will center on mechanical accuracy and the natural mapping of the interaction. This mechanic is distinct as it requires a different mouse gesture to function than that of other interaction types. To be exact, it requires circling mouse gestures.

The implementation within Amnesia displayed the same sensitivity that was perceived on the door interactions, with it being overtly easy to rotate. However, it did follow the speed of the mouse movements. The implementation within Godot was almost identical; being less sensitive to mouse movements, but because of this it felt more intuitive and responsive. Furthermore, the mouse circle gestures also changed the speed of the rotation. However, there was a perceived cap that could not be exceeded. This did not make the interaction feel unbelievable, as mouse gestures felt to be translated in a way that one would expect. This would suggest a strong adherence to natural mapping.

Regarding joint constraints, both implementations were observed to allow torque application on a single axis, and did not display any unwanted behavior like the wobbling that was encountered in the previous attempt.

Mouse movements in Amnesia were translated accurately, with smaller movements causing slower rotations and larger movements causing larger rotations. Additionally, the rotation direction was also matched accordingly. The implementation within Godot displayed similar behavior to such mouse gestures, even matching the direction of the circles accurately.

The interaction didn't display any anomalous behaviors and was perceived to be stable. The only thing of note being a little ‘hiccup’ that happens on larger rotations. Otherwise, no additional or unwanted transformations were perceived.

Overall, this implementation was a success, with the mechanic accurately replicating the core behavior, responsiveness, and stability of its counterpart from Amnesia. To conclude this subsection, the implementation within Godot achieved the mechanical accuracy goals presented in the checklist and largely met or exceeded them in terms of stability and direct input mapping. To conclude this subsection, the implementation within Godot largely met or exceeded the mechanical accuracy goals set in the checklist. Specifically in the terms of stability and direct input mapping, with the resulting interactions being appropriately responsive and intuitive.

6.1.4 Levers

This subsection will focus on comparing the lever mechanic within Godot against the benchmark system within Amnesia: The Dark Descent based on the checklist. This comparison will center on mechanical accuracy and the natural mapping of the interaction.

The implementation of the lever within Amnesia seems to utilize torque application, which takes into consideration the player's position / perspective. A similar factor which was included in Godot door implementation where it was flawed. However, this was not the case with the lever implementation, which was observed to function similar to that of in Amnesia. The torque was being applied properly on the lever, with the mouse gestures being interpreted differently depending on the player's position. The most notable difference between the two systems being the smoothness derived from the PID controller. The way the PID controller gains are tuned, in addition to the damping, makes it appear slow. However, it still adheres to the concepts of natural mapping.

Both levers were observed to feature a -45 to 45-degree range of motion, with the hinge joint providing stiff constraints that allowed for no other transformations to occur. Thus, allowing for incredibly stable interactions. The levers are able to be pulled to each end without issues. This coupled with the stability, contrasts greatly with the issues encountered in the previous attempt. This suggests an improvement over the initial prototype.

The lever implementation within Amnesia was again observed to be sensitive to mouse input, with there seeming to be a limit on how much torque is allowed on the lever. This means that faster movements are not inherently responsive. The implementation within Godot does not feature the same type of sensitivity to input and a cap on the applied torque. However, due to the high damping and tuning of the PID gains, it cannot be manipulated intensely. Although, the mouse input does control the speed of the lever accordingly during smaller mouse movements.

When the lever in Amnesia is pulled to either end of the lever’s total range, it stops for a second before returning to the center. It also returns to the center whenever the interaction is stopped. While the stop at reaching either end is not implemented within Godot’s lever, the return-to-center behavior is with the PID gains tuned to give it slight springiness.

As consistent with other interaction types, no instability issues nor undesired transformations were observed.

Overall, this is a successful implementation. It closely resembles the lever behavior present in Amnesia, with only minor differences. Otherwise, the core functionality and stability function accurately. To conclude this subsection, the Godot implementation of the lever is mechanically accurate to the example system in terms of input mapping, and stability.

6.2 Overall Mechanical Accuracy and Natural Mapping

The comparative analysis conducted based on a detailed checklist within Amnesia: The Dark Descent, highlights that the replicated interaction system is largely successful. The mechanical accuracy of most replicated interaction types, strongly resembles the functionality and behavior of their counterparts within the benchmark system. The grabbing, wheel, and lever mechanics being notable successes. The door interaction however, featured major differences in behavior and input mapping. This assessment is purely grounded in the systematic comparison conducted against the detailed checklist that was established in the introduction.

