

Project Proposal - MComp Research Project - Ashley Cromack - CRO12450621

Investigating dynamic terrain as a novel mechanic to solve puzzles in games within a real-time physics simulation context

Introduction

In the sphere of video games, there is a certain line of thinking that aims to instill as much realism into games as possible, to make them a truer approximation to the world around us and thus deepening the level of immersion that the player feels. One way in which developers try to achieve this goal is to implement a system of physics simulation in their games, this is such a popular component to games now that most games have some form of it influencing the way their game plays and there are even multiple middleware solutions to aid developers' implementation of it into their titles.

In regards to physics simulation, there are a number of approaches developers can take to implementing it. One such way, one which many academics and their subsequent studies tend to investigate, is that of creating the most accurate simulation of real-world physics as possible. This typically involves an in-depth understanding and use of equations and accurately recreating the properties of real world materials, such as accurately modelling soil and how objects such as tires may interact with them. To simulate this interaction as accurately as possible, a large amount of complex operations need to be executed and every small detail needs to be considered at every time step. This tends to lead to simulations that require a greater amount of time to calculate and render each frame than would be appropriate for a video game, which tend to require a response time in milliseconds to achieve a simulation that feels responsive to the player.

Hence, in games that do implement physics simulation, they tend to utilise a more simplified model of how the world operates mechanically (for instance considering objects as simple rigid bodies or particles). This allows the amount and complexity of the mathematical functions involved to be streamlined massively, allowing the simulation to respond much sooner to the players' inputs and thereby demonstrating the impact they're having on the world in real-time, making it a much more viable system for use in games.

One possible use for a physics engine is to model terrain and making it dynamic, allowing the player to interact with the terrain and see the effects incurred on it. However, when it comes to the application of terrain deformation in modern video games, there isn't a wide range of titles to look to as examples. One of these examples is *Spintires* (Zagrebelnyj, 2014), an off-road driving simulator game that tasks the players with transporting cargo through a range of environments. This game utilizes the *Havok* (Havok, 2011) physics engine to simulate how the muddy terrain deforms when the player drives over it, added a depth of complexity to the game as players must be careful not to get themselves stuck. Much like the serious simulations mentioned prior, this game takes a more serious approach to the physics of the world. This means that players will need a decent understanding of how to control their vehicle on a variety of surfaces, such as

wet mud, rocks and water, as well as how the tires will impact the terrain's surface. This can make the game somewhat unfriendly to newcomers, especially those with only a rudimentary understanding of how vehicles behave with terrain, so a more streamlined and simplified approach to this deformation and how to cause it may help to foster the players' understanding of how it works.

One of the most prevalent components to games, one which can often become a genre in and of itself, is that of puzzle solving. Typically, the most popular puzzle games on the market are 2D affairs that utilise a simplistic approach to both presenting puzzles to the user and presenting them with the means to solve them. However, with the advent of *Portal* (Valve Corporation, 2007) in 2007, Valve experimented with developing a whole puzzle game around the simulation of physics (in this case through momentum and how projectiles interact with surfaces). This proved to be an incredibly popular title which, along with Valve's other implementation of a physics engine in their *Half Life 2* (Valve Corporation, 2004) titles, showed that the idea of utilising a simulation of real-world physics as a game mechanic was one that the games industry, and indeed the consumers of said industry, were more than receptive to.

This project looks to investigate physics simulation, in particular the implementation of deformable terrain in this context, and how it can be applied as a principle mechanic in a video game. The key concept of this resulting game will be to present a player with a puzzle to solve and the key tool at their disposal will be the ability to deform the terrain around them, causing them to consider how they can impact their game world to their advantage and be aware of some of the challenges this may throw in their path. This work will therefore be attempting to explore new ground with the way physics systems are implemented in games, using dynamic terrain in this context to create new and unique mechanics in the field of video-game puzzle solving.

Aims and Objectives

The aim of this project is to experiment with the concept of deformable terrain as a game mechanic to solve puzzles, using a real-time physics simulation context to model this in an intuitive way. This deformable terrain concept will therefore be implemented as a novel mechanic to solve puzzles in a game setting and throughout the course of the project the impact on players will be evaluated, investigating how it affects their perception of the puzzles presented to them and how they can solve them using the tools at their disposal.

