Creating gameplay mechanics with deformable characters

Grant Clarke

This conference paper was presented at the 19th annual European Game-ON Conference (GAME-ON'2018) on Simulation and AI in Computer Games, Abertay University, Dundee, UK, 18-20 September 2018

CREATING GAMEPLAY MECHANICS WITH DEFORMABLE CHARACTERS

Grant Clarke
School of Design and Informatics
Abertay University
40 Bell Street, Dundee, United Kingdom, DD1 1HG
email: g.clarke@abertay.ac.uk

KEYWORDS

Physics-based Animation, Gameplay Mechanics, Soft Body Simulation

ABSTRACT

This paper presents how soft body simulation can create deformable characters and physics-based game mechanics that result in a more varied gameplay experience. A framework was implemented that allows the creation of a fully deformable soft body character within a games application where the simulation model properties could be altered at runtime to create gameplay mechanics based on varying the deformation of the character. The simulation model was augmented to allow appropriate methods of player control that complemented the character design and its ability to deform. It was found that while the implementation of deformation-based mechanics created a more varied gameplay experience, the underlying simulation model allowed for a limited amount of deformation before becoming unstable. The effectiveness of the framework is demonstrated by the resulting mechanics that are not possible through the use of previous methods.

INTRODUCTION

Many computer games use physics simulation as an underlying system to create real-world-like motion in objects. Whether it's to create the feel of driving vehicles over undulating terrain or have characters fall, ragdoll-like, down a staircase. In terms of game design, titles such as Marble Madness (Atari Games 1984) and Angry Birds (Rovio Entertainment 2009) use physics-based gameplay to create their core game mechanics. This can result in dynamic, but predictable behaviour that can provide players with a varied and engaging user experience.

These games are underpinned by rigid body physics simulation where the shape of the player character within game remains constant throughout the simulation. While this doesn't necessarily limit the different physics-based gameplay mechanics that can be created, it does suggest that the additional use of soft body simulation to create deformable characters could lead to some innovate player experiences in the area of physics-based gameplay.

Soft body dynamics in games isn't new, but its use is often limited to improving the visual aesthetic of digital characters rather than form a core part of the gameplay mechanics. For example, simulated cloth on character garments or providing secondary animation in the form of belly jiggle to make a character's movement appear more dynamic.

To create deformable characters there are several methods of soft body simulation, and the choice of which to use is dependent on application requirements in terms of performance, stability, deformation properties and the means of control.

Mass-spring models have been popular in games to create elastic deformable materials due to the relative simplicity and efficiency of the underlying implementation. The elasticity is controlled by manually adjusting the spring stiffness and damping constants to create the desired amount of deformation across the body.

Finite element methods can be used to create a more physically accurate model of deformable materials that contain both elastic and plastic properties. While more computationally expensive than a mass-spring model, interactive rates are possible. Values that define the deformation properties of real-world materials can be used with this model potentially reducing the amount of manual adjustment required. (Müller and Gross 2004)

Parker and O'Brien (2009) demonstrates the application of a FEM-based model to create realistic deformation and fragmentation in a game environment. The deformation of objects is solely used to improve the visual aesthetics, but nevertheless, this shows this method can be used in a fully featured commercial game.

Position based dynamics allows for more control as the position of the particles that form the soft body can be manipulated directly without compromising the integrity of the simulation model. It is also unconditionally stable and efficient while creating plausible, but not wholly accurate deformations.(Müller et al. 2007)

In terms of soft body control, Liu et al. (2013) proposes a skeleton-driven approach that drives the motion of an FEM-based soft body. Skeletal animation allows for creative control of the overall motion of the soft body character in cases where procedurally generating a similar motion would be too complex. The two-way coupling between the soft body skin and rigid body skeleton does allow for applied forces on the soft body to affect the underlying skeleton, but the rigid skeleton does limit the amount of deformation possible. Tan et al. (2012) proposes the use of simulated muscle fibres to control the motion of FEM-based soft bodies. This allows for the creation of more cartoon-like motion that exhibits a greater range of squash-and-stretch when compared to skeleton-based methods as the character motion is not restricted by the rigidity of an underlying skeleton. However, results show that this method is too computationally expensive for interactive rates required for game applications.

