Здесь будет титульный лист. TODO

РЕФЕРАТ

Здесь будет реферат. TODO

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Здесь будет введение. TODO

1 Теоретические сведения

1.1 Физический движок TODO

Физический движок – программное обеспечение, предназначенное для приближённой симуляции физический систем реального мира в виртуальном, например динамику твёрдого тела (включая обнаружение столкновений и обработку ударов).

Иными словами, система физического моделирования (т.е. моделирующая физику – в данном контексте симуляция синонимична моделированию) и есть физический движок.

Традиционно, физические движки делятся на 2 типа: игровые и научные. Отличительная черта первых – возможность работы в режиме реального времени, то есть воспроизводить физические процессы в игре с той же самой скоростью, в которой они происходят в реальном мире. Отличительная черта вторых – важна точность вычислений, скорость вычислений не играет существенной роли.

Кроме того, в игровых движках зачастую достаточно симулировать только текущее состояние виртуального мира без сохранения результата, а в научных движках необходимо сохранять все смоделированные в виртуальном мире взаимодействия для последующего анализа.

Одна из задач данной работы – создать движок, имеющий свойства и научного, и игрового – чтобы присутствовала возможность и моделирования в реальном времени, и эффективной «перемотки».

1.2 Апостериорный и априорный подход

В философском смысле, апостериори означает знание, полученное из опыта; а априори означает знание, полученное до опыта и независимо от него [1, с. 105-106].

В аналогичном порядке и делят подходы к реализации физических моделей: определение столкновения апостериори и априорный подход (обнаружение до происхождения столкновения).

В апостериорном случае физическое моделирование продвигается на небольшой шаг, а затем проверяется, пересекаются ли какие-либо объекты или,

по видимости, считаются пересекающимися. На каждом шаге моделирования создается список всех пересекающихся тел, а положения и траектории этих объектов «фиксируются» для учета столкновения. Этот метод называется апостериорным, поскольку он, как правило, не учитывает фактический момент столкновения, а фиксирует столкновение только после того, как оно произошло.

В априорных методах существует алгоритм обнаружения столкновений, который сможет очень точно предсказать траектории физических тел. Моменты столкновения рассчитываются с высокой точностью, при этом физические тела никогда фактически не пересекаются. Это называется априорным, потому что алгоритм обнаружения столкновений вычисляет моменты столкновения до того, как он обновит конфигурацию физических тел.

Основные преимущества апостериорных методов заключаются в том, алгоритму обнаружения столкновений не нужно знать о множестве физических переменных; алгоритму подается простой список физических тел, и программа возвращает список пересекающихся тел. Алгоритму обнаружения столкновений не нужно понимать трение, упругие столкновения или, что еще хуже, неупругие столкновения и деформируемые тела. Кроме того, апостериорные алгоритмы в действительности на одно измерение проще, чем априорные алгоритмы. Априорный алгоритм должен иметь дело с переменной времени, которая отсутствует в апостериорной задаче.

С другой стороны, у апостериорных алгоритмы существует шаг «исправления», где пересечения (которые физически не являются правильными) должны быть исправлены и это может создать проблему. Более того, если дискретный шаг слишком велик, столкновение может остаться незамеченным, в результате чего объект пройдет сквозь другой, если он достаточно быстрый или маленький.

Преимуществами априорных алгоритмов являются повышенная точность и стабильность. Физическое моделирование трудно отделить от алгоритма обнаружения столкновений. Однако во всех случаях, кроме самых простых, проблема определения времени когда два тела столкнутся заранее (учитывая некоторые начальные данные), не имеет аналитического решения – обычно требуется численный поиск корней (что и описано в 1.6).

Обычно для игровых физических движков используется апостериорный подход потому что не важна точность, а дискретное моделирование хорошо ложиться на расчёты в реальном времени. Однако данная работа призвана исследовать априорный подход и при этом создать движок, способный на симуляцию в реальном времени.

Как указано ранее, априорный подход должен учитывать моделируемые законы физики. Но при этом много что трудно учесть, поэтому симулируемая модель будет упрощённая, без учёта деформации, вращения, формы. Подвижные объекты – только тела идеально круглой формы, движение равноускоренное (равнозамедленное, но в дальнейшем везде называется равноускоренным), и т.д. Подробнее модель описана под пунктом 1.3.

