VGP230 2D Games Final Project

2012

Blastoids

Game Design Document

Daniel Schenker

Copyright Daniel Schenker

# Revision History

|  |  |
| --- | --- |
| Date | Details |
| 2012-10-03 | Created GDD. |
| 2012-10-04 | Restructured GDD.  Added Table of Contents.  Added References. |
| 2012-10-05 | Edited Input.  Added Weapons to Objects list. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Contents

[Revision History 2](#_Toc337164534)

[Summary 4](#_Toc337164535)

[Document Purpose 4](#_Toc337164536)

[Concept 4](#_Toc337164537)

[Game Play 5](#_Toc337164538)

[Overview 5](#_Toc337164539)

[*List of abilities:* 5](#_Toc337164540)

[Objects 7](#_Toc337164541)

[Character's Space Ship 7](#_Toc337164542)

[Weapons 7](#_Toc337164543)

[Components 8](#_Toc337164544)

[Game Engine 8](#_Toc337164545)

[Physics 8](#_Toc337164546)

[Graphics 8](#_Toc337164547)

[Control 8](#_Toc337164548)

[Input 8](#_Toc337164549)

[Default Controls 8](#_Toc337164550)

[Artificial Intelligence 9](#_Toc337164551)

[References 10](#_Toc337164552)

[Task List and Schedule 11](#_Toc337164553)

# Summary

## Document Purpose

This document is designed to layout and explain the plans for the game.

All of the features listed here may or may not be implemented in the final version, but are kept here for reference as potential features.

This includes the game design features, as well as technical plans and an overall guideline as to how it will be developed in when coded.

This Game Design Document is subject to change, but the goal is to have as much of the planning done through the use of this document before actual coding of the game begins.

## Concept

Blastoids is a 2D space shooter arcade style game with a heavy emphasis on the player attempting to clear a course as fast as possible, with the added ability of being able to freeze time by creating combo kills.

The game has a colourful art style, which is reminiscent of a *Geometry Wars: Galaxies*1 style, but with many more fireworks particle effects.

# Game Play

## Overview

* You control a space ship from a 2D aerial view in which the space ship can move in any direction, and shoot in any direction regardless of which way the ship is facing.
* The goal is to complete missions by reaching your destination and destroying the boss.
* Levels have two main components, which allow for two different methods of traversal.
  + In the first method, the player navigates through a maze, while either attacking or evading enemies along the way.
  + In the second method, the player enters a battle zone, and cannot leave until they have accumulated enough energy to blast a hole in the wall.
    - Energy is obtained by killing enemies.
    - A hole can be made wherever there are cracks in the wall.
    - Each battle zone has a countdown timer.
      * If the time hits 0, the player is killed.
      * The player receives a score reward proportional to how much time is left on the timer when the player escapes the battle zone.
* The playing field is surrounded by a prism beam containment field to allow for better ship control.
  + The playing field can be expanded wherever a blue hazy cosmic stardust is by a variety of methods, most notably by shooting the boundary with a powerful enough weapon.
  + The player cannot leave the playing field.

## List of abilities:

|  |  |
| --- | --- |
| Kamikaze2 | Temporarily increases the players maximum speed greater than the normal kMaximumVelocity, to kKamikazeVelocity.  The player takes no damage from normal enemies (not massive, bosses, etc) while in kamikaze mode.  The player can kamikaze through normal enemies, applying severe damage, and potentially fatal damage.  The player receives combo points for the amount of enemies that they applied damage to while in kamikaze mode. |
| Shield | The player can negate damage by shielding off enemy projectiles (but not the enemies’ main body).  If the player times the use of the shield ability perfectly, the player’s shield turns into a reflector, and all incoming projectiles are reflected with the same effect that was used in order to attack the player (e.g., a normal bullet reflects off, while a wall bouncing laser reflects off while maintaining its pre-collision bounce ability and bounce count, while a homing missle will then turn around and use its homing ability towards the original sender of the projectile).  Note: Not all projectiles can be shielded against, such as massive projectiles and others. |
| Homing Missiles3 | Explosive missiles which home in on enemy targets, but are easily shot out of the air. Upon contact with anything the missiles detonate immediately. |
| Place Turret4 | Places down a turret at the players location, which remains until it is either destroyed or the player leaves the level. |
| Spinfusor | A projectile disc with small splash damage that inherits the properties of the character’s velocity when fired. |
| Laser Bounce | A laser that bounces off of walls until the laser bounce count is depleted or the laser collides with a non-wall. |
| Splitter Slam | Fires a sticky grenade with limited range, that upon a successful stick, emits 5 smaller sticky grenades around itself, which then start a timer.  When the timer reaches 0, all the grenades explode.  If a grenade is shot, it detonates on the spot with a larger explosion than normal. |
| Frozen Chomp | The player’s ship turns into a hungry pacman like object, which bursts forward a short distance in the direction the player’s move joystick is facing at the time. The player “eats” all enemies (not massive, bosses, etc).  For every enemy the player eats, a sonic explosion occurs which freezes all enemies in a small radius.  If the player ate at least a single enemy on the last chomp burst, the player receives another chomp burst5, netting a higher combo score with each successful burst.  Example: (Assume all enemies for example have a base worth of 100 points).  The player chomps 1 enemy during the 1st chomp burst, 3 during the 2nd chomp burst, and none during the 3rd chomp burst.  The player receives (1x100) + (2)(3x100) + (3)(0) = 700 points. The (3)(0) is not included in the formula, but is written in the example to show that no points are received during a failed chomp burst. |
| Lightning Arc | The player’s ship shoots an electrical beam forward which randomly splits into multiple branching paths. Enemies hit by the lightning arc get their controls scrambled temporarily, as well as receive damage over time for the duration that they are within the electric field.  Casting the Lightning Arc disables all control for the player for a very short period of time. |