To meet these aims, the following objectives will need to be met through the course of the project:

- To investigate and experiment with the current implementations of physics systems and middleware to find a suitable basis to build a real-time solution for terrain deformation, such as *PhysX* and *Bullet*. Additionally, to explore ways to implement this system into a game, potentially through existing games engines like *Unity* and *Unreal Engine*.

- The game should go through an extensive design and prototyping process to ensure that it delivers an enjoyable and coherent experience for the players, with focus on conveying the mechanics and how they operate clearly.
- To successfully develop the game into a functioning prototype, demonstrating the implementation of terrain deformation that responds in an appropriate real-time context to the player's interactions with it. The game will be designed with this player-controlled terrain deformation in mind as the primary tool to solving puzzles.
- To evaluate the effect that using dynamic terrain as a game mechanic has on the gameplay experience and how it influences their approach to puzzle-solving. To do this the artefact will need to be presented to a group of play-testers to investigate what their experience with the concept was like, see what components they enjoyed, and which features they gravitated towards, along with more quantitative data from logging their interactions with the game.

Literature Review

Models of Terrain Deformation

While most models of deforming terrain focus on the surface being a form of soil, there are cases where a real-time simulation can be used to deform sand in response to the wind and the vegetation strewn around the environment (Wand and Hu, 2009). This approach's main aim is to produce a visually realistic approximation of a desert; however, they also place focus on implementing the method on the GPU to parallelize the process to make it more suitable in a real-time setting. This use of the GPU to optimise the performance of the physics simulation could be an important consideration to ensure that the game produced in this project performs adequately for the gameplay.

When it comes to modeling the interactions of vehicles with dynamic terrain, there are many established approaches. One such model is investigated in a paper where the academics present a scientific model of how to simulate a tire's interactions with a terrain and how it deforms said terrain (Madsen et al, 2012). Where this approach diverges is that it goes on to present the model in a serious context, measuring the amount of energy or power the vehicle needs to deform the terrain, in order to produce a realistic driving simulator for the military. However, many of the key concepts should still be applicable when developing the dynamic terrain for a game setting like with this project.

It is important to consider that, in these cases of vehicles deforming the terrain, the ground should not only look altered but also impact how objects interact with it. One way is to propose the use of modern GPU features in their dynamic terrain visualization method to alter the topology of the terrain being deformed (Wang et al., 2009). The implementation they achieve purportedly runs at a high frame-rate, making it perfectly suitable for an interactive 3D application, such as a video game as with this project.

In a similar paper, academics once again look at how to modify terrain as a result of interactions with a vehicle, however the focus shifts to implementing it on a larger scale of terrain (Zhang et al., 2010). They specifically use a GPU-based terrain deformation algorithm to help generate said alterations to the terrain, using this in conjunction with shaders and OpenGL. This is a much more manual approach, and fails to produce a visually realistic scene, but it also manages to run at a high frame-rate suitable for games, so it may therefore prove a useful approach to consider when developing the game for this project.

Game Design, Implementation and Evaluation

When designing a game, it can be important to consider an established framework along the way, to ensure that the end result has the desired impact on the players. One such framework in the field of game design is the MDA approach to game design and research (Hunicke et al., 2004). This framework outlines a process of design by considering the game from a data representation standpoint, all the way up to the end experience that the player has, along with the connection between these two things. This system also provides a way to approach investigation and critique of games, which will come to the forefront for the play-testing stage of this project.

An important component to any game is how it is rendered and presented to the player as a complete package, frameworks can be useful guidelines to accomplishing this. One framework proposed by academics looks at the generation, modelling and rendering of a game world (Catanese et al, 2011), one component of which being the support of a physics engine (in this case using the *PhysX* middleware). This framework therefore not only looks at how to produce the game world, but also how to simulate some of the objects within it and the interactions that they can have with each other.

In terms of how to evaluate the effect that a game can have on the players, and indeed the effects of using a simulation of Newtonian physics in these games, it's vital to look at how others have conducted their investigations. In one such study looked into helping people to learn physics by exposing them to computer simulations. In this case they investigated the efforts of the physics-based game *Portal*, measuring whether playing it caused improvements in the player's physics intuitions and spatial cognition skills (Adams et al., 2016). While the results may be important to consider, the way in which they conduct this study can be useful for evaluating this project in the play-test stage.