Most of the success for the replicated mechanics stemmed from the accurate recreation of Frictional Games' natural mapping. Most interactions featured intuitive controls that accurately resembled the action being performed; with the only exception being the door mechanic, which featured faulty flipped rotations. Natural mapping was a specific focus within the checklist due to its importance for intuitive physics-based interactions that are core to FrictionalGames' interaction model.

A recurring pattern that was noticed throughout the comparative analysis was the frequent use of PID controllers to provide smoother results. They are utilized in most interaction types to some extent. For example, the grabbing mechanic utilizes it in manipulating the object towards the target point, while the lever utilizes it to provide intuitive interactions. However, their use was omitted in the wheel mechanic. This was due to them causing issues with oscillation, which is common with incorrectly tuned PID controllers. Its exclusion from the wheel mechanic's logic enabled it to achieve greater mechanical accuracy, as well as stability. Furthermore, stability was another noticed reoccurring pattern which primarily resolved issues encountered in the previous attempt, such as wobbling of wheels and angles that broke levers. This can be attributed to the choice of switching to proprietary logic for each interaction type. This allowed the logic for a mechanic to be contained within its own code base; resolving the issues encountered in the initial prototype where a universal system was attempted by reusing the grabbing logic within each interaction type. This caused the logic for each interaction to become convoluted and difficult to debug. These improvements highlight the benefits of using an iterative learning process, which is critical for implementing complex systems like the physics driven interaction system outlined in this thesis.

Overall, synthesizing the evaluations done based on the checklist, the implemented interaction system successfully captures the mechanical accuracy perceived in FrictionalGames' model. Although most nuanced aspects that FrictionalGames' interaction model is known for were not replicated, based on the checklist evaluations most implemented mechanics featured the core, mechanical functionalities and correct natural mapping. Therefore, the primary goal of replicating FrictionalGames' interaction model within Godot 4 was realized to a satisfactory degree within the scope of this thesis.

6.3 Success in Replicating Benchmark Functionality

The primary objective of the thesis was to replicate the core mechanics of FrictionalGames' interaction model, which would then be benchmarked for mechanical accuracy using Amnesia: The Dark Descent. This replicated model would be evaluated via a detailed mechanical checklist. The aim was to understand and implement the physics and interaction design principles at the foundation of FrictionalGames' interaction model, rather than simply copy.

The implemented interaction types that featured the strongest mechanical accuracy were grabbing, wheel, and lever mechanics; with doors only replicating stability. The strongest implemented mechanics closely resembled the benchmark system's mechanical behavior and input mapping as defined by the checklist. Most of the success can be attributed to the strategies detailed in Chapter 5, such as the choice to utilize a hinge joint for the wheel, which directly addressed shortcomings like the wheel wobbling from the previous attempt.

While most interaction types were a success, the door implementation was the biggest deviation from the benchmark; featuring a multitude of flaws, such as applying torque instead of forces, which undermined intuitive natural mapped control. This deviation stemmed from switch in logic during development. As noted in the implementation of hinged objects section in Chapter 5, the first attempted approach would have been the closest to the example system.

Additionally, minor deviations existed in successful mechanics as well. For example, as discussed in the grabbing mechanic's analysis, the player was able to push held objects through walls by walking into them. This could potentially be resolved by disabling collision between the player and the held object. This is the same approach used in FrictionalGames' own interaction model, as observed in the benchmark system. Any additional minor deviations observed with other mechanics primarily revolved around the tuning of PID controllers. Thus, they can likely be resolved through additional tuning.

These perceived deviations, which mostly involve capturing the desired 'feel' for an interaction, are common in physics based game development; with them being highlighted by FrictionalGames themselves (Grip 2008; Grip 2010a).

Overall, the core aim of mechanical replication as defined by the thesis was met satisfactorily. Although featuring varying levels of deviations in the mechanics, the project successfully demonstrated the ability to implement a functional set of Frictional-style interaction mechanics within Godot 4; with the benchmark highlighting key successes for most mechanics in stability, natural mapping, and responsiveness. As the focus was on mechanical accuracy which could be benchmarked via a checklist, it can be confidently said that this project represented a significant step forward from the previous attempt. This project additionally showcased the successful practical application of the principles outlined in Chapter 2.

6.4 Limitations of the Comparison and Checklist

Although the comparative analysis conducted via a checklist in section 6.1 provided a structured approach to evaluating the thesis' practical work, there are limitations in its process that have to be acknowledged. The aim for this section then stands to explore these limitations, not to invalidate the findings from the current analysis, but to provide context for its interpretation and suggest potential areas for further investigation.