There are some notable game applications that use character deformation as part of their core mechanics. Gish (Cryptic Sea 2003) and Cats are Liquid (Last Quarter Studios 2015) are 2D platform games that let the player increase the elasticity of their main characters so they can squeeze through small gaps in the level.
Deformers (Ready At Dawn 2017) is a fast-paced multiplayer battle game where the elasticity of the characters helps to give the characters a more cartoon-like quality and their ability to change shape from a ball to a cube is used in the games blocking mechanic.
Claybook (Second Order 2017) uses a custom position based dynamics system to create a fully deformable environment where the player can use the shape-changing character to alter and interact with the world to solve puzzles.

Reviewing related work has shown that creating and controlling deformable characters is an active area of research in computer graphics, but has also shown there is a lack of literature into the design and implementation of soft body physics-based game mechanics.

By focusing on the unique behaviours that can be created through the use of soft body dynamics this paper aims to explore how deformable characters can be used to create a wider range of physics-based game mechanics.

METHODOLOGY

Two gameplay prototypes were created with different character representations. The following approach was taken when creating deformation-based game mechanics in each prototype:

- Identify the characteristics of the simulation model and the properties that can be manipulated to alter its behaviour.
- Define a method of controlling the basic movement of the character.
- Formulate deformation-based character behaviours that can be created by interacting with the underlying simulation model.
- Using one or more behaviours, create game mechanics that are generally not feasible via a more traditional rigid body character representation.

Identifying Characteristics of the Simulation Model

A FEM-based model was chosen because of its predominant use in existing research in creating deformable characters. The soft body is represented by a mesh of tetrahedral elements and it is the simulated movement of these elements that cause the deformation. The model is characterised by the following properties: elasticity, plasticity and fracturing.

Elasticity allows the body to deform when under stress. Stress is applied in the form of external forces to the individual elements that describe the body. Removing these external forces result in the body returning to its undeformed state. The elasticity of the body is controlled by two parameters, Young's modulus which describes the stiffness and Poisson's ratio which describes how a solid material tends to expand or contract in a direction perpendicular to any longitudinal stress the soft body is under.

Plasticity allows deformations to remain when external forces are removed. This is achieved by storing the plastic strain caused by the internal elastic strain on the elements that form the soft body. It is this plastic strain that retains the deformation. As it is a stored value within the model, it can be removed at any time. The plasticity is controlled by three parameters, $plastic_{yield}$, the threshold amount of elastic strain that when exceeded, results in the storing of plastic strain. $plastic_{creep}$, the rate at which plastic strain is applied and $plastic_{max}$, the maximum amount of elastic strain that is permitted.

Fracturing allows tearing of a soft body when the internal stress of elements within the body exceeds a fracture threshold. It was decided that for these initial prototypes, fracturing would not be used to avoid the complexity of implementing a character that consisted of multiple separate parts.

The individual parameters from each property were combined to describe what was referred to as a physics material. The implementation of the FEM model was amended to allow the different elements within the model to have their own material applied to them. This allowed for different parts of the model to have different deformation characteristics to one another and produces a much more flexible simulation model.

Prototype 1

To explore the initial challenges in creating deformable characters and the subsequent game mechanics, a ball-shaped character consisting of a single physics material was created.

Character Physics Model

A tetrahedral mesh defines the character shape in the soft body simulation and mesh coupling is used to deform the high-resolution graphical mesh seen in-game.

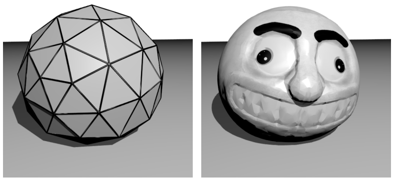


Figure 1: Ball-shaped Character.

Table 1: Material settings for spherical character.

Tetrahedra	118
Surface triangles	118
Density $[kg/m^3]$	1000
Young's modulus $[N/m^2]$	100000
Poisson's ratio	0.99

The refinement of the tetrahedral mesh was chosen so that it was high enough to produce a level of deformation that looked visually plausible when the character collided with objects in the environment, but not cause a significant drop in frame rate.

