

KNOX

Curves

Knox Game Design

July 2022

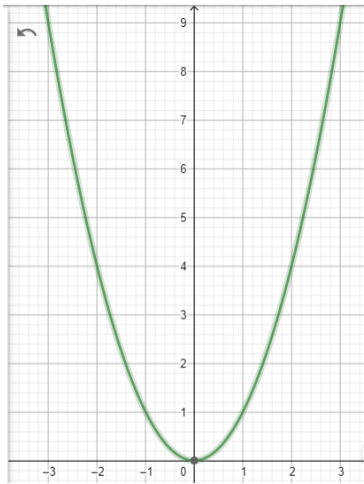
Levi D. Smith

DESIGN

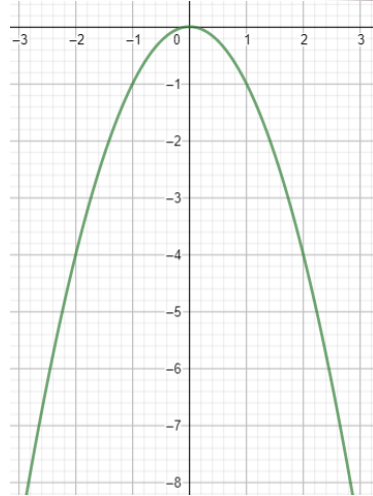
Curve applications for games

- Shooting a heavy projectile
- Wave projectiles
- Jumping
- Path, train tracks

Basic Parabola

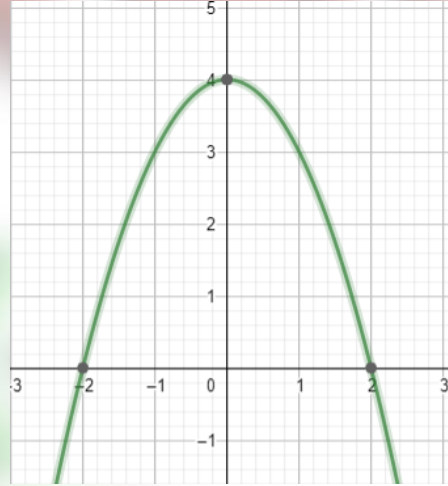


$$y = x^2$$



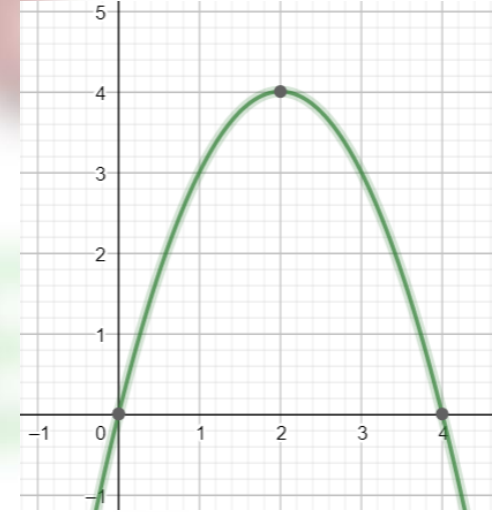
$$y = -x^2$$

flip on x axis



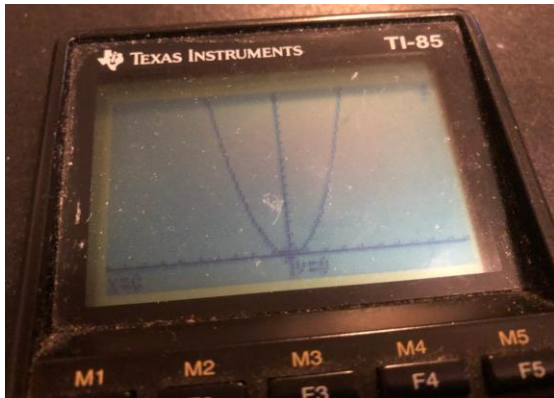
$$y = -x^2 + 4$$

shift up on y axis



$$y = -(x - 2)^2 + 4$$

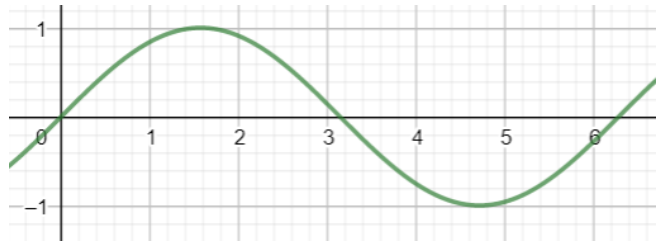
shift right on x axis
by x intercept to
center on origin