The initial key limitation would be subjectivity. This is because defining ‘mechanical accuracy’ for a checklist that would be tested on observations of the benchmark system's mechanics might involve some subjective judgment. For example, interpreting how a force ‘feels’. This limitation inherently stems from the difficulty of translating detailed observations into fixed-response checklist items like a pass or fail, which can significantly oversimplify the nuanced mechanical behaviors.

The secondary key limitation would be the scope of the benchmark. This is because the checklist testing was conducted entirely on a self-created test scene within Amnesia: The Dark Descent. The limitation stems from the test scene only covering the core mechanics, which might not encompass every possible interaction scenario present in the full game or across all FrictionalGames’ titles. Thus, the checklist and its subsequently conducted benchmark are only representative of this specific context.

The third key limitation would be the exclusion of the subjective ‘feel’, as well as broader immersion from the benchmark. While these are crucial aspects to FrictionalGames’ design philosophy, they are notoriously difficult to measure objectively with a binary checklist (Kukshinov et al. 2025). Hence, being outside the defined scope of this specific evaluation, which primarily focuses on technical replication.

To conclude this section, as mentioned throughout this section the analysis of the benchmark system was primarily based on observation and interpretation of mechanical behavior. Therefore, no direct dissection of its source code or detailed design documents was performed for precise parameters during testing. This potentially subjective, observational approach meant that while overall behaviors were compared, it restricted the ability to compare precise technical details in the benchmark.

6.5 Comparative Analysis Summary

The key findings from the comparative analysis show that the replication of most interaction mechanics present in FrictionalGames' interaction model, such as grabbing, wheels and levers, was met satisfactorily. The only exception was the door mechanic, which deviated greatly from the benchmark system based on checklist evaluations.

Overall, the critical aspects such as natural mapping and stability were achieved for most interactions.

To provide a final statement for the replicated system, based on the checklist it successfully replicated most of the mechanical functionalities found in FrictionalGames' interaction model, thereby achieving the key objective of this thesis. The implemented system built upon prior insights and resolved most shortcomings from the previous attempt, which indicates a significant improvement over the initial prototype outlined in Chapter 4. All of this is supported by the findings discussed in Section 6.3.

The specific use of a self-created test scene within the benchmark game alongside a detailed mechanics-focused checklist proved to be a sufficient way to evaluate mechanical accuracy. This allowed for a targeted and, to some extent, objective evaluation of the replicated system’s mechanical accuracy, even when considering its limitations that were detailed in Section 6.4.

In conclusion, the replication of FrictionalGames’ interaction model served as an effective and demanding benchmark for exploring physics-driven interactions specifically within Godot. The implementation detailed in Chapter 5 did not only produce a working model, but also provided useful insights into the complexities involved with replicating such nuanced mechanics. Therefore, this practical work successfully fulfilled the exploratory objectives set out by the thesis. The observed deviations like interaction sensitivity and the inaccurately tuned PID gains could be addressed in future iterations.

7 Conclusion & Future Work

The core objective of this thesis was to replicate the physics-driven interaction system that appears in titles by FrictionalGames. This was to be conducted within the open-source game engine Godot 4, primarily, with a focus on replicating the key interaction types (grabbing, doors, wheels, levers), which were subsequently benchmarked and analyzed.

The benchmark results clearly highlighted that most implemented interaction types showcased strong mechanical accuracy against the benchmark system, additionally achieving their natural mapping and stability.

The only exception to this replication was the door mechanic, which featured the most deviations from the benchmark system according to the checklist-based evaluation. Its greatest deviation was the differences in force application methods; the benchmark system applying forces, while the implemented mechanic utilized torque. The second deviation was its faulty input mapping which caused interactions to be flipped at certain angles.

Despite these deviations, the other implemented interaction types closely resembled their counterparts from the benchmark system. Therefore, it can be suggested that the thesis achieved its initial objectives and defined scope to a satisfactory extent.

7.1 Reflection on the Development Process and Learnings

The demanding task of replicating a complex system such as FrictionalGames' interaction model, required a streamlined development process. Hence, an iterative process was employed, which consisted of the following order: implementation, testing, and refinement. This iterative approach also enabled insights gathered in the previous attempt to be carried over. This helped in shaping the methodologies and technical choices made in the current, more successful implementation.

Evaluating the mechanical accuracy of the current implementation required a detailed comparative analysis, which systematically assessed each of the replicated mechanics against the benchmark system. A detailed mechanical checklist was used to conduct the analysis, which provided a structured and targeted framework for evaluating each interaction type individually.