Particle/Mesh Physical Models

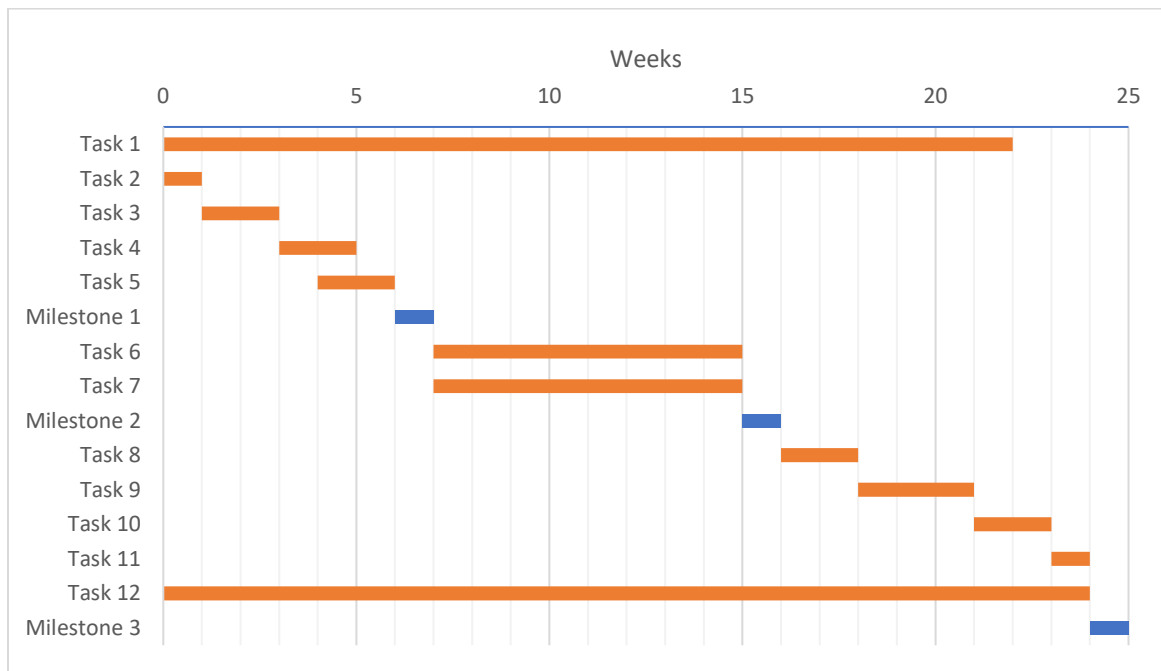
In another approach to modeling dynamic terrain in a driving simulation, a main focus of the paper is on the implementation of such a simulation with multiple displays, however they also detail the process by which they create their dynamic terrain (Ni et al., 2009). In this study they use the Real-time Optimally Adapting Mesh (ROAM) algorithm, discuss how they generate the deformable terrain with this algorithm, the use of height-maps in this system and the step-by-step process in which the terrain mesh is generated. This in-depth look at their approach may therefore be a useful insight into how to generate the dynamic terrain in an adaptable fashion.

A different approach to modeling the ground is to simulate the soil as being loose, using a more particle-based model (Chanclou et al., 1996). Their application of this system allowed the simulated soil to compress and pile, meaning the vehicles interacting with it would leave tracks and even sink into the ground. Employed on a large scale, this solution appears unsuitable for a real-time application, due to its heavy resource overhead requiring multiple seconds to process each frame. However, implemented as a smaller component in a real-time simulation, this system may still be viable therefore it could be a system to investigate for a smaller component of the project's problem-solving gameplay.

As demonstrated with the varying studies produced on the subject of dynamic terrain, there are some approaches that take a simpler mesh-based stance, whereas some use a more complex particle representation. However there is also the potential to use a combination of these two methods, to produce a model that presents a good approximation of shifting terrain around and leaving depressions in the ground, while still being efficient enough with resources to make it applicable for real-time applications such as games (Prautzsch et al., 2009). This model may therefore prove to be an invaluable study when considering how best to implement the terrain deformation in this project in order to achieve the aims.

Project Plan

The plan for this project comprises of the following Gantt chart, with the associated tasks being detailed beneath the chart. The milestones of key points throughout the course of the project have been highlighted in bold.