1.3 Описание модели

Тело. Абсолютно твёрдое тело в форме круга равномерной плотности (центр масс в центре круга) обладающее массой (m), коэффициентом трения (μ), радиусом (r), начальной скоростью ($\vec{v_0}$), положением (координаты x и y или радиус-вектор $\vec{r}$). На тело действует сила трения ($F_{\text{тр.}}$).

Точка. Неподвижная точка в пространстве, определена через координаты.

Линия. Неподвижная прямая линия в пространстве, может быть ограничена точкой с двух или одной сторон образуя отрезок или луч соответственно. Определена через общее уравнение прямой.

Сцена. Множество тел, линий, точек и постоянных (например, ускорение свободного падения).

Обновлённая сцена – сцена, в которой обновлены параметры тел, линий, точек или постоянных.

Сцена через время Δt – обновлённая сцена, в которой все тела обновлены так, что новая начальная скорость равна скорости в этот момент времени (1).

$$\vec{v_{0_{\text{new}}}} = \vec{v}(\Delta t) \tag{1}$$

где $\vec{v_{0_{\text{new}}}}$ – новая начальная скорость;

$\vec{v}(\Delta t)$ – старая скорость в момент времени Δt.

Модель. Множество пар (t, S), где t – момент времени, а S – сцена. Иными словами, модель представляет собой цепочку сцен, для каждой из которой указан момент времени.

Сцена в момент времени t_1 – такая сцена S_0 через время $t_1 - t_0$, где пара (t_0, S_0) является членом модели, при этом соблюдается (2).

$$\forall (t, S) \in M \;\; (t \leqslant t_0 \vee t > t_1) \tag{2}$$

где M – модель;

t_0 – время, выбранное для получения модели в момент времени t_1;

Иными словами, для того чтобы получить сцену в момент времени, надо из цепочки сцен найти такую, у которой время будет максимально, но при этом меньше требуемого момента времени и получить сцену через разность требуемого и найденного времени по формуле (1).

Столкновение. Так как тела не могут пересекаться, и при этом передвигаются, могут происходить столкновения. Так же тела не могут пересекаться с точками и линиями. Т.е. тела могут сталкиваться с телами, или линиями, или точками. Уравнение столкновения тела с телом (3)-через радиус-вектор, (4)-через координаты.

$$|\vec{r_1}(t) - \vec{r_2}(t)| = r_1 + r_2 \tag{3}$$

где $\vec{r_1}(t)$ – радиус-вектор положения первого тела;

$\vec{r_2}(t)$ – радиус-вектор положения второго тела;

r_1 – радиус первого тела;

r_2 – радиус второго тела.

$$\sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2} = r_1 + r_2 \tag{4}$$

где $x_1(t)$ – координата положения первого тела по оси X;

$y_1(t)$ – координата положения первого тела по оси Y;

$x_2(t)$ – координата положения второго тела по оси X;

$y_2(t)$ – координата положения второго тела по оси Y.

Эти уравнения получены исходя из того что разность векторов является вектором из центра одного тела в центр другого [3, с. 39]. И тогда, если его длина равна сумме радиусов этих тел, значит тела столкнулись.

Подобным образом можно определить уравнение столкновения тела с точкой: (5)-через радиус-вектор, (6)-через координаты.

$$|\vec{r}(t) - \vec{p}| = r \tag{5}$$

где $\vec{r}(t)$ – радиус-вектор положения тела;

$\vec{p}$ – радиус-вектор точки;

r – радиус тела.

$$\sqrt{(x(t) - p_x)^2 + (y(t) - p_y)^2} = r \tag{6}$$

где $x(t)$ – координата положения тела по оси X;

$y(t)$ – координата положения тела по оси Y;

p_x – координата точки по оси X;

p_y – координата точки по оси Y.

С обнаружение столкновением тела и линии ситуация несколько иная, надо воспользоваться формулой расстояний от точки до прямой (7) [4, с. 452] и тогда получится (8).

$$\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}} \tag{7}$$

где A, B, C – коэффициенты общего уравнения прямой;

x, y – координаты точки.

$$\frac{|Ax(t) + By(t) + C|}{\sqrt{A^2 + B^2}} = r \tag{8}$$

где $x(t)$, $y(t)$ – координаты тела в момент времени t.

В дальнейшем будут определены формулы для нахождения положения тела, а именно: (17), (18).

1.4 Определение формул скорости и траектории тела

Далее, под моментом времени t будет подразумеваться время относительно сцены, а не модели.

