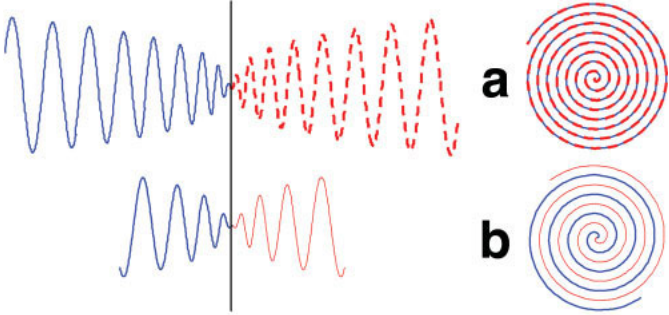


Game Programming: Exercise 10: Animations

Learning objectives	<p>Learning objectives</p> <ul style="list-style-type: none">• Implement Cel animations / sprite animations based on physics simulation• Implementing easing animation as an improvement to simple linear interpolation.• Implement spline curves for advanced movement• (optional) Implement a simple animation script using Lua <p>Handing in: Individual hand in. Create a zip-version of source-files, header-files and resources (CMakeFileLists.txt, .json and png). We will build your project using CMake, so make sure it works before hand-in (Note: If you make sure to keep all files in the same directory it should work without you needing to change the CMakeFileLists.txt).</p> <p>Controls:</p> <p>Left / Right - controls the character</p> <p>Space - Jump</p> <p>D - debug menu and visualize fly movement path</p> <p>Z - zoom</p>
10-1	<p>Cel animations</p> <p>Implement cel animations of the main character, by implementing the function <code>CharacterController::updateSprite()</code></p> <ul style="list-style-type: none">• If the character has zero velocity, then the standing sprite must be used.• If the character is grounded then the walk1 and walk2 sprites must be used. Choose an appropriate animation speed based on the horizontal velocity.• If the character jumps, use the sprite flyUp, fly and flyDown for the three phases of the jump.• The sprite must be flipped around the x-axis if the character moves towards the left.
10-2	<p>Platform moving using easing functions</p> <ul style="list-style-type: none">• Choose a non-linear easing function to move the platforms in <code>MovingPlatformComponent::update()</code>
10-3	<p>Bird Movement using Quadratic Bézier curves</p> <ul style="list-style-type: none">• The bird (or fly?) moves along a sequence of predefined points. Intermediate positions are defined by linearly interpolating between pairs of points using <code>glm::mix()</code> in the <code>BirdMovementComponent::computePositionAtTime()</code> method.• Replace the linear movement trajectory with Quadratic Bézier curves. Note that you also need to change <code>BirdMovementComponent::getNumberOfSegments()</code> to return the correct number of segments.

Bonus 1	<p>(Optional) SpiralBird Movement using ScriptComponent</p> <p>The objective of this exercise is to understand how to effectively use the sol library to create Lua bindings. To solve this, you will need to read and understand parts of sol that we did not cover during the lecture.</p> <ul style="list-style-type: none"> • Uncomment the block comment in PlatformerGame::initLevel() line 129-159 • In ScriptComponent::init() extend GameObject Lua definition to include the functions needed to run the Lua script from PlatformerGame::initLevel(). <ul style="list-style-type: none"> ◦ Hint: look at the errors in the debug console, when the Lua is executed. • Once you have the second bird flying in circles, change the Lua script in PlatformerGame::initLevel(), to update radius continuously so that the bird flies in a spiral with radius between [10, 100] • Example of radial movement over time (i.e. 100->10->100) 
Bonus 2	<p>(Optional) Scripting system architecture</p> <p>The objective of this exercise is to design a scripting system from the ground up, following some generic requirements like you could get in a small game development team. There is no “correct” solution to this exercise, but in two weeks time we will have a discussion during the lab with the people trying this exercise to compare solutions and reason about trade-offs.</p> <p>You can go ahead and implement your proposed solution, but pseudocode, UMLs and other high level descriptions are encouraged.</p> <p>Requirements of our scripting system:</p> <ul style="list-style-type: none"> • As a game designer, I want to edit scripts without recompiling the game engine • As a game designer, I want to ignore architecture and focus on game logic • As a game designer, I want to a GUI that guides me in the scripting process • As a game designer, I want to have more than one GameObject available when writing scripts • As an engine programmer, I want to be sure that a script can’t crash the engine, no matter what • As an engine programmer, I want other systems to be completely unaware of the existence of the scripting system • As an engine programmer, I want to be able to do everything that a script can do in C++ • As an engine programmer, I want the bindings to be generated automatically • As an engine programmer, I want to avoid updating the scripting system when other systems change • As an engine programmer, I want scripts to be as fast as possible <p>Hint: you are on a 2 week deadline, with many other tasks. What of this wall of text is reasonable in the allotted time? What is a “need” and what is a “want”?</p>