The implementation process faced several challenges; however, none were more notorious than achieving a nuanced 'feel' for interactions. While this was a key target in the implementation, fully perfecting this aspect for each interaction type would have been a complex and time-consuming task. This is because of the subjective nature of the ‘feel’, which would have gone beyond the scope of simple mechanical replication. It requires extensive tuning of various parameters like physics properties and PID controller gains to accomplish, and is something even FrictionalGames invests immense amount of time into achieving (Grip 2010).

7.2 Limitations of the Current Work

Despite the thesis achieving its primary objective of replicating Frictional-style interaction mechanics, it is also important to acknowledge certain limitations in its methodology and scope. As discussed in Chapter 6, these limitations offer context for interpreting the findings and help suggest areas of focus for future investigations.

As highlighted in the comparative analysis, when the researcher is the one defining ‘mechanical accuracy’ and also evaluating it based on their observations, it holds a heavy weight of subjectivity, even with the inclusion of a detailed mechanical checklist. This is because the checklist and the subsequent benchmark were being crafted based on the researcher’s subjective judgments on aspects, such as the ‘feel’ of a force or the specific nuances of an interaction.

Furthermore, the analysis of the benchmark system was by necessity, primarily observational. This limited the comparison on precise technical details, such as exact force values or the tuning of PID gains used by FrictionalGames. This is a commonly faced issue when attempting to replicate proprietary systems. As noted in Chapter 3, specific low-level implementation details of FrictionalGames’ interaction algorithms are largely undocumented (Grip 2008; Grip 2010a). Despite them releasing the source code for Amnesia: The Dark Descent, it is presented in the high-level programming language C++ (FrictionalGames 2020). Hence, the analysis was limited to observable behaviors and interpretation rather than direct dissection of the benchmark’s source code.

The scope of the benchmark was constrained due to a self-created test scene within Amnesia. This limited the analysis to specific scenarios, which might not represent every interaction scenario present across Amnesia: The Dark Descent or all FrictionalGames’ titles. Additionally, Amnesia: The Dark Descent is a game released a decade ago, which suggests it could be outdated as FrictionalGames might have improved the nuances of the interactions since then.

The systematic evaluation, which was conducted via a detailed checklist, had a primary focus on the mechanical replication of the interaction mechanics from FrictionalGames’ titles, thus, directly omitting subjective aspects such as ‘game feel’ or player immersion from the analysis. Although immersion was key in understanding FrictionalGames’ design philosophy, the thesis's evaluation tools were primarily geared towards objective and observable mechanical behaviors. The choice to omit subjective aspects stemmed from their notoriety as difficult to measure (Kukshinov et al. 2025).

7.3 Suggestions for Future Work

While this thesis achieved its core objectives, it additionally uncovered several interesting paths for further development. This section serves to discuss those interesting paths, which might stem from the limitations discussed in Section 7.3 and from the inherent complexities related to replicating nuanced physics-driven interactions.

Beginning with improvements to the practical work done for this thesis. As discussed in Section 7.1, most interaction types were replicated accurately based on the mechanical checklist. However, the door mechanic was explicitly highlighted as an exception to this success. Therefore, improvements to its logic and input mapping would be the primary suggestion for future work from this replicated system. Additional deviations were observed with other interaction types, such as the ‘surface-clipping’ bug with the grabbing mechanic being secondary suggestions.

Additional tuning of PID gains and physics properties such as mass and damping to achieve the accurate ‘feel’ for each interaction type serves as the third suggestion. This suggestion is mostly relevant for the object grabbing and lever mechanics. While being deemed successful, they still featured minor deviations, particularly in the context of ‘feel’. Investing additional time into the tuning of PID gains and the physics properties, would allow these interactions to become accurate to the benchmark system, as well as more responsive and smoother.

Moving away from the setbacks of the replicated system, we suggest broadening the scope. For example, implementing additional features such as the push mechanic, "push mechanic" for heavier objects and sliding objects such as drawers. These would serve to broaden the catalog of mechanics closer to what exist in FrictionalGames’ titles. Although sliding doors were a mechanic replicated within the current practical work, they were omitted from the discussion as they did not exist in the benchmark system, and because of that its mechanical accuracy could not be evaluated. In addition to replicating other interaction types, more complex interaction scenarios could be suggested, such as combining multiple interaction types into a single object.