Как указано выше, у тела есть начальная скорость и на него действует сила трения. По второму закону Ньютона [2, с. 114], у тела есть ускорение так как на него действует сила. Такое движение называется равноускоренным.

Скорость при равноускоренном движении определяется формулой (9)[2, с. 96].

$$\vec{v}(t) = \vec{v_0} + \vec{a}t \tag{9}$$

где $\vec{v}(t)$ – вектор скорости тела в момент времени t;

$\vec{v_0}$ – вектор начальной скорости тела;

$\vec{a}$ – вектор ускорения тела;

t – момент времени.

При этом вектор ускорения сонаправлен вектору силы (по второму закону Ньютона, (10)).

$$\vec{a} = \frac{\vec{F}}{m} \tag{10}$$

где $\vec{a}$ – вектор ускорения тела;

$\vec{F}$ – вектор силы действующей на тело;

m – масса тела.

Но вектор силы трения $\vec{F_{\text{тр.}}}$ противонаправлен вектору скорости тела [2, с. 21]. Поэтому, и вектор ускорения тела будет противонаправлен вектору скорости тела.

При этом вектор скорости тела $\vec{v}(t)$ должен быть сонаправлен вектору $\vec{v_0}$ потому что тело не может поменять направление движения при воздействии силы трения. Для того чтобы выяснить, при каких t сонаправленность векторов $\vec{v}(t)$ и $\vec{v_0}$ в уравнении (9) соблюдается, достаточно увидеть, что длина вектора $\vec{v_0}$ должна быть больше длине вектора $\vec{a}t$ и получить неравенство для t (11).

$$t < \frac{|\vec{v_0}|}{|\vec{a}|} \tag{11}$$

А для остальных t, $\vec{v}(t)$ следует принять нулю. Тогда скорость выражается через (12):

$$\vec{v}(t) = \begin{cases} \vec{v_0} + \vec{a}t, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ 0, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{12}$$

Проекции на ось абсцисс (13) и ординат (14):

$$v_x(t) = \begin{cases} v_{0_x} + a_x t, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ 0, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{13}$$

где $v_x(t)$ – проекция вектора скорости тела $\vec{v}(t)$ в момент времени t на ось X;
v_{0_x} – проекция вектора начальной скорости тела $\vec{v_0}$ на ось X;
a_x – проекция вектора ускорения тела $\vec{a}$ на ось X.

$$v_y(t) = \begin{cases} v_{0_y} + a_y t, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ 0, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{14}$$

где $v_y(t)$ – проекция вектора скорости тела $\vec{v}(t)$ в момент времени t на ось Y;
v_{0_y} – проекция вектора начальной скорости тела $\vec{v_0}$ на ось Y;
a_y – проекция вектора ускорения тела $\vec{a}$ на ось Y.

Теперь найдём формулу для траектории движения тела. Формуле, соответствующей (9), только для траектории, соответствует (15):

$$\vec{r}(t) = \vec{r_0} + \vec{v_0}t + \frac{\vec{a}t^2}{2} \tag{15}$$

где $\vec{r}(t)$ – радиус-вектор положения тела в момент времени t;
$\vec{r_0}$ – радиус-вектор начального положения тела.

Исходя из (12), уравнение для траектории с учётом того, что вектор скорости должен быть противонаправлен вектору ускорения, будет (16):

$$\vec{r}(t) = \begin{cases} \vec{r_0} + \vec{v_0}t + \frac{\vec{a}t^2}{2}, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ \vec{r_0}, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{16}$$

Соответствующие проекции на ось абсцисс (17) и ординат (18):

$$x(t) = \begin{cases} x_0 + v_{0_x} t + \frac{a_x t^2}{2}, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ x_0, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{17}$$

где $x(t)$ – координата положения тела $\vec{r}(t)$ в момент времени t на ось X;
x_0 – координата начального положения тела $\vec{v_0}$ на ось X.

$$y(t) = \begin{cases} y_0 + v_{0_y} t + \frac{a_y t^2}{2}, & 0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}, \\ y_0, & t \geqslant \frac{|\vec{v_0}|}{|\vec{a}|}. \end{cases} \tag{18}$$

где $y(t)$ – координата положения тела $\vec{r}(t)$ в момент времени t на ось Y;
y_0 – координата начального положения тела $\vec{v_0}$ на ось Y.