Find x intercepts by substituting 0
for y and solving for x
 $0 = -x^2 + 4$
 $x^2 = 4$
 $x = 2, -2$

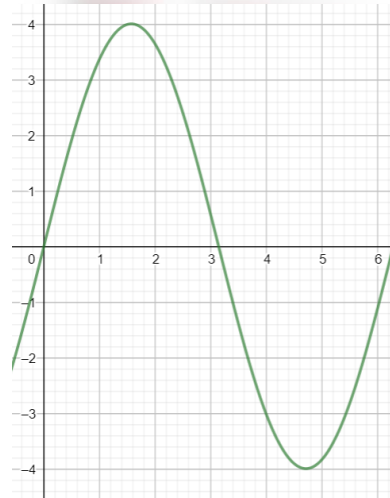
KNOX
GAME
DESIGN

Sine wave



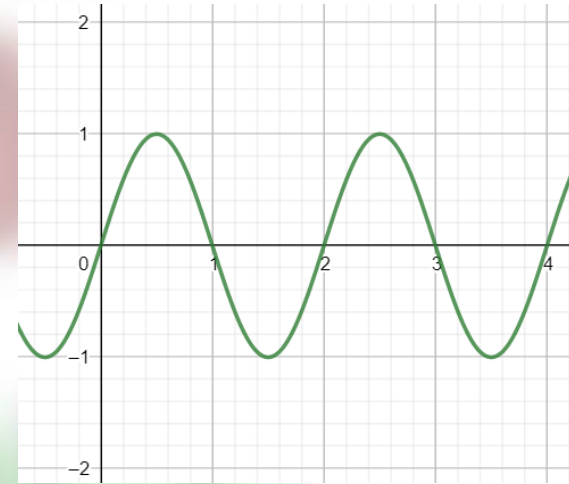
$$y = \sin(x)$$

x intercepts are π (3.14), 2π (6.28)...



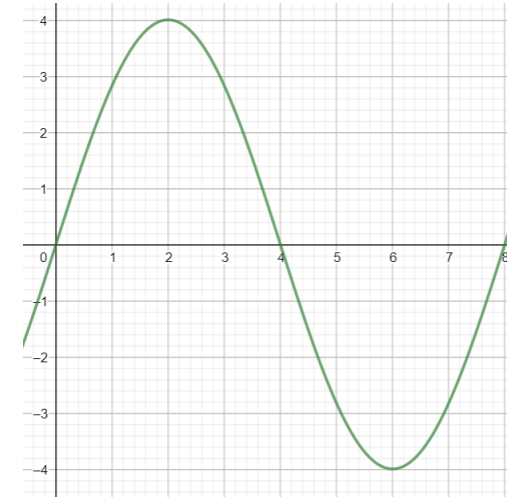
$$y = 4 * \sin(x)$$

Increase amplitude
Max $y = 4$, Min $y = -4$



$$y = \sin(\pi * x)$$

Normalize frequency
x intercepts to 1, 2, 3, 4...



