JP's Arcade Physics v3.0

(jpsalas 2020. 2nd edition)

Why new physics settings?

For several years I have been using the default VPX physics, which are explained in the file PhysicValues.txt, and they are quite good. But there were always a few things that I always felt they could be better. Many other authors have made their own changes and all them are quite good. But I wanted to use what VPX has to offer, and simply adjust some parameters to make the tables play a little closer to a real pinball. I always felt the default physics make the ball feel more like a football (a soccer ball) or even a billiard ball, more than a pinball ball. The same for the flippers which were missing some functionality, and the shooting angles were not optimal.

Here it is what I wanted to accomplish:

- Stop the ball making strange stops and changes in speed or direction.
- Increase side to side action.
- Enable flipper tricks like drop catches, cradle separation and backhands.
- Consistent flipper shooting angles, including backhands shots.
- Reduce ball rebound on top of a flipper when it is up.
- Better ball bounce on objects, mostly rubbers, both at higher and lower speeds.
- Easy way to control the ball speed.
- The settings should not hogg the cpu.

To do all these I wanted a set of values that should be the same on all the tables, older and newer, and they should be easy to add to a table. Rubbers should be rubbers on all the tables, they should not change elasticity or friction values, and the same applies to metals, plastics and wood. Flippers should work just as good on older tables than in new tables.

I called these physics "JP's Arcade Physics 3.0", because, yes, they are my third effort to get them right :) And why the name "Arcade", well, I had to call them something ;)

These physics 3.0 set started when I was watching the PAPA videos to learn to make flipper tricks. They were using the Stern Star Trek pinball machine, so I build my own VPX Start Trek and I started testing flipper settings. There is only one flipper trick I didn't managed to do, and this is a flipper pass by tapping the flipper button. If I changed the flippers to do this trick then those settings ruined the normal flipper behavior, and they feel sluggish.

You will find several files in the download zip:

- JP's Arcade Physics v3.0.pdf *this file which explains all the settings*
- Rubber_Objects.vpx *all the rubber objects you can cut & paste to your own tables*
- Elasticity_Test.vpx *a simple test of the new elasticity settings*
- JP's Arcade Physics v3.0.vpp *table and flipper settings*
- JP's Physics Materials v3.0.mat *the physic materials for easily change the object properties. I recommend using collection.*
- JP's Arcade Physics v3.0.vbs *a simple script you can copy & paste.*

I'll be uploading several of my tables changed with these settings, which you may test these settings.

The Settings

	Metals	**Plastic**	**Wood**	(made mostly with wall objets)
Elasticity	0,2	0,15	0,25	*very low compared to rubbers*
Elasticity FallOff on Metals/Plastic/Wood				*0 because they are usually walls and don't have a fallout value. If you are using primitives for these objects then you may still use these values, although you may want to change them all, elasticity, falloff and friction.*
Friction	0,15	0,15	0,15	*also very low values*

Rubbers	**Elasticity**	**Elasticity FallOff**	
(posts)	0,95	0,5	*normal sized rubber. Thick plastic or metal pegs with a round rubber*
(pins)	0,85	0,55	*thinner rubber. These are metal pegs with just a thinner round rubber*
(pegs)	0,9	0,5	*thin rubber. I call "pegs" those metal pegs with a thin rubber sleeve.*
(long bands)	0,95	1,7 + -	*very long rubbers so they can absorb high speed balls*
(short bands)	0,95	0,75 + -	*a little less bouncing that on posts*

Other rubbers with different lengths should have values between 0,75 and 1,7 depending on the length of the rubber.

Friction 0,25

Rubbers Hit height	27	
Rubbers Height	30	*pins are a little higher, at 32, but the hit height should still be 27 for better VPX collision and physics*.

Scatter Angle 5 + *on everything. Bumpers can usually be 10 or higher*.

Flipper size 3” (Williams/Stern)

Rubber: 5,9, 17,6 , 23,5

Size: 20,6 11,7 114,7

Angle: between 50-53 degrees, f ex.