Control Method for Basic Movement

The ball-shaped character's overall shape is virtually identical to a comparable spherical rigid body. It was therefore decided to mimic the movement mechanic seen in Marble Madness-style games where the direction the character moves in is controlled by 2D axis user input (forward and back, left and right). In order to achieve this, a frame of reference is needed for the soft body. In a rigid body, this is simply the position and rotation of the body. As a soft body consists of a collection of interconnected elements that can be moving in different directions as a result of varying forces being applied across the body, there is no explicit position and rotation for the body as a whole. To achieve this, a fixed node at the centre of the soft body was added during the generation of the tetrahedral mesh.

Because the motion of the body is controlled by applying a force to the body in the desired direction of travel irrespective of the overall orientation of the ball, there is no need to explicitly calculate the overall orientation of the soft body ball. To determine the current direction of travel for the ball, the position of the origin node is sampled over time to calculate its velocity. This velocity is a useful attribute for the player model to expose. For example, it can be used to implement a camera the orients itself with the direction the player is moving in. With the addition of a frame of reference, any mechanic (e.g. jumping) that is straightforward to implement on a rigid body can be transferred to the soft body simply by applying the appropriate uniform force to all elements within the soft body.

Formulating Behaviours and Mechanics

"Softening" The Character

By decreasing the stiffness of the soft body, the increased amount of deformation enables the character to be squeeze through gaps in the environment they would not normally be able to pass if they were implemented as rigid bodies.

This can be achieved simply by lowering Young's modulus in the physics material over time to create a more malleable character.

It's worth noting that with the FEM simulation model chosen here, changing Young's modulus in the physics material requires parts of the underlying simulation model to be reinitialised. The amount of process time this takes is proportional to the number of elements present in the tetrahedral mesh and is, therefore, another place the refinement of the mesh can have an impact on performance.

The player can now consider how this "softness" can be used to traverse the game environment. Existing games, like Gish and Cats are Liquid, enable this ability through user control, but it doesn't have to be limited to this type of use. Environmental objects can be used to change the state of the player providing a range of ways to interact with this mechanic. For example, simulated heat can be used to temporarily melt the character and as it moves from the heat source, it can return its default stiffness over time.

Stretching The Character

In order to explore how stretching can be used as a mechanic, positional constraints were used to temporarily hold a part of the character in place. In essence, a sticky surface was created. Without these constraints, any external forces used to stretch a part of the character would result in dragging the entire shape. The initial concept was to apply stretch forces to the unconstrained side of the character in a direction perpendicular to the sticky surface and see what effect this could have in terms of gameplay. For this character, the overall shape and topology of the underlying tetrahedral mesh prohibited plausible deformations taking place. This was due to the relative coarseness of the mesh and the rolling nature of the character movement. When the character rolled into the sticky surface, depending on its orientation, a varying number of nodes in the tetrahedral mesh would become constrained to the block and in the case where this may have been a low number of contact points, the stretching would cause unwanted irregular spikes in the deformation. This could be potentially be counteracted by increasing the refinement of the mesh, but this then had a significant detrimental on performance. This approach was therefore abandoned.

Prototype 2

Here a cylindrically shaped character was used for a number of reasons. The shape allows more potential to create stretch based mechanics due to its shape and topology. Its shape is very similar to the capsule shape that is commonly used to represent characters in physics simulations for games. This would allow the creation of deformable characters whose basic movement is implemented in a way familiar to the player.

Character Physics Model

To create a cylinder-based character that exhibits elasticity in its default state, multiple physics materials were used to create the tetrahedral model. The bottom section was created with a physics material with a higher density and stiffness to the middle and top sections so the character wouldnt topple easily. The stiffness of the remaining sections was set to create a natural sway in the character as it moved to emphasise it's deformable nature.

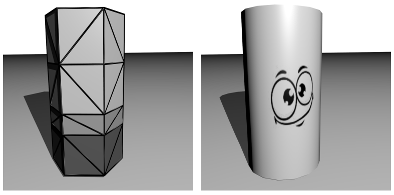


Figure 2: Cylinder-shaped Character.