$$y = 4 * \sin((\pi / 4) * x)$$

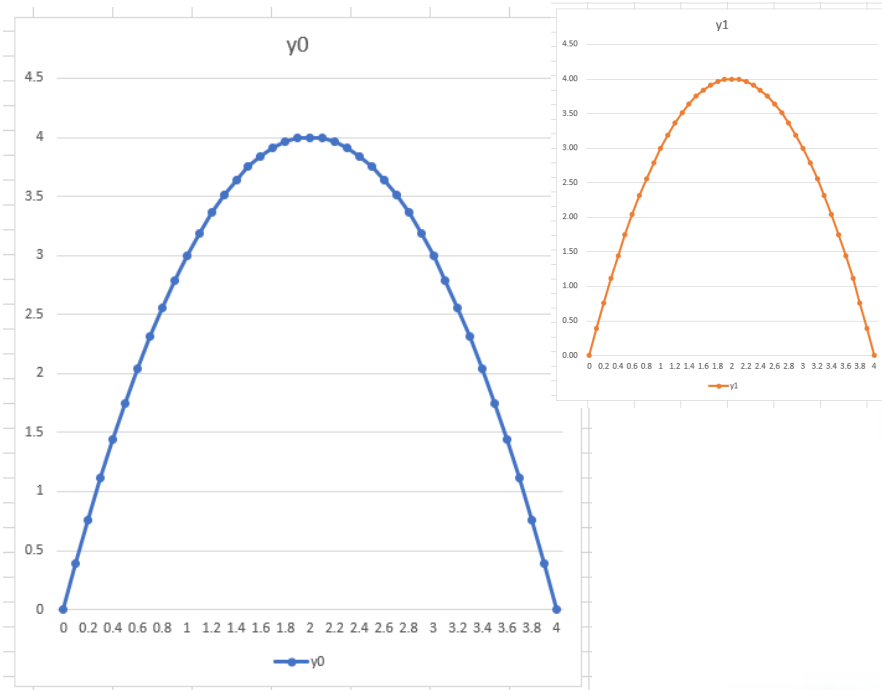
Decrease frequency
x intercepts 4, 8...

DESIGN

KNOX
GAME
DESIGN

Physics, Velocity, Acceleration

Distance = m (meters)
 Time = s (seconds)
 Velocity = m / s (meters per second)
 Acceleration = m / s² (meters per second squared)



time	delta time	y	init vel	y vel	gravity	y0	y1	conv	y0 conv	y1 conv
0	0	19.6	19.60	-9.8	-9.8	0	0	0.204082	0	0.00
0.1	0.1	19.6	18.62	-9.8	1.91	1.91	1.91	0.204082	0.39	0.39
0.2	0.1	19.6	17.64	-9.8	3.72	3.72	3.72	0.204082	0.76	0.76
0.3	0.1	19.6	16.66	-9.8	5.44	5.44	5.44	0.204082	1.11	1.11
0.4	0.1	19.6	15.68	-9.8	7.06	7.06	7.06	0.204082	1.44	1.44
0.5	0.1	19.6	14.70	-9.8	8.58	8.58	8.58	0.204082	1.75	1.75
0.6	0.1	19.6	13.72	-9.8	10.00	10.00	10.00	0.204082	2.04	2.04
0.7	0.1	19.6	12.74	-9.8	11.32	11.32	11.32	0.204082	2.31	2.31
0.8	0.1	19.6	11.76	-9.8	12.54	12.54	12.54	0.204082	2.56	2.56
0.9	0.1	19.6	10.78	-9.8	13.67	13.67	13.67	0.204082	2.79	2.79
1	0.1	19.6	9.80	-9.8	14.70	14.70	14.70	0.204082	3	3.00
1.1	0.1	19.6	8.82	-9.8	15.63	15.63	15.63	0.204082	3.19	3.19
1.2	0.1	19.6	7.84	-9.8	16.46	16.46	16.46	0.204082	3.36	3.36
1.3	0.1	19.6	6.86	-9.8	17.20	17.20	17.20	0.204082	3.51	3.51
1.4	0.1	19.6	5.88	-9.8	17.84	17.84	17.84	0.204082	3.64	3.64
1.5	0.1	19.6	4.90	-9.8	18.38	18.38	18.38	0.204082	3.75	3.75
1.6	0.1	19.6	3.92	-9.8	18.82	18.82	18.82	0.204082	3.84	3.84
1.7	0.1	19.6	2.94	-9.8	19.16	19.16	19.16	0.204082	3.91	3.91
1.8	0.1	19.6	1.96	-9.8	19.40	19.40	19.40	0.204082	3.96	3.96
1.9	0.1	19.6	0.98	-9.8	19.55	19.55	19.55	0.204082	3.99	3.99
2	0.1	19.6	0.00	-9.8	19.60	19.60	19.60	0.204082	4	4.00
2.1	0.1	19.6	-0.98	-9.8	19.55	19.55	19.55	0.204082	3.99	3.99
2.2	0.1	19.6	-1.96	-9.8	19.40	19.40	19.40	0.204082	3.96	3.96
2.3	0.1	19.6	-2.94	-9.8	19.16	19.16	19.16	0.204082	3.91	3.91
2.4	0.1	19.6	-3.92	-9.8	18.82	18.82	18.82	0.204082	3.84	3.84
2.5	0.1	19.6	-4.90	-9.8	18.38	18.38	18.38	0.204082	3.75	3.75
2.6	0.1	19.6	-5.88	-9.8	17.84	17.84	17.84	0.204082	3.64	3.64
2.7	0.1	19.6	-6.86	-9.8	17.20	17.20	17.20	0.204082	3.51	3.51
2.8	0.1	19.6	-7.84	-9.8	16.46	16.46	16.46	0.204082	3.36	3.36
2.9	0.1	19.6	-8.82	-9.8	15.63	15.63	15.63	0.204082	3.19	3.19
3	0.1	19.6	-9.80	-9.8	14.70	14.70	14.70	0.204082	3	3.00
3.1	0.1	19.6	-10.78	-9.8	13.67	13.67	13.67	0.204082	2.79	2.79
3.2	0.1	19.6	-11.76	-9.8	12.54	12.54	12.54	0.204082	2.56	2.56
3.3	0.1	19.6	-12.74	-9.8	11.32	11.32	11.32	0.204082	2.31	2.31
3.4	0.1	19.6	-13.72	-9.8	10.00	10.00	10.00	0.204082	2.04	2.04
3.5	0.1	19.6	-14.70	-9.8	8.58	8.58	8.58	0.204082	1.75	1.75
3.6	0.1	19.6	-15.68	-9.8	7.06	7.06	7.06	0.204082	1.44	1.44
3.7	0.1	19.6	-16.66	-9.8	5.44	5.44	5.44	0.204082	1.11	1.11
3.8	0.1	19.6	-17.64	-9.8	3.72	3.72	3.72	0.204082	0.76	0.76
3.9	0.1	19.6	-18.62	-9.8	1.91	1.91	1.91	0.204082	0.39	0.39
4	0.1	19.6	-19.60	-9.8	0.00	0.00	0.00	0.204082	0	0.00

