Paul Wallace

10/30/2012

2D Graphics

HeliDrop Game Document

1. HeliDrop
   1. Copyright Information
2. Implementation Team
   1. Paul Wallace
      1. (508) 246-2892
      2. PaulWallacc@gmail.com
   2. Programmer, Designer, Team Leader
      1. Paul Wallace
   3. Artist
      1. Currently in need of an artist.
3. Executive Summary
   1. Project Overview
      1. HeliDrop will be a 2d game where you must drop crates and other objects from a helicopter onto people running on a pier below. For the scope of the project, there will not be any scrolling background or landscape. Instead you will see one motionless environment where the players move about. I will incorporate animation, depth, particles, and pixel shader effects to create a visually pleasing game. This project will have to be completed in a little over five weeks. Knowing the tight deadline I have a small number of in scope features for the game.
      2. HeliDrop will be coded in C# using Microsoft’s XNA 4.0 framework.
   2. Delivery Platforms
      1. PC
         1. HeliDrop testing will take place on machines running the Windows OS. At a minimum they will be have 4gb of memory and quad core processors. Although I am sure that there are older systems that would have no trouble running the game I will not be doing any such testing. Proceed at your own risk.
         2. HeliDrop will be a very small game so disk space will not be much of a factor.
      2. XBOX
         1. HeliDrop will be a free download via the Xbox Live marketplace. Its total size is currently unknown but it will be quite small. Disk space will not be an issue.
4. SECTION V: DEVELOPMENT PLAN
   1. Game Mechanics/Player movement
      1. Helicopter
         1. PC
            1. In Scope

The helicopter will be able to move back and forth across the screen using the left and right arrow keys. There will only we two dimensional movement.

You will be able to drop a crate using the spacebar.

* + - * 1. Out of Scope

Using depth, the helicopter will appear to be able to move in four directions: left, right, toward the user, and away from the user.

* + - 1. XBOX
    1. Runners
       1. In Scope
          1. The runners will not be controllable. They will be randomly generated and will run across the screen at different speeds. Runners will appear at different depths but collision will only take place in one dimension.
       2. Out of Scope
          1. The runners will not be controllable. They will be randomly generated and will run across the screen at different speeds. For the out of scope feature, the runners will appear at different depths but unlike the in scope requirement the collision detection will be based on two dimensions.
  1. Physics
     1. The only physics I will need to incorporate into the game will be the crate that falls from the helicopter.
  2. Artificial Intelligence
     1. Because I am very unfamiliar with artificial intelligence the computer controlled players in HeliDrop will not be very robust. This isn’t an issue because there isn’t much room in the game to include sophisticated AI as the computer controlled players will only be running along a straight boardwalk.
  3. Graphics
     1. Animation
        1. In Scope
           1. People running along board walk
        2. Out of Scope
           1. Helicopter
           2. Crate falling from helicopter
           3. Clouds
           4. Birds
     2. Particle System
        1. In Scope
           1. Exhaust from helicopter
        2. Out of Scope
           1. Footsteps from runners
           2. Dust off runners feet
           3. Explosions
           4. Dust when box lands
           5. Rain
     3. Effects
        1. In Scope
           1. Reflection of game on water in foreground
        2. Out of Scope
           1. Nigh time game play
           2. Light from helicopter during night time play.
           3. Black and white game play
     4. Camera
        1. The camera will not move; there will only be one perspective.
  4. Collision
     1. Crates hitting people
        1. In Scope
           1. Simple collision based on coordinates
        2. Out of Scope
           1. Research into a more advanced form of collision detection
  5. GUI
     1. HUD
        1. In Scope
           1. Game time
        2. Out of Scope
           1. Score
           2. Objectives
           3. Fuel gauge
           4. Crate hits
     2. Menu Flow Diagram
        1. In Scope
           1. Use the generic Microsoft menu layout and give credit.
        2. Out of Scope
           1. Create a custom menu system that suits HeliDrop.
  6. Sound
     1. In Scope
        1. No sounds will be included until the game has met all in scope requirements.
     2. Out of Scope
        1. Collision sounds
        2. Helicopter sounds
        3. Game theme song

1. Tasks
   1. Programming Tasks
      1. GUI
         1. In Scope
            1. Screen Layout

This will be broken down into three sections:

Water in foreground which will be a small section at the bottom of the screen

Pier/boardwalk for runners

Large sky section at the top of the screen

* + - * 1. Textures

Helicopter sprite

Runner sprite

Crate sprite

* + - * 1. Fonts

Create font to display game time

* + - 1. Out of Scope
         1. Cloud sprites
         2. Bird sprites
         3. Sun sprites
         4. Create HUD that displays score, fuel gauge etc…
    1. Game Mechanics
       1. Movement

1. Scheduling
   1. Week 1
      1. Create screen sections:
         1. Sky
         2. Pier
         3. Side of pier
         4. Water
      2. Get runners to run across screen with animation
      3. Get helicopter to move back and forth
         1. Try and get a sprite sheet for animation
   2. Week 2
      1. Get a box to drop from the helicopter
         1. Using the current velocity of the helicopter calculate the trajectory of the box
      2. Detect collisions with runners
      3. Delete runner if collision
   3. Week 3
      1. Water reflection
      2. Helicopter exhaust particle effect
      3. Incorporate Microsoft menu screens
   4. Week 4
      1. Test current features
      2. Add additional features
      3. Test additional features.
   5. Week 5
      1. Finalization