Week 0: 26th October 2017 | Week 25: 26th April 2018

- Task 1: Research existing literature in the subject domain, making notes of their relevance and consider how their findings could be used to shape this project. This will be an ongoing process throughout this project.
- Task 2: Investigate the existing physics and game engines available for use, making selections appropriate for this project.
- Task 3: To experiment with the chosen physics and game engines to see what their feature set is.
- Task 4: To draft designs and prototypes for how the game could play and function, consider how dynamic terrain could be used in a unique way to solve puzzles
- Task 5: To refine these concepts to produce a definitive design document that will lay out the functionality the game should have, how it should play and how it should be presented,
- **Milestone 1: Have a completed design document for the game, highlighting how the game should look, play and behave. The physics and game engines to be used should also be selected.**
- Task 6: Develop the game, using the previously chosen physics and games engines, according to the specifications laid out in the game's design document
- Task 7: Test out the state of the game and make any necessary tweaks along the way, keeping track of these and noting why they may have been required
- **Milestone 2: The development of the game should be complete, ready to be presented to a selection of players to test it**
- Task 8: To consider the information required from the user testing phase, write up a post-test questionnaire and a consent form for the play-testers
- Task 9: Put a call out for people who would be willing to play-test the game from this project and answer a few questions about their experience with it.
- Task 10: Carry out the play testing phase, allotting time for all the previously sourced play-testers to try out the game, retrieve a log of their play sessions and ask them questions about their experience.
- Task 11: Compile the results of the user testing phase and evaluate their experience
- Task 12: Write up the project report. The continual, overlapping nature of this task demonstrates that this will be an ongoing process throughout the project, with small additions and alterations being made throughout.
- **Milestone 3: The project should be fully completed by this point**

The milestones, denoted by the blue bars, also comprise the slack weeks available in this project, totaling 3 weeks of available time should any of the tasks require it.

Risks

As with all projects, there are the typical generic risks with regards to personal issues arising and data being lost, these are not factored into the following break-down, but the necessary precautions will be taken to mitigate them. For instance, all data regarding the project will be backed up on multiple devices and stored in an online service continually to ensure no loss of work.

Green = Low Risk (1% - 35%)

Yellow = Moderate Risk (36% - 70%)

Red = High Risk (71% - 100%)

Risk	Likelihood (1 = Not Likely 10 = Very Likely)	Impact (1 = Minimal Impact 10 = Severe Impact)	Risk Quotient	Mitigation
The testers will be unfamiliar with the game and may therefore be confused and unable to play it	7	7	49	When the players are introduced to the game they will be presented with a document to explain the mechanics of the game, the controls and what their goal is. They are also free to ask any questions should they arise
It may be challenging to conceptualise suitable puzzles to make use of the game mechanics	6	8	48	The designing and prototyping phase has been allotted a reasonably substantial time-frame to ensure that ideas can be fleshed out and built upon to ensure a polished concept.
Implementing the terrain deformation mechanically could be challenging than anticipated	6	7	42	This concept of physics simulation will be investigated before implementation to ensure a concrete knowledge. Appropriate time has also been allocated to this component of the project, ensuring that any issues ran into can be worked through.
The physics engine may be difficult to implement into an appropriate game engine	6	7	42	Appropriate time has been allocated in the project to ensure that the available physics and game engines have been investigated, thereby allowing time to ensure they can be used in conjunction appropriately.

A lack of familiarity with the software may make implementation challenging	4	9	36	Should any issues arise, and before this, the associated documentation for the software will be sought out and referred to.
The design document for the game may specify components that are not possible with the chosen physics or game engine	4	9	36	The design document will be produced while referring to the physics and game engines' documentation, ensuring that the necessary components can be implemented.
A lack of familiarity with the software may make implementation challenging	4	9	36	Should any issues arise, and before this, the associated documentation for the software will be sought out and referred to.
Due to unforeseen issues with individual testers, the play-test phase may take longer than anticipated	5	7	35	Additional time has been accounted for in the project plan should any tasks overrun, so any necessary time can be taken here.
May not be able to find enough people available for the user testing component	4	8	32	Searching for people to participate in the play-testing stage will take place well in advance to allow plenty of time to find enough participants

Bibliography

Adams, D.M., Pilegard, C. and Mayer, R.E. (2016) Evaluating the cognitive consequences of playing "portal" for a short duration. *Journal of Educational Computing Research*, 54(2) 173-195.