- Example data uses 10 frames per second
- Typical games use 60 frames per seconds
- Physics uses continuous variables, but games use discrete update events
 - y0 uses continuous calculation
 - must track total lifetime
 - calculated from initial velocity
 - y1 uses discrete values
 - velocity and position updated on each update
 - new velocity calculated based on previous velocity and acceleration
- Discrete is simpler, does not require keeping track of lifetime. Probably good enough for most games
- Unity provides FixedUpdate method to ensure that update intervals are always the same
- Could make gravity 1 m/s² to simplify calculations
- Issues with setting full velocity on button press

- gravity = -9.8 m/s²
- max height = 4 m
- y initial velocity = 19.6 m/s
 - gravity * max height / 2
- $y_0 = (y \text{ initial velocity} + (0.5 * \text{time} * \text{gravity})) * \text{time}$
 - Units: m = m/s - (s * m/s²) * s
- $y_1 = \text{previous position} + (\text{current velocity} * \text{delta time}) + (0.5 * \text{gravity} * \text{delta time}^2)$
 - Add (gravity * delta time) to get new velocity
- multiply by constant to set max value
 - const = max y / y starting velocity
 - const = 4 / 19.6 = 0.2041

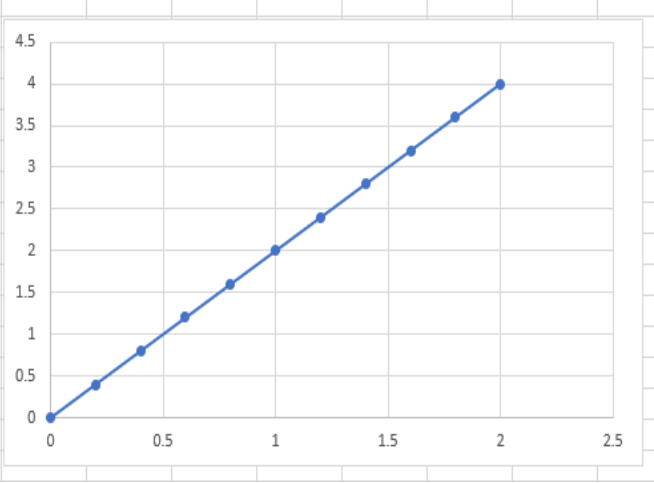
MMX
GAME
DESIGN

Bezier

- Two or more control points
- First point is start, Last point is end
- Must keep track of current time, 0 to 1