To gain a more holistic understanding of the system's efficiency, future work should include play testing to some extent. They would provide a direct player-centric perspective to the evaluation, which would provide invaluable information on the perceived intuitiveness, the subjective ‘feel’, and the overall success of the natural mapping in interactions (Norman 2013). These insights could then inform further iterations. Additionally, this could be taken further by attempting to evaluate player immersion more directly. Although modern methods to measuring immersion have not been credible, that might have changed by the time the future work is conducted (Kukshinov et al. 2025). This measurement would provide valuable insights into how well the system fosters a sense of presence (PI) and plausibility (Psi), which are key components of an immersive experience based on FrictionalGames' design philosophy (Grip 2013).

Finally, a deeper technical exploration into the Jolt Physics backend within Godot, would provide invaluable additional insights into implementing these physics-driven mechanics. Alongside this, the development of specialized in-editor tuning tools tailored for these types of interactions could significantly expedite the refinement process, making it easier to achieve the desired ‘feel’.

7.4 Concluding Remarks

In essence, this thesis successfully replicated the core mechanics of FrictionalGames’ renowned physics-driven interaction model within the Godot 4 game engine (Grip 2008; Grip 2010a). The conducted comparative analysis grounded in a detailed mechanical checklist, ultimately demonstrated that the replicated system closely resembled the fundamental mechanical functionalities and natural mapping principles of the benchmark system to a satisfactory degree. Therefore, fulfilling the thesis's central objectives. This success can be attributed to the iterative development process, which built upon insights from previous attempts (Aun 2023), as well as the structured evaluation against the example system.

Besides the satisfactory outcome of the prototype, the practical work provided considerable insight into the complexities of replicating such nuanced interaction mechanics. The development process detailed in Chapter 5 underscored the meticulous attention to detail required to even approach the characteristic ‘feel’ and immersive qualities that define FrictionalGames’ approach. Therefore, this thesis served as an effective and demanding exploration of physics-driven interaction design, particularly within the context of the Godot game engine.

Ultimately, this thesis contributes to the understanding of Frictional-style interaction design, by providing detailed documentation on a successful attempt at its replication. Furthermore, it showcased a structured methodology for its development, and detailed evaluation conducted within the Godot 4 game engine. The knowledge gained from these steps provide a foundational framework, which enables it to be potentially expanded upon with additional features or refinement through an iterative development process.

References

Abtahi, P., Hough, S. Q., Landay, J. A., and Follmer, S. 2022. Beyond Being Real: A Sensorimotor Control Perspective on Interactions in Virtual Reality. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 358, 1–17. https://doi.org/10.1145/3491102.3517706

Amnesia Wiki. 2025. HPL Engine. Retrieved May 1, 2025, from https://amnesia.fandom.com/wiki/HPL_Engine.

Åström, K. J. and Hägglund, T. (1995). PID Controllers: Theory, Design, and Tuning. 2nd ed. Research Triangle Park, NC: Instrument Society of America.

Aun, F. 2023. “Physical Interaction System in Godot 4”. Accessed April 19, 2025. https://summit-2324-sem1.game-lab.nl/2023/09/27/physical-interaction-system-in-godot-4/.

Baraff, D. 1997a. "An introduction to physically based modeling: Rigid body simulation i-unconstrained rigid body dynamics." SIGGRAPH '97 Course Notes. New York, NY: ACM SIGGRAPH, p. 82. Accessed May 1, 2025. Available from: https://www.cs.cmu.edu/~baraff/sigcourse/notesd1.pdf.

Baraff, D. 1997b. "An introduction to physically based modeling: rigid body simulation ii—nonpenetration constraints." SIGGRAPH '97 Course Notes. New York, NY: ACM SIGGRAPH, D31–D68. Accessed May 1, 2025. Available from: https://www.cs.cmu.edu/~baraff/sigcourse/notesd2.pdf.

Belga, J., Skarbez, R., Hmaiti, Y., Chen, E. J., McMahan, R. P., and LaViola, J. J. 2025. The Fidelity-based Presence Scale (FPS): Modeling the Effects of Fidelity on Sense of Presence. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (CHI '25), 1–15. http://dx.doi.org/10.1145/3706598.3713566.

Boonen, C. S. and Mieritz, D. 2018. "Paralysing Fear: Player Agency Parameters in Horror Games." In Proceedings of Nordic DiGRA 2018 Conference. Tampere: DiGRA. Accessed April 28, 2025. doi:10.26503/dl.v2018i3.1051.

Brown, E. and Cairns, P. 2004. "A grounded investigation of game immersion." In E. Dykstra-Erickson (ed.) *CHI '04 Extended Abstracts on Human Factors in Computing*

Systems. New York, NY: Association for Computing Machinery, 1297–1300. Accessed April 28, 2025. doi:10.1145/985921.986048.