Формулы (17) и (18) являются ключевыми в этой работе. TODO

1.5 Определение уравнений для обнаружения столкновений

Подставив формулы (17) и (18) в уравнения (4), (6), (8) можно получить алгебраические уравнения от t четвёртой степени.

Например, для случая, когда $0 \leqslant t < \frac{|\vec{v_0}|}{|\vec{a}|}$ частичный вывод уравнения времени столкновения тела с телом будет таким (19):

$$\begin{gathered} \sqrt{(x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2} = r_1 + r_2 \\ (x_1(t) - x_2(t))^2 + (y_1(t) - y_2(t))^2 = (r_1 + r_2)^2 \\ (x_{0_1} + v_{0_{x_1}} t + \frac{a_{x_1} t^2}{2})^2 - (x_{0_2} + v_{0_{x_2}} t + \frac{a_{x_2} t^2}{2})^2 + \\ + (y_{0_1} + v_{0_{y_1}} t + \frac{a_{y_1} t^2}{2})^2 - (y_{0_2} + v_{0_{y_2}} t + \frac{a_{y_2} t^2}{2})^2 = (r_1 + r_2)^2 \end{gathered} \tag{19}$$

где $x_1(t)$ – координата первого тела в момент времени t по оси X;
$x_2(t)$ – координата второго тела в момент времени t по оси X;
$y_1(t)$ – координата первого тела в момент времени t по оси Y;
$y_2(t)$ – координата второго тела в момент времени t по оси Y;
x_{0_1} – начальная координата первого тела по оси X;
x_{0_2} – начальная координата второго тела по оси X;
y_{0_1} – начальная координата первого тела по оси Y;

y_{0_2} – начальная координата второго тела по оси Y;
$v_{0_{x_1}}$ – проекция вектора начальной скорости первого тела на ось X;
$v_{0_{y_1}}$ – проекция вектора начальной скорости первого тела на ось Y;
$v_{0_{x_2}}$ – проекция вектора начальной скорости второго тела на ось X;
$v_{0_{y_2}}$ – проекция вектора начальной скорости второго тела на ось Y;
a_{x_1} – проекция вектора ускорения первого тела на ось X;
a_{y_1} – проекция вектора ускорения первого тела на ось Y;
a_{x_2} – проекция вектора ускорения второго тела на ось X;
a_{y_2} – проекция вектора ускорения второго тела на ось Y;
r_1 – радиус первого тела;
r_2 – радиус второго тела.

Из (19) видно, что дальнейшее раскрытие квадратов, сокращение, вынос t за скобки, перенос из правой части в левую приведёт к тому, что уравнение примет вид (20).

$$P(t) = 0 \tag{20}$$

где $P(t)$ – многочлен от t.

Значит, уравнение алгебраическое одной переменной. Теперь найдём степень уравнения (т.е. максимальную степень одночлена). Возьмём одночлен $\frac{a_{x_1}t^2}{2}$, который является частью многочлена, который возведён в квадрат. Следовательно, в результирующем многочлене будет присутствовать одночлен (21).

$$(\frac{a_{x_1}t^2}{2})^2 = \frac{{a_{x_1}}^2t^4}{4} \tag{21}$$

Как видно, в результирующем многочлене t возведён в 4 степень. При этом в результирующем уравнении этот одночлен не сократится потому что у других подобных одночленов другие коэффициенты. Значит, в результирующем уравнении будет присутствовать одночлен степени 4.

Так как для каждого случая подстановки формул (17) и (18) в уравнения (4), (6), (8) придётся производить 8 таких выводов (для каждого из 2 случаев из (17) следует рассмотреть 2 случая из (18) при подстановке в (4) и по 2 случая при подстановке в (6) и (8)), проще составить программный алгоритм (который

описан позднее в TODO), который применяя методы компьютерной алгебры, сможет строить выражения и вычислять их значения, подставляя переменные.

1.6 Решение алгебраических уравнений четвёртой степени

Так как для работы создаваемого физического движка не требуется обнаруживать время столкновения без погрешности и корни могут быть только действительными, можно воспользоваться численными методами. К тому же, аналитические методы решения алгебраических уравнений четвёртой степени обладают громоздкостью. И при этом, предложенный далее метод численного решения алгебраических уравнений позволяет решать уравнения любой степени, что позволит теоретически внести в модель и реализацию, без существенного переписывания кода, например разноускоренное движение (т.е. величину рывок, от которой зависит ускорение) и тогда уравнение, подобное (19), получится уже пятой степени, что не имеет решений в радикалах вовсе по теореме Абеля о неразрешимости уравнений в радикалах [5, с. 112].

