RocketModule

/\*

coupled to init() and LoadQueue because module requires 4 HTMLImageElements:

* Falcon9.png
* Falcon9Fire.png
* Falcon9Thrusters.png
* Smoke.png

Also because need to add to stage

Coupled to GameModule because thrustLevel variable and thrustChanged flag is used in animation

\*/

var rocket; //container to pass on to other modules

var rocket\_sheet, fire\_sheet, thruster\_sheet; //spritesheets

//even these could be referenced through the rocket, through the particular sprites; don’t actually need reference to these; however, makes building body and legs sprites easier if only need to build spritesheets once and reference them

//get all other objects through rocket container, using names

//var tinyPt, smallPt, mediumPt, largePt, thrusterPtL, thrusterPtR; //move internal to rocket

//functions

buildSpriteSheets();

buildRocket(regX, regY, angle){

buildBody();

buildLegs();

buildFire();

buildThrusters();

buildCenterOfMass(); //doesn’t need argument because only for rocket

buildThrusterPoints(); //tied to specific sprite

buildFlamePoints(); //tied to specific sprite

}

removeRocket(container); //coupled to stage, decouple by adding parameter to method

~~buildAndPlaceRocket();~~ //find a way not to rebuild every time

placeRocket();

//rocket animations

~~updateEngine();~~

showTinyFire(); //add instead to decouple from thrustLevel? What about thrustChanged flag?

showMediumFire ();

showLargeFire();

showSmallFire();

cutEngineAnimation();

~~updateThrusters();~~

showLeftThruster();

showRightThruster();

cutLeftThruster();

cutRightThruster();

cutThrustersAnimation();

flareThrusters();

landedAnimations();

crashAnimations(level);

//smoke animation

buildSmoke();

thrusterSmoke();

engineSmoke();

fadeout(e);

smokeComplete();

GUIModule

/\*

coupled to StartupModule - init() and LoadQueue - because module requires an HTMLImageElement:

* Loading.png

Also because required to add to stage

Coupled to GameModule:

* thrustLevel
* altitude
* velocityX
* velocityY
* fuel
* mono
* THRUST
* START\_FUEL
* START\_MONO
* START\_VX
* START\_VY
* PIXELS\_PER\_METER

Coupled to RocketModule:

* rocket.rotation

\*/

var endText, physicsText, helpText, pausedText;

var fuelBar\_fuel, fuelBar\_mono; //these replace references to graphics command objects

//var fuelText; //if fuel bars are made into a class with text and shape

//initialization

buildText(txt, style, color, alignment, x, y, visible, alpha); //return Text object

buildBar(x,y, type, fillColor); //return FillBar object

//update methods

updateStats();

updateFuelBars();

LandingSiteModule

/\*

Coupled to StartupModule because must add to stage

\*/

var landingSite; //hidden Shape indicates where rocket can land

buildLandingSite(); //design so that landing site can be moved?

toggleLandingSite(); //make visible or invisible for diagnostic purposes

BackgroundModule

/\*

Background is built in two pieces to simulate a solid landing surface through which rocket flames cannot penetrate. This is accomplished by having the background be a full image, and including a horizontal slice of the bottom horizontal half of the landing site. If the slice is kept in front of the rocket and its sprites, rocket flames that extend past the horizontal center are hidden.

coupled to StartupModule - init() and LoadQueue - because module requires multiple HTMLImageElements:

* Ocean.png
* OceanSlice.png
* Earth.png
* EarthSlice.png

Also because required to add to stage

Coupled to GameModule:

* Level

Coupled to LandingSiteModule

- images depend on location vertically of landing site, so this can’t be altered once decided

\*/

var eBackground, eSlice, //earth background set

var oBackground, oSlice; //ocean background set

//initialization

~~buildEarthBackground();~~ //replace with one method builds a background set

~~buildOceanBackground();~~ //slice and background are same height to simplify to one method

buildBitmap(image); //returns Bitmap, can be stored as slice or background

//misc

showBackground(name); //”earth” or “ocean”, keeps mechanics for visibility inside module

StartupModule

/\*

coupled to HTML canvas element

coupled to RocketModule, GUIModule, BackgroundModule, LandingSiteModule, GameModule

\*/

//contains Ticker, Window already

var stage, queue;

init(); //load HTML Image Elements

load(); //build stage, game objects, GUI, background, landingSite

setGameVariables(); //sets initial values for all game variables

InputModule

/\*

Coupled to GameModule for pausing, thrustLevel, level

Move flags to GameModule instead?

\*/

const SPACEBAR = 32;

const LEFT\_ARROW = 37;

const UP\_ARROW = 38;

const RIGHT\_ARROW = 39;

const DOWN\_ARROW = 40;

const A\_KEY = 65;

const D\_KEY = 68;

const S\_KEY = 83;

const W\_KEY = 87;

var wKeyDown = sKeyDown = dKeyDown = aKeyDown = false;

var spacebarDown, leftArrowDown, rightArrowDown, upArrowDown, downArrowDown;//add flags to decouple?

buildKeyInput(); //method sets window.onkeydown and window.onkeyup

detectKey(); //need to decouple from calling methods directly – pause, thrustLevel, changeLevel

removeKey();

MovementModule

/\*

should this be added to RocketModule instead of being a separate module??

\*/

const THRUST = 35;

updateRocket(); //RocketModule

calcNextPosition(); //RocketModule, GameModule

calcNextRotation(); //RocketModule, GameModule, InputModule

getStandardAngle(rotation); //no coupling

getYThrust(angle); //GameModule

getXThrust(angle); //GameModule

degreesToRadians(degrees); //no coupling

radiansToDegress(radians); //no coupling

detectCollision(pt); //RocketModule, LandingSiteModule, GameModule

renderRocket(); //RocketModule

GameModule

/\*

run coupled to MovementModule

* updateRocket,
* updateAnimations
* updateFuelLevels
* updateAltitude

pause coupled to GUIModule

- pausedText

- updateStats

- updateFuelBars

endingSequence coupled to RocketModule, GUIModule

- landedAnimations

- removeRocket

- crashAnimations

- gameoverText //remove coupling by adding name to child and using method to get gameoverText

\*/

const START\_FUEL = 500;

const START\_MONO = 100;

const START\_VX = 0;

const START\_VY = 10;

const PIXELS\_PER\_METER; //really based on images of rocket, but those are fixed and won’t change during this game

var gravity = 9.81; //maybe so gravity could be varied in future expansions of game

var thrustLevel; //move to rocket as property?

var level;

var velocityX, velocityY; //move to rocket as property?

var altitude; //move to rocket as property?

var landed; //move to rocket as property?

var gameover;

var count; //used to limit endingSequence to being called once during run, rather than multiple times

var fuel; //move to rocket as property?

var mono; //move to rocket as property?

var thrustChanged;

run(e); //coupled to MovementModule, RocketModule, GUIModule, StartupModule

pause(); //coupled to GUIModule, StartupModule (stage), InputModule

endingSequence(); //RocketModule, GUIModule

resetGame(); //coupled to GUIModule, RocketModule

resetGameValues(); //coupled to BackgroundModule, GUIModule

changeLevel(); //coupled to InputModule

increaseThrust(); //coupled to InputModule

decreaseThrust(); //coupled to InputModule

updateFuelLevels(); //coupled to InputModule

reduceMono();

reduceFuel();

updateAltitude(); //coupled to LandingSite, RocketModule