Cairns, P., Cox, A., Nordin, A.I., 2014. Immersion in Digital Games: Review of Gaming Experience Research, in: Handbook of Digital Games. Wiley, pp. 337–361. https://doi.org/10.1002/9781118796443.ch12.

Calleja, G. (2011). In-Game: From Immersion to Incorporation. Cambridge, MA: The MIT Press. doi:10.7551/mitpress/8429.001.0001.

Carpio, R., Birt, J., and Baumann, O. 2023. Using case study analysis to develop heuristics to guide new filmmaking techniques in embodied virtual reality films. Creative Industries Journal, 18(1), 3–24. https://doi.org/10.1080/17510694.2023.2171336.

Davies, R. (2009). Interview: Frictional Games. Dead Pixel Post. Accessed May 4, 2025. https://deadpixelpost.blogspot.com/2009/02/interview-frictional-games.html.

Domingues, J. M., Filipe, V., Luz, F., and Carita, A. 2023. The Impact of Perceived Challenge on Narrative Immersion in RPG Video Games: A Preliminary Study. https://doi.org/10.33965/MCCSIS2023_202304C009.

Entertainment Software Association (ESA). (2024). 2024 Essential Facts About the U.S. Video Game Industry. Accessed 17 May 2025. https://www.theesa.com/resources/essential-facts-about-the-us-video-game-industry/2024-data/.

Ericson, C. (2004). Real-time collision detection. Boca Raton: CRC Press. doi:10.1201/b14581.

FrictionalGames. 2020. Amnesia: The Dark Descent. GitHub. Accessed May 2, 2025. https://github.com/FrictionalGames/AmnesiaTheDarkDescent.

Godot Engine. 2025. “Using Jolt Physics.” Accessed April 19, 2025. https://docs.godotengine.org/en/stable/tutorials/physics/using_jolt_physics.html.

Grip, T. 2008. ’Penumbra: Black Plague’ Developer Diary #3 – Screens. Accessed April 17, 2025. https://frictionalgames.com/2008-01-penumbra-black-plague-developer-diary-3-screens/.

Grip, T. 2009a. Future of adventure game interaction. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2009-12-future-of-adventure-game-interaction/.

Grip, T. 2009b. Puzzles in horror games Part 3: Why physics puzzles is not the "promised land" of adventure games. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2009-08-puzzles-in-horror-games-part-3/.

Grip, T. 2010a. Let's not forget about Physics. Accessed April 17, 2025. https://frictionalgames.com/2010-09-lets-not-forget-about-physics/.

Grip, T. 2010b. Closer look at teaser and more. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2010-02-closer-look-at-teaser-and-more/.

Grip, T. 2011. "Evoking Emotions and Achieving Success by Breaking All the Rules." YouTube video. Available from: https://youtu.be/zBhiE6QZjzY?si=S_Dh2SU3yoIXvwfg. Accessed April 17, 2025.

Grip, T. 2013. The five foundational design pillars of SOMA. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2013-12-the-five-foundational-design-pillars-of-soma/.

Grip, T. 2016. SOMA: Crafting existential dread. GDC Vault. Accessed May 4, 2025. https://gdcvault.com/play/1023399/SOMA-Crafting-Existential.

Grip, T. 2017a. Evoking presence. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2017-04-evoking-presence/.

Grip, T. 2017b. Mental models. Frictional Games. Accessed May 4, 2025. https://frictionalgames.com/2017-04-mental-models/.

Hadzinsky, C. A. 2014. A Look into the Industry of Video Games Past, Present, and Yet to Come. Senior Thesis, Claremont McKenna College. Scholarship @ Claremont. https://scholarship.claremont.edu/cmc_theses/842/.

Hermansson, M. 2023. "Godot Jolt." GitHub repository. Available from: https://github.com/godot-jolt/godot-jolt. Accessed April 19, 2025.

Horreur 404. 2023. "Frictional Games: The Bunker, a turning point for the series?" Accessed April 19, 2025. https://www.pointnthink.fr/en/frictional-games-the-bunker-a-turning-point-for-the-series/.

Iacovides, I., Cox, A., Kennedy, R., Cairns, P., and Jennett, C. 2015. "Removing the HUD: The Impact of Non-Diegetic Game Elements and Expertise on Player Involvement." In P. Cairns, M. S. Hancock, M. R. Gibbs, and L. E. Nacke (eds.) Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '15). New York, NY: Association for Computing Machinery, 13–22. Accessed April 29, 2025. doi:10.1145/2793107.2793120.