Start angle: 118 to 122, depending on the table. This angle should be aligned with the inlane plastic, so the ball should not jump or change its speed.

End angle: 68 to 72. 68 will give you better strait up angle shots and backhand shots. It will make it easier to trap the ball.

End angle of 70 is average, but backhands will be weaker.

End angle of 72 will make it harder to catch the ball, but it will make it easier for the ball to pass from one flipper to the other. Backhands shots will be harder to do.

Flipper settings (all flippers: 2.5”and 3”)

Mass	1,5	
Strength	5000	very strong, use Const maxvel to limit ball speed in rolling sub
Elasticity	0,85	thin rubber, but can be changed to thick rubber (0.95)
Elasticity Falloff	1,2 +-	from 0,5 to 1,7, 1.7 gives hard rubbers, this is easy flippers.
Friction	0,25	
Return Strength	0,15	
Coil Ramp up	0	
EOS Torque	0,75	(it is adjusted in the script’s flipper subs from 0,1 to 0,75) EOS
Torque Angle 10		

Playfield settings

Gravity constant	0,980665		
Playfield friction	0,025		
Playfield Elasticity	0,2		
Contact Scatter Angle	5		
Min & Max Slope	4 +- (EM)	5+ (early SS)	6 +- (SS)

Ramps	same as metal/plastic
Bumpers strength	10+
Slingshots strength	10+
Plunger strength	140+

Script changes

The changes in the script are minimal. Just 3 main changes:

- the ball size and mass
- the flipper section with the EOS changes
- the rolling sound routine which includes the ball speed control.

Ball: Size: 50

Mass: 1,7 this value fits well to a flipper's mass, and it was first mentioned by Clarkkent in his flipper settings. Due to this higher ball mass, the bumpers & plunger need to increase their force. With the ball speed control it is not a necessity to use this higher ball mass, but it will give you a better range of speed on the bumpers, and the ball will make some more small bounces on the rubbers at low speeds.

Flippers:

Change EOSTorque value in the script when the keyup or keydown is pressed to reduce the rebound when the flipper is up, as this value will be high, and to enable some flipper tricks like cradle separations and drop catches when the flipper is on its way down..

```
Sub SolLFlipper(Enabled)
  If Enabled Then
    PlaySoundAt SoundFX("fx_flipperup",DOFContactors), LeftFlipper
    LeftFlipper.EOSTorque = 0.75: LeftFlipper.RotateToEnd
  Else
    PlaySoundAt SoundFX("fx_flipperdown",DOFContactors),LeftFlipper
    LeftFlipper.EOSTorque = 0.1: LeftFlipper.RotateToStart
  End If
End Sub

Sub SolRFlipper(Enabled)
  If Enabled Then
    PlaySoundAt SoundFX("fx_flipperup",DOFContactors), RightFlipper
    RightFlipper.EOSTorque = 0.75: RightFlipper.RotateToEnd
  Else
    PlaySoundAt SoundFX("fx_flipperdown",DOFContactors),RightFlipper
    RightFlipper.EOSTorque = 0.1: RightFlipper.RotateToStart
  End If
End Sub
```

Ball speed control:

Added to the ball rolling routine. Change maxvel variable according to table, for ex. 25-30 for EM, 30-40 early SS or 40-50 for modern tables with high ramps. It all depends of the speed you want to give to the table.

```
Const maxvel = 45 'max ball velocity

    If BOT(b).VelX AND BOT(b).VelY <> 0 Then
      speedfactorx = ABS(maxvel / BOT(b).VelX)
      speedfactory = ABS(maxvel / BOT(b).VelY)
      If speedfactorx <1 Then
        BOT(b).VelX = BOT(b).VelX * speedfactorx
        BOT(b).VelY = BOT(b).VelY * speedfactorx
      End If
      If speedfactory <1 Then
        BOT(b).VelX = BOT(b).VelX * speedfactory
        BOT(b).VelY = BOT(b).VelY * speedfactory
      End If
    End If
```