## Objects

### Character's Space Ship

* Center of Mass
  + World coordinate vector
* Collision Volume
  + Each vertex of the object is represented by a vector local to the center of mass.
* Angle in relation to world coordinates
* Velocity
  + velocity
  + kMaximumVelocity
  + kKamikazeVelocity
* Acceleration
  + acceleration
  + kMaxAcceleration
  + kKamikazeAcceleration
* Mass
* Texture
  + The scale of the texture is determined by the wingspan of the ship (preferably have the texture match the collision volume perfectly)
  + The angle (direction facing) of the texture is the same as the angle used in the physics calculations
* Weapons

### Weapons

* Name
* Damage
* Velocity
  + mVelocityMinimum
  + mVelocityMaximum
* Acceleration
  + mAccelerationMinimum
  + mAccelerationMaximum
* Fire Time
* Reload Time
* Ammo Type
* Ammo Count
* Animation
  + Creation
  + Motion
  + Destruction
  + Dissipation

# Components

## Game Engine

The game engine used will be SGE (Simple Game Engine) by Peter Chan, which is based off of Microsoft’s DirectX9 version June 2010.

## Physics

A rigid-body physics system will be used.

The game will have multiple layers, but will only use two dimensional mathematics

Objects in separate depths, will be mathematically calculated as if they are in the same depth, meaning that they will only have the appearance of being in three dimensions graphically.

## Graphics

As many of the graphics as possible will be created through mathematical programming art.

Textures created using Microsoft Paint or Paint.net.

Graphics can be imported from external sources such as the internet or artists if needed.

## Control

### Input

* Game Pad (testing done with PlayStation 3 DualShock controller)
* PlayStation Move (including Navigation controller)
* Keyboard and Mouse

### Default Controls

|  |  |  |  |
| --- | --- | --- | --- |
|  | Game Pad | PlayStation Move | Keyboard and Mouse |
| Move | Left Joystick | (Left) Joystick | WASD |
| Aim | Right Joystick | LED Aiming Reticle | Mouse Cursor |
| Fire | Right Shoulder Button | T Button (Right Trigger) | Left Mouse Button |
| Ability | Left Shoulder Button | L2 Button (Left Trigger) | Spacebar |

Note: Controls can be remapped.

#### Move

Game Pad and PlayStation Move

Movement is calculated by the force applied due to thrust.

Thrust is a force which is an applied vector based off of the input from the joystick.

The direction vector is determined by

Magnitude: tilt \* kMaxAcceleration

tilt: Tilt of Joystick, 0 = no tilt, 1 = full tilt

Direction: 360 degrees direction that joystick is facing.

If the joystick is centered, the direction remains as previous.

Keyboard and Mouse

Movement is calculated by the force applied due to thrust.

Thrust is a force which is an applied vector based off one of the input from the arrow keys.

The direction vector is determined by

Magnitude: 0 or 1

Direction: One of eight 45 degree angles created by a combination of the arrow keys that are pressed.

#### Aim

Game Pad

Aim is calculated by the angle in radians' that the joystick is facing.

If the joystick is centered, the direction remains as previous.

'*Radians are used because calling inverse cosine in the dot product is inefficient. Radians also provide us with all the information that we need*.

PlayStation Move

Aim is calculated by a vector from the turret of the ship to the location of the pointer reticle on the screen, which is can be represented by a reticle and/or a laser.

Keyboard and Mouse

Aim is calculated by a vector from the turret of the ship to the location of the mouse cursor on the screen, which is can be represented by a reticle and/or a laser.

#### Fire

Game Pad, PlayStation Move or Keyboard and Mouse

Fires the default weapon, which has infinite ammunition.

Since the button can be held down, the weapon fires continuously without reloading.

#### Ability

Game Pad, PlayStation Move or