Side Notes

Catapult Wars Training Kit – Create Games for Windows Phone 7

Lab version: 1.0.0

Last updated: 9/29/2010

Contents

[Overview 3](#_Toc272700767)

[Exercises 4](#_Toc272700768)

Overview

# Catapult Wars Training Kit

* 1. This training kit is a set of resources aimed at assisting the trainer in leading the trainees to fully understand and experiment with building a 2D game. The kit consists of a set of small exercises for trainees, and a larger advanced exercise.

# File Structure

* 1. The exercises are located under the Exercises folder, each within its own subfolder containing the exercise solution.
  2. The advanced example is located under the Advanced folder.

# Regular vs. Advanced Exercises

* 1. This document describes a few regular exercises and one advanced exercise. The regular exercises are basic and demonstrate how to add one feature—such as a health bar or supply crate, each considered elementary for 2D games—at a time. It is important that regular exercises are understood before performing the advanced exercise. The Advanced exercise illustrates a combined feature set working together over a virtual screen space, incorporating different points of view, and providing a performance challenge.
  2. All of the above features are located within a static 2D screen.

Exercises

# Exercise – Health Bar

* 1. Manage and display a health bar for each player in the Heads Up Display (HUD):
  + Health decreases when hit by the opposing catapult’s rock.
  + A player’s catapult is destroyed when its health is depleted.
  + The decrease in health is scaled down from the center of the catapult (hitting it in the center causes the most damage).
  1. Figure 1. Health bars added for both players
  2. 

# Exercise - Second Human Player

* 1. Change the second player, currently controlled by Artificial intelligence (AI), to a human player:
  + Reverse the shot power feedback arrow direction.
  + Change the arrow’s color to red.
  + Acquire a user input for the second player.
  1. Figure 2. Second human player firing
  2. 

# Exercise – Shot Angle

* 1. Allow the user to define the shot’s angle:
  + Expand the **Drag** gesture usage to allow **FreeDrag** instead of **HorizontalDrag**.
  + Calculate the angle and velocity by using the **DragEnd** gesture's X and Y positions in relation to the gesture's original location.
  1. Figure 3. Shot angle
  2. 

# Exercise – Shot Guide

Calculate the ballistic curve while the user drags to select angle and velocity:

* + Show the curve as a dotted line ending where the projectile would hit.
  1. Figure 4. Shot Guide displayed
  2. 

# Exercise – Supply Crate

* 1. Add another object—a cargo box—to the game:
  + Each player has a cargo box rendered in front of the player's catapult.
  + A player that hits the opponent's cargo box gets a special weapon.
  + The special weapon is a rock that splits into three rocks midflight.
  + Support the three-rock weapon by calculating the descent of each rock with a variant fall path from the highest point in the projectile flight path.
  1. Figure 5. Supply crate and triple projectile
  2. 

# Advanced Exercise - Scrolling Screen

* 1. Expand the battlefield horizontally, extending the virtual size beyond the phone's hardware screen limits:
  + Use **Drag** and **Flick** gestures to enable battlefield panning. Remember to calculate the kinetic movement when detecting the **Flick** gesture.
  + Use the **Pinch** gesture to zoom in and zoom out of the battlefield. Use the hardware scalar to recalculate the displayed graphic elements upon zooming.
  + Support a chase camera mode that moves the battlefield background to track launched projectiles while in flight.
  1. Figure 6. Scrolling screen
  2. 