1.6.1 Метод бисекции

Метод бисекции или метод деления отрезка пополам – простейший численный метод для решения нелинейных уравнений вида $f(x) = 0$, является частным случаем бинарного поиска [6].

Метод бисекции позволяет найти корень функции на отрезке $[x_l, x_r]$, если $sign(f(x_l)) \neq sign(f(x_r))$. При этом, если функция $f(x)$ не монотонна, она имеет несколько корней и метод найдёт лишь один корень [6]. Поэтому, метод бисекции не подходит для общего случая поиска решений алгебраических уравнений без предварительной подготовки, которая описана в пункте 1.6.2.

При этом можно обобщить этот метод для поиска корней на промежутках вида $(-\infty, x_r]$, $[x_l, +\infty)$. Недостающую границу поиска можно подобрать кратным (например, с каждой итерацией увеличивать изменение в 2 раза) увеличением (в случае бесконечности справа) или уменьшением (в случае бесконечности слева) относительно известной границы до того момента, пока знак функции на левой и правой границе не станет разным. Но при этом надо учесть, что корня на промежутке может не быть в случае когда при

изменении (увеличении или уменьшении, как указано ранее) неизвестной границы, значение функции не становится ближе к нулю.

Описание метода: на каждой итерации берётся середина отрезка (22).

$$x_m = \frac{x_l + x_r}{2} \tag{22}$$

где x_m – середина отрезка $[x_l, x_r]$.

Теперь следует вычислить значение функции в середине отрезка и если (23), то корень (равный x_m) найден; иначе следует разбить отрезок $[x_l, x_r]$ на два отрезка $[x_l, x_m]$ и $[x_m, x_r]$.

$$|f(x_m)| \leqslant \varepsilon \tag{23}$$

где ε – заданная точность по оси Y.

Далее, если x_l и x_m имеют разный знак, провести итерацию с отрезком $[x_l, x_m]$. Иначе, разный знак должен быть у x_m и x_r, значит провести итерацию с отрезком $[x_m, x_r]$.

1.6.2 Описание реализованного метода

TODO

2 Проектирование

Здесь будет вторая глава. TODO

2.1 Язык программирования OCaml

TODO

2.2 Обзор экосистемы языка OCaml

TODO

2.2.1 Компиляторы OCaml в JavaScript

js_of_ocaml, rescript, ocamljs, (wasm TODO) TODO

2.2.2 Стандартные библиотеки

TODO
ocaml: stdlib, base, core, containers, batteries,
rescipt: belt, tablecloth

2.2.3 ML-синтаксис и Reason-синтаксис

TODO

2.2.4 Библиотеки конкурентного программирования

lwt, async, eio TODO

2.2.5 Библиотеки для построения пользовательского интерфейса

TODO
js_of_ocaml: jsoo-react, ocaml-vdom, virtual_dom, incr_dom, bonsai
rescript: rescript-react, bucklescript-tea

2.2.6 Библиотеки для RPC

TODO

2.2.7 Web-фреймворки

dream TODO

2.2.8 Библиотеки для тестирования

TODO

2.3 Выбор библиотек и подходов

TODO

3 Третья глава TODO

Здесь будет третья глава. TODO

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Критика чистого разума Сочинения в шести томах. Том 3 Академия наук СССР Институт философии Издательство социально-экономической литературы «Мысль» Москва - 1964. TODO

2. Роуэлл, Г. Физика : учебное издание / Г. Роуэлл, С. Герберт. – Москва : Просвещение, 1994. – 576 с. – ISBN 5-09-002920-2.

3. Math for Progammers TODO https://www.manning.com/books/math-for-programmers

4. Larson R., Hostetler R. . Precalculus: A Concise Course. — Boston: Houghton Mifflin, 2007. — xvii + 526 + 102 p. — ISBN 0-618-62719-7. TODO

5. Алексеев, В. Б. Теорема Абеля в задачах и решениях. — М.: МЦНМО, 2001. — 192 с. — ISBN 5-900916-86-3. TODO

6. TODO Autar K Kaw Numerical Methods with Applications Chapter 03.03 Bisection Method