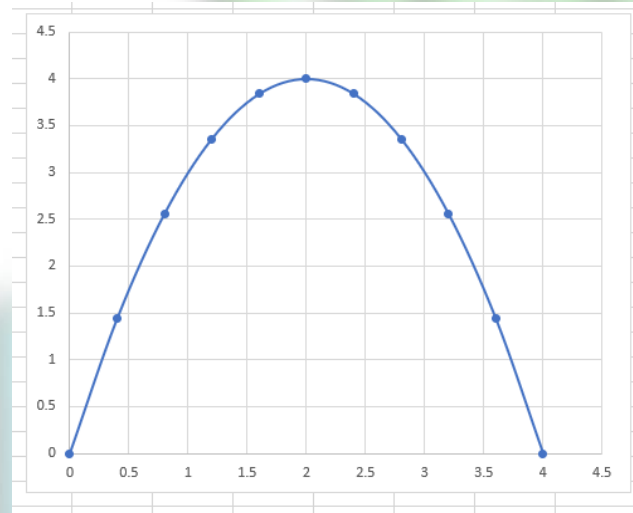
	x	y	x	y	x	y
P0	0	0	0	0	0	0
P1	2	4	2	8	2.66	12
P2			4	0	5.33	-12
P3					8	0

t	1-t	P.x	P.y	P.x	P.y	P.x	P.y
0	1	0	0	0	0	0	0
0.1	0.9	0.2	0.4	0.4	1.44	0.79829	2.592
0.2	0.8	0.4	0.8	0.8	2.56	1.59712	3.456
0.3	0.7	0.6	1.2	1.2	3.36	2.39643	3.024
0.4	0.6	0.8	1.6	1.6	3.84	3.19616	1.728
0.5	0.5	1	2	2	4	3.99625	0
0.6	0.4	1.2	2.4	2.4	3.84	4.79664	-1.728
0.7	0.3	1.4	2.8	2.8	3.36	5.59727	-3.024
0.8	0.2	1.6	3.2	3.2	2.56	6.39808	-3.456
0.9	0.1	1.8	3.6	3.6	1.44	7.19901	-2.592
1	0	2	4	4	0	8	0



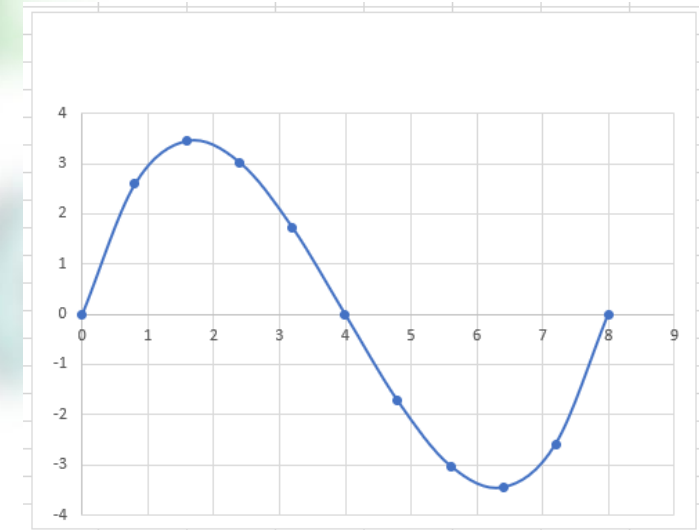
Two control points

$$P = (1-t)P_1 + tP_2$$



Three control points

$$P = (1-t)^2P_1 + 2(1-t)tP_2 + t^2P_3$$



Four control points

$$P = (1-t)^3P_1 + 3(1-t)^2tP_2 + 3(1-t)t^2P_3 + t^3P_4$$

KNOX
GAME
DESIGN

Curve applications for games

- Shooting a heavy projectile, throwing grenade - **Basic parabola**
- Wave projectiles - **Sine wave**
- Jumping - **Physics, Velocity, Acceleration**
- Enemy movement path, train tracks - **Bezier**