Table 2: Material settings for cylindrical character.

Tetrahedra	104
Surface triangles	72
Density $[kg/m^3]$	750
Poisson's ratio	0.99
Bottom: Young's modulus $[N/m^2]$	500000
Mid: Young's modulus $[N/m^2]$	300000
Top: Young's modulus $[N/m^2]$	10000

Control Method for Basic Movement

Character controllers in game engines such Unity (Uni 2008) allow for more precise control by setting the desired linear and angular velocity of the controller rather than through the application of forces when momentum based acceleration/deceleration is not wanted. To duplicate this in the deformable character, a reference coordinate frame is needed that provides both position and orientation. A fixed node was added in the tetrahedral mesh generation in the bottom region to define the origin of the character, similar to the ball-shaped character. As orientation is also needed, additional nodes were added to define an orthogonal coordinate frame within the bottom region.

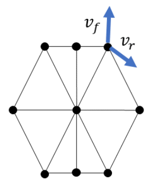


Figure 3: Linear forces acting on core nodes.

To create the controllable movement in the character the bottom region was treated as a rigid core that would essentially pull the rest of the character along as it moved. To achieve this, the linear velocity of each node in the core region was set after the soft body simulation was run each frame.

The total linear velocity of each node within the core region is defined as:

$$v_{total} = v_f + v_r \tag{1}$$

where v_f is the desired linear velocity and v_r is the linear velocity resulting from the desired angular velocity, which is subsequently defined as:

$$v_r = \omega \times r \tag{2}$$

where ω is the desired angular velocity and r is the position of the core node relative to the core origin node. This essentially created a controllable rigid lower region coupled with soft deformable upper regions but without the need to couple a separate rigid body simulation with the soft body simulation model.

Rigid body based mechanics like jumping can now be created to the cylinder-based character by applying uniform forces to all nodes in the core region or through the manipulation of the desired linear and rotational velocities of the character.

Formulating Behaviours and Mechanics

Stretching Up

A stretch mechanic was created to pull the character into a longer shape. This was achieved by constraining the top and bottom of the character with anchors and then raising the top anchor over a period.

During this time, the plasticity parameters were altered to allow the deformation to remain once the anchor was moved. At this point, the plastic creep would be reset to zero to prevent further deformations to remain permanent. This was required so the default elastic movement in the character would return. In this elongated state, the character could now reach items or locations in the level which were too high without the change in shape.

Table 3: Plasticity settings for stretching

.

$plastic_{yield}$	0.02
$plastic_{creep}$	20
$plastic_{max}$	0.613

Squashing Down

A squash mechanic was created using a similar method to the stretch mechanic, but with the top anchor moving downward. This allows the character to shrink down to a smaller size and fit under obstacles. This shows how the deformation can be used to achieve something similar to existing crouch mechanics.

Table 4: Plasticity settings for squashing

.

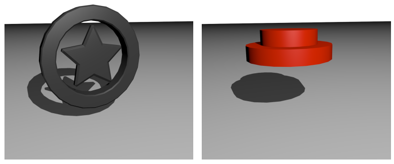
$plastic_{yield}$	0.02
$plastic_{creep}$	20
$plastic_{max}$	0.35

Swinging

To create a more novel means of traversing gaps within a game level, a swing mechanic was created. The top surface of the character was constrained to an anchor within the level. The core based character movement was disabled at this point and a linear acceleration was applied based on the forward user input. This linear acceleration, combined with the centre of mass in the bottom region of the character and the elasticity of the upper regions created a natural swinging motion in the character. The player is then able to build up the swing momentum by alternating the forward user input in time with the swing and cover a greater distance when released from the anchor point. Additionally, when the character is anchored at the top it can also be pulled down to stretch its shape. This is achieved by lowering Youngs modulus to 9000 in the physics material for the top region and apply a downward force to the origin node in the core region. This allows for the character to reach items or locations in the level which were too low to reach without the change in shape.