IMARC Group. (2024). Gaming Market Size, Industry Share, Growth Report, 2033. Accessed 17 May 2025. https://www.imarcgroup.com/gaming-market.

IndieDB. 2010. HPL Engine. Accessed May 4, 2025. https://www.indiedb.com/engines/hpl.

Ishii, H. and Ullmer, B. 1997. "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms." In S. Pemberton (ed.) Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97). New York, NY: Association for Computing Machinery, 234–241. Accessed April 28, 2025. doi:10.1145/258549.258715.

Johnson, W. L. (2005). "Game Physics." Physics Today. 58(3), 64. Accessed April 30, 2025. doi:10.1063/1.1897568.

Kukshinov, E., Tu, J., Szita, K., Senthil Nathan, K., and Nacke, L. E. 2025. Widespread yet Unreliable: A Systematic Analysis of the Use of Presence Questionnaires. *Interacting with Computers*, iwae064, https://doi.org/10.1093/iwc/iwae064.

Leadwerks. 2013. Leadwerks interviews Frictional Games co-founder Thomas Grip. Leadwerks Community. Accessed May 4, 2025. https://www.leadwerks.com/community/blogs/entry/557-leadwerks-interviews-frictional-games-co-founder-thomas-grip/.

Li, S., Ma, Y., Yan, J., Dai, B., Peng, Y., Zhang, C., and Zhu, Y. 2025. "A simulation-heuristics dual-process model for intuitive physics." arXiv preprint arXiv:2504.09546. https://arxiv.org/abs/2504.09546. Accessed April 30, 2025.

McEwan, M. W. 2017. The influence of naturally mapped control interfaces for video games on the player experience and intuitive interaction. PhD thesis, Queensland University of Technology. QUT ePrints. https://eprints.qut.edu.au/107983/.

McEwan, M. W., Blackler, A. L., Johnson, D. M., and Wyeth, P. A. 2014. "Natural mapping and intuitive interaction in videogames." In Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '14). New York, NY: Association for Computing Machinery, 191–200. Accessed April 28, 2025. doi:10.1145/2658537.2658541.

McMahan, R., Bowman, D., Zielinski, D., and Brady, R. (2012). "Evaluating Display Fidelity and Interaction Fidelity in a Virtual Reality Game." IEEE Transactions on Visualization and Computer Graphics. Vol. 18 (April 2012), 626-633. doi:10.1109/TVCG.2012.43.

Millington, I. (2007). *Game Physics Engine Development*. Boca Raton: CRC Press. doi:10.1201/9781482267327.

Norman, D. A. (2013). The design of everyday things. Revised and expanded ed. New York: Basic Books.

Norrman, A. 2020. User Interface’s Impact on Player’s Immersion. Dissertation, Umeå University. DiVA portal. https://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-170644.

Pagulayan, R., Keeker, K., Wixon, D., Romero, R. L., and Fuller, T. 2012. "User-Centered Design in Games." In J. A. Jacko (ed.) *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. Third Edition. Boca Raton: CRC Press, 883–906. Accessed April 28, 2025. doi:10.1201/b11963-39.

Rapier. 2025. Joint constraints. Accessed April 28, 2025. https://rapier.rs/docs/user_guides/rust/joint_constraints/.

Seabra, M., Fernandes, F., Lopes, D., and Madeiras Pereira, J. 2023. "Position Based Rigid Body Simulation: A comparison of physics simulators for games." In WSCG 2023: Full Papers Proceedings: 31st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. Brno: Václav Skala - UNION Agency, 351–360. Accessed May 1, 2025. doi:10.24132/CSRN.3301.59.

Shafer, D., Carbonara, C., and Popova, L. (2014). "Controller Required? The Impact of Natural Mapping on Interactivity, Realism, Presence, and Enjoyment in Motion-Based Video Games." Presence: Teleoperators and Virtual Environments. 23(3), 267–286. Accessed April 29, 2025. doi:10.1162/PRES_a_00193.

Shneiderman, B. 1997. "Direct manipulation for comprehensible, predictable and controllable user interfaces." In *Proceedings of the 2nd International Conference on Intelligent User Interfaces (IUI '97)*. New York, NY: Association for Computing Machinery, 33–39. Accessed April 28, 2025. doi:10.1145/238218.238281.

Slater, M. 2009. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philos Trans R Soc Lond B Biol Sci.*, 364(1535), 3549-57. doi: https://doi.org/10.1098/rstb.2009.0138.

