RocketModule

/\*

Coupled to LoadQueue because module requires images.

Coupled to gravity object, gameover object

Coupled to resetGame text, stage

Coupled to resetGaameValues background, landingsite

\*/

//SpriteSheet objects don’t go in container

SpriteSheet rocket\_sheet;

SpriteSheet fire\_sheet;

SpriteSheet thruster\_sheet;

Object thrust; //constructor??

xThrust property

set / get //requires rocket angle, thrustLevel

yThrust property

set / get

degreesToRadians(degrees)

radiansToDegrees(radians)

thrustChanged event //??where to put

updateFireAnimation

Object altitude;

heightAboveSeaLevel property

get / set

Object velocity; //depends on gravity

xVelocity property

set / get

yVelocity property

set / get

aVelocity property //angular

set / get

Object fuels;

rocketFuel property

get / set

decreaseFuel

monoFuel property

get / set

decreaseFuel

outOfMono event

cutoutThrusterAnimation

outOfRocketFuel event

cutoutFireAnimation

Sprite fire; //or make Object fire??

showTinyFire

showSmallFire

showMediumFire

showLargeFire

cutoutFireAnimation

engineFiring event

engineSmoke()

Object thruster //create two, one left, one right

Sprite object

orientation property //direction to orient sprite

showThrust

cutoutThrust

thrusterFiring event

thrusterSmoke()

//smoke sprite is added to stage, not rocket, when created

Object smoke

buildSmokeSprite(x,y)

fadeout(e)

removeSprite()

Container Rocket;

//variables

const THRUST; //base thrust value of this rocket

const START\_FUEL; //base rocket fuel

const START\_MONO; //base monopropellant

const START\_VX //base x velocity

const START\_VY //base y velocity

const START\_Y //base y position

const PIXELS\_PER\_METER //conversion ratio based on image

Int thrustLevel

//Objects

Object thrust;

Object altitude;

Object velocity;

Object fuels;

//Shape objects in rocket container

Shape tinyPt;

Shape smallPt;

Shape mediumPt;

Shape largePt;

Shape thrusterPtL;

Shape thrusterPtR;

Shape center\_of\_mass;

//Sprite objects in rocket container

Sprite body;

Sprite legs;

Sprite fire;

Thruster thrusterL;

Thruster thrusterR;

//initialization

buildRocket()

buildBody()

buildLegs()

buildFire()

buildThrusters()

buildCenterOfMass()

buildThrusterPts()

buildFirePts()

placeRocket(regX,regY,angle, slice)//calc x,y, angle in other module

//place behind slice bitmap

resetRocketProperties() //altitude, velocity, fuels

//movement

updateRocket

~~calcNextRotation(aKeyDown, dKeyDown)~~

rotateLeft() //limited by mono, keep rotate if no input??

rotateRight() //limited by mono

calcNextPosition(wKeyDown, gravity) //better way??

getStandardAngle(rotation)

detectCollision(nextPt, landingSite) //better way??

renderRocket()

rocketMovement event //y value changes

updateAltitude

updateFuels //engine or thruster used?

rocketLanded event

landedSequence //includes animation, hiding rocket

resetGame //coupled to stage

rocketCrashed event

crashSequence

resetGame