RESULTS

Figure 4 shows objects used in the prototypes to represent the goal to reach for and the anchor used to constrain the character when creating squash and stretch. Figures 5 - 8 show the visual deformation of the character and demonstrates how the implemented mechanics allow the character to reach an item that would be unattainable if the character was unable to deform.



(a) Goal object (b) Anchor object

Figure 4: Prototype objects.

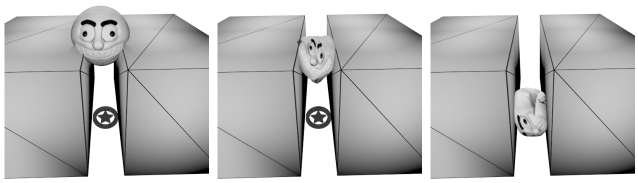


Figure 5: Softening mechanic.

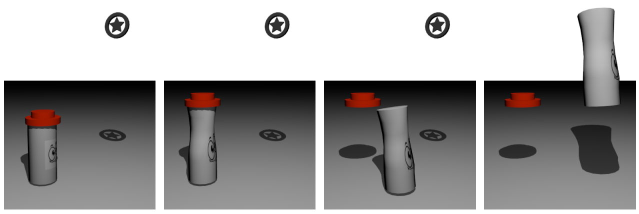


Figure 6: Stretching mechanic.

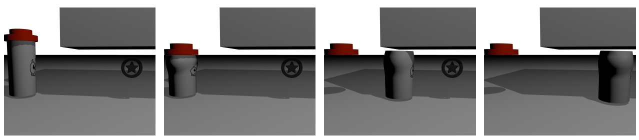


Figure 7: Squashing mechanic.

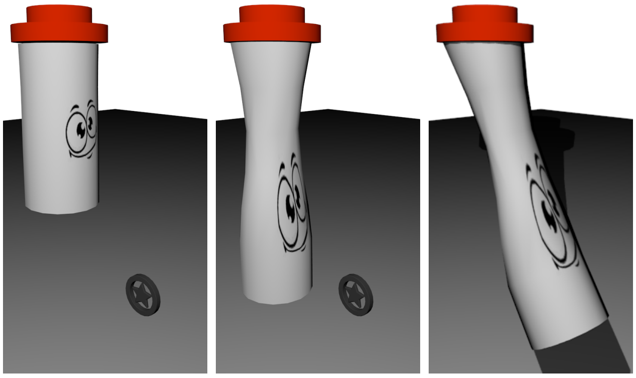


Figure 8: Swinging mechanic.

DISCUSSION

Qualitative Evaluation

The prototype with the ball-shaped character was successful as a proof of concept in terms of testing the pipeline and creating a deformable character like those seen in existing titles. It also demonstrated that beyond the ability to change the overall elastic characteristics of the body to enable the character to squeeze through gaps, limitations based on the mesh topology and the rolling nature of the movement made it difficult to implement other deformation based behaviours. This may indicate why existing games with this style of deformable character predominately use this mechanic and nothing beyond it.

The formation of the tetrahedral mesh topology in the cylinder-based character allows for a more varied use of the characteristics of the simulation model. Constraining specific nodes to anchors that were in line with the stretch direction created deformation in keeping with the overall shape of the character when compared to the deformation problems seen in the ball-shaped character.

The implementation of the rigid core region in the cylinder-shaped character allows for this style of deformable character to use similar methods of motion control to those found using rigid body-based character controllers. By avoiding the coupling of a rigid body to create this core, allows for this region to interchange between rigid and soft body behaviour when used to implement different mechanics.
The elastic and plastic properties of the FEM simulation model enabled the creation of novel deformation-based game mechanics in this relatively small set of experiments. The limitations of the implementation lead to restrictions when balancing the refinement of the generated tetrahedral mesh with runtime performance. Large node displacements did result in invalid model configurations for the soft body simulation. This was particularly evident in the creation of the squashing mechanic where only a relatively small downward displacement was possible before the model folded in on itself. This could potentially be solved by implementing collision response between internal nodes in the tetrahedral mesh. In addition, changing material parameters often required an expensive reinitialise step. While the properties of this simulation are appropriate for creating deformation-based mechanics a more stable simulation model should be considered.

