Nick Hippen

Intro to Game Programming

Design Document

March 29, 2017

Sprite Game Design Document

# **Game Design**

I decided to create my sprite game as a top-down game because my group project will be a top-down game, and I wanted to get some experience with it. I found that it was essentially the same as side-scrolling games, however.

In the game, you control a mage character who is capable of moving up, down, left, and right. If you press ‘Spacebar’, a spell will appear traveling across the middle of the screen. If it hits the player, the player will die, changing into a death animation, making the one additional “ability” be death. The player can be revived by pressing ‘R’. The map contains rocks around the map to serve as a border as well as two trees to further demonstrate bounding boxes. The spell contains the circle bounding shape. Bounding boxes can be displayed by pressing ‘B’.

# **edu.unomaha.nhippen.sprite.game**

This package only contains the base game java class.

## SpriteGame.java

This class contains all of the base logic to the game. On initialization it creates all of the sprites that will be used in the game (except spells, which are spawned in). The player can move with WASD. A “spell” can be spawned by pressing spacebar. Because the base image was large, I used scaling to bring the size down. The spell also rotates. If the spell hits the player, the player will die, and will change into the death animation frame. The death animation will be rotated appropriately based on the direction you are facing. You can reset your character by pressing ‘R’.

# **edu.unomaha.nhippen.sprite.sprites**

This package contains all of the sprite related classes. All of the classes are either the SpriteObject class itself subclasses of it, except for the Direction enum.

## ActorSprite.java

The actor sprite is the mage character the player controls. She can move around in 4 directions, each with different walking animations, and die. It is a sub-class of BoundingSprite. It also demonstrates multi-bounding shape functionality, with both outer and inner bounds.

## BoundingSprite.java

The bounding sprite is a sub-class of SpriteObject. It describes a sprite object that has bounds. The class can handle both outer and inner bounds. If it has no inner bounds, it will treat the outer bound as the bound for the object.

## Direction.java

This is just a simple enumerator that describes 4 directions: north, south, east, and west.

## GrassSprite.java

This class is a sub-class of SpriteObject, as it has no hit detection. It does nothing special, it simple handles the grass sprite loading within the class.

## RockSprite.java

This class is a sub-class of BoundingSprite. It does nothing special, it simple handles the rock sprite loading within the class and sets up a square bounding object around the sprite.

## SpellSprite.java

This class is a sub-class of BoundingSprite. It does nothing special, it simple handles the rock sprite loading within the class and sets up a circle bounding object around the sprite.

## SpriteObject.java

This class is the base class to all sprite objects. It allows for translation, rotation, and scaling. It uses AffineTransform to manipulate the sprites.

## TreeSprite.java

This class is a sub-class of BoundingSprite. It does nothing special, it simple handles the tree sprite loading within the class and sets up a square bounding object around the trunk of the tree sprite.

# **edu.unomaha.nhippen.sprite.util**

This package has essentially no changes at all, it just contains utility classes to be used.

## KeyboardInput.java

No changes

## Matrix3x3f.java

No changes

## RelativeMouseInput.java

No changes

## SimpleFramework.java

No changes

## Utility.java

No changes, besides the addition of the intersection logic provided on the powerpoints.

# **edu.unomaha.nhippen.sprite.vectors**

This package contains all of the vector objects. I used the vector objects to implement my bounding shapes.

## AxisAlignedBoundingBox.java

This class is a sub-class of VectorObject. It describes an object to be used as an axis aligned bounding box. It holds 2 points, for min and max.

## BoundingCircle.java

This class is a sub-class of VectorObject. It describes an object to be used as a bounding circle.

## Drawable.java

No changes

## Vector2f.java

No changes

## VectorObject.java

No changes