Catanese, S.A., Ferrara, E., Fiumara, G. and Pagano, F. (2011) Rendering of 3D Dynamic Virtual Environments. In: *4th International ICST Conference on Simulation Tools and Techniques*, Barcelona, Spain, 21-25 March. New York, USA: ACM, 351-358. Available from <https://dl.acm.org/citation.cfm?id=2151116> [accessed 15 October 2017]

Chanclou, B., Luciani, A. and Habibi, A. (2002) Physical Models of Loose Soils Dynamically Marked by a Moving Object. In: *Proceedings Computer Animation '96*, Geneva, Switzerland, 3-4 June 1996. New Jersey, USA: IEEE, 27-35. Available from <http://ieeexplore.ieee.org/abstract/document/540485/> [accessed 14 October 2017]

Havok (2011) *Havok* [software]. Dublin: Havok. Available from <https://www.havok.com/> [accessed 12 October 2017]

Hunnicke, R., LeBlanc, M. and Zubek, R. (2004). MDA: A formal approach to game design and game research. In: *Proceedings of the AAAI Workshop on Challenges in Game AI*, 4, 1-5.

Madsen, J., Negrut, D., Seidl, A., Reid, A., Ayers, P., Bozdech, G., Freeman, J. and O'Kins, J. (2012) A physics-based Vehicle/Terrain interaction model for soft soil off-road vehicle simulations. *SAE International Journal of Commercial Vehicles*, 5(1) 280-290. Available from <http://www.dtic.mil/docs/citations/ADA574119> [accessed 16 October 2017]

Ni, T., Zhao, D. and Zhang, H. (2009) Realistic Vehicle Driving Simulator with Dynamic Terrain Deformation. In: *IMCA 2009. International Conference on Mechatronics and Automation*, Changchun, China, 9-12 August. New Jersey, USA: IEEE, 4795-4800. Available from <http://ieeexplore.ieee.org/abstract/document/5246436/> [accessed 14 October 2017]

Pavel Zagrebelnyj (2014) *Spintires* [game]. Gorleston-On-Sea: Oovee Game Studios. Available from <http://www.spintires.com/> [accessed 11 October 2017]

Prautzsch, H., Schmitt, A., Bender, J. and Teschner, M. (2009) Soil Deformation Models For Real-Time Simulation: A Hybrid Approach. In: *Proceedings of the 6th Workshop on Virtual Reality Interaction and Physical Simulation*, Karlsruhe, Germany, 2009. Graz, Austria: Eurographics, 21-30. Available from https://www.researchgate.net/profile/Torsten_Kuhlen/publication/221622677_Soil_Deformation_Models_for_Real-Time_Simulation_A_Hybrid_Approach/links/0912f50c196bf67fbc000000.pdf [accessed 14 October 2017]

Valve Corporation (2007) *Portal* [game]. Bellevue: Valve Corporation. Available from <http://store.steampowered.com/app/400/Portal/> [accessed 14 October 2017]

Valve Corporation (2004) *Half Life 2* [game]. Bellevue: Valve Corporation. Available from http://store.steampowered.com/app/220/HalfLife_2/ [accessed 14 October 2017]

Wang, N and Hu, B.G. (2010) Aeolian Sand Movement and Interacting with Vegetation: A GPU Based Simulation and Visualization Method. In: *2009 Third International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications (PMA)*, Beijing, China, 9-13 November. New Jersey, USA: IEEE, 401-408. Available from <http://ieeexplore.ieee.org/abstract/document/5474660/?reload=true> [accessed 15 October 2017]

Wang, D., Zhang, Y., Tian, P. and Yan, N. (2009) Real-Time GPU-Based Visualization of Tire Tracks in Dynamic Terrain. In: *CiSE 2009. International Conference on Computational Intelligence and Software Engineering*, Wuhan, China, 11-13 December. New Jersey, USA: IEEE, 1-4. Available from <http://ieeexplore.ieee.org/abstract/document/5365552/> [accessed 14 October 2017]

Zhang, Y., Wang, D., Yan, N. and Shang, Y. (2010) Real-Time Visualization of Tire Tracks in Large Scale Dynamic Terrain. In: *3rd IEEE International Conference on Computer Science and*

Information Technology, Chengdu, China, 9-11 July. New Jersey, USA: IEEE, 263-266.
Available from <http://ieeexplore.ieee.org/abstract/document/5565005/> [accessed 15 October 2017]