Future Work

The shapes used in the experiments are basic primitive shapes, but the techniques used here does allow for the creation of more complex character shapes. The addition of limbs, for example, can create characters representing creatures or humanoids. This does, however, mean that implementing the basic movement of the character may become more complex. References showed how skeleton animation can be combined with soft body deformation to create deformable characters. This work could be extended to create stretchable bones to allow external forces to alter the overall skeletal shape rather than just the deformation of the soft body skin. This would allow for more creative control in the movement of the character that is augmented by the elastic properties of the soft body rather than relying on procedurally generated animation by the soft body simulation alone.
Using alternate simulation models can not only help eliminate the stability issues but with a different set of physical properties can provide the potential for a more varied set of gameplay mechanics. Macklin et al. (2014) demonstrates soft bodies that interact with rigid bodies, fluid and gases in the same simulation model that could lead to some interesting mechanics where a character could change between these states.

CONCLUSIONS

This paper presents a set of game mechanics that are made possible by the use of soft body simulation to create deformable game characters. The findings showed that even when a somewhat limited set of mechanics, the gameplay prototypes produced demonstrate the potential for more varied and unique gameplay experiences for the player. The deformation behaviour of the character is fully dynamic and controllable. The use of volume regions within the simulation model allows for parts of the characters to deform differently. This was used to create a rigid core within the simulation model that controls the overall motion of the character without the need to couple with a rigid body system. This facilitates a method of control similar to rigid body character controllers that allow players to direct the overall motion of the characters in an intuitive manner while still being able to use the deformation properties of the character as a means to interact with the environment to achieve goals within a game. The topology of the tetrahedral mesh, the configurable properties of the underlying simulation model and the extent these properties can be affected in real-time directly impacts the range of gameplay mechanics possible.

In the current FEM-based implementation, plausible deformations are shown when the character is subjected to external forces, but the integrity of the model can be lost if the displacement of the internal elements of the simulation model is too large which negatively impacts the player experience. This can be overcome by exploring more robust simulation models that have different characteristics and hence lead to the creation of more deformation based game mechanics.

REFERENCES

Atari Games, 1984. *Marble Madness.* Atari Games [Vide Game].

Cryptic Sea, 2003. *Gish.* Chronic Logic [Video Game].

Last Quarter Studios, 2015. *Cats Are Liquid.* Last Quarter Studios [Video Game].

Liu L.; Yin K.; Wang B.; and Guo B., 2013. *Simulation and Control of Skeleton-Driven Soft Body Characters. ACM Transactions on Graphics*, 32, no. 6, 1–8. ISSN 07300301. doi:10.1145/2508363.2508427.

Macklin M.; Müller M.; Chentanez N.; and Kim T.Y., 2014. *Unified Particle Physics for Real-Time Applications. ACM Transactions on Graphics*, 33, no. 4, 1–12. ISSN 07300301. doi:10.1145/2601097.2601152.

Müller M. and Gross M., 2004. *Interactive Virtual Materials.* In *Proceedings of Graphics Interface 2004.* Canadian Human-Computer Communications Society, 239–246.

Müller M.; Heidelberger B.; Hennix M.; and Ratcliff J., 2007. *Position Based Dynamics. Journal of Visual Communication and Image Representation*, 18, no. 2, 109–118. ISSN 10473203. doi:10.1016/j.jvcir.2007.01.005.

Parker E.G. and O'Brien J.F., 2009. *Real-Time Deformation and Fracture in a Game Environment.* In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* ACM, 165–175.

Ready At Dawn, 2017. *Deformers.* GameTrust [Video Game].

Rovio Entertainment, 2009. *Angry Birds.* Chillingo [Video Game].

Second Order, 2017. *Claybook.* Second Order [Video Game].

Tan J.; Turk G.; and Liu C.K., 2012. *Soft Body Locomotion. ACM Transactions on Graphics (TOG)*, 31, no. 4, 26.

WEB REFERENCES

2008. *Unity.* https://unity3d.com. [Accessed June 2018].

BIOGRAPHY

Grant Clarke is a lecturer in Computer Games Development and programme leader of the Computer Games Technology degree course at Abertay University.