Slater, M., Steed, A., and Usoh, M. 2013. *Being There Together: Experiments on Presence in Virtual Environments (1990s)*. Technical Report, Department of Computer Science, University College London, UK. Available from: https://publicationslist.org/data/melslater/ref-233/beingthere%202013.pdf. Accessed April 23, 2025.

Smith, L. 2023. Creating tension through game-centric design in survival horror video games. Bachelor's thesis, Tampere University of Applied Sciences. Theseus. https://urn.fi/URN:NBN:fi:amk-202304054771.

Song, S., Lan, L., Yao, J., and Guo, X. (2022). "Continuous collision detection with medial axis transform for rigid body simulation." Communications in Information and Systems. 22(1), 53–78. Accessed May 1, 2025. doi:10.4310/CIS.2022.v22.n1.a3.

Sweetser, P., and Wyeth, P. (2005). "GameFlow: a model for evaluating player enjoyment in games." Computers in Entertainment. Vol. 3, No. 3 (July 2005), 3. doi:10.1145/1077246.1077253.

Swink, S. (2008). Game Feel: A Game Designer's Guide to Virtual Sensation. United Kingdom. CRC Press.

Tang, J. 2024. "Harnessing Machine Learning to Model Intuitive Physics: Insights from Game Engines and Cartoon Simulations." In M. ISTANBULLU (ed.) Proceedings of the 2nd International Conference on Machine Learning and Automation. Applied and Computational Engineering, Vol. 104. Oxford, UK: EWA Publishing, 189–194. Accessed April 30, 2025. doi:10.54254/2755-2721/104/20241193.

Unity Technologies. 2025. Unity - Manual: Joints. Accessed April 28, 2025. https://docs.unity3d.com/6000.0/Documentation/Manual/Joints.html.

Walter, M. E. 2023. ‘Being’ Alyx Vance : an autoethnographic analysis of immersion as experienced within narrative-based virtual reality video games. PhD thesis, University of Pretoria. UPSpace. https://repository.up.ac.za/handle/2263/94241.

Zook, A., Fruchter, E., and Riedl, M. O. 2019. "Automatic Playtesting for Game Parameter Tuning via Active Learning." arXiv preprint arXiv:1908.01417. https://arxiv.org/abs/1908.01417. Accessed May 1, 2025.

Declaration

I acknowledge the use of AI-assisted tools during the writing and editing of this thesis. These tools were used to improve writing speed, the structure, and grammar without altering the originality or academic integrity of the thesis.

Tools Used:

1. **Superwhisper**
   Link: https://superwhisper.com/ (Accessed June 2025)
   Use: Tool was used to increase the speed of writing through dictation. Initially used with a custom academic writing prompt. It aimed to transcribe dictated speech and refine it into formal academic prose. An example of such a prompt might have been: *"You are an AI assistant for academic text generation. Convert the following dictated speech into clear, concise, and academically appropriate language. Ensure a formal tone, precise vocabulary, and correct grammatical structure, suitable for thesis writing. Eliminate conversational filler and redundancy."*
   However, this approach resulted in robotic responses, and was switched in favor to one of Superwhisper's pre-made writing templates or 'modes' known as 'Message'. This mode helped transcribe spoken thoughts directly into clear, readable text, removing common speech errors. Output was manually adjusted to fit the wording and flow of the target sentence/paragraph and thesis's overall voice.

2. **ChatGPT (OpenAI)**
   Link: https://chat.openai.com (Accessed June 2025)
   Use: Tool was used to improve the flow between sentences and transitions between paragraphs, find synonyms for words, and help brainstorm names for chapters and their subsections.
   Examples of prompts used:
   - *"Could you help me improve the flow between these sentences/paragraphs: [insert text]"*
   - *"Help me create a smooth transition between these paragraphs: [insert text]"*
   - *"Suggest some synonyms for '[word or phrase]' in the context of [brief description]"*
   - *"Help me brainstorm names for a chapter/subsection based on the following content: [insert text or brief description]"*

Output was used as a reference to adjust existing text after review. It was ensured that the end result aligned with the thesis's original purpose and voice.

3. **Perplexity**
   Link: https://www.perplexity.ai/ (Accessed June 2025)
   Use: Tool was used to conduct deeper research into niche topics. It provided summaries of its findings along with lists of relevant sources. Prompts used were detailed research queries that attempted to cover a huge area specific to the niche topics.
   Output was used to expand knowledge on the thesis's topic, gather supporting sources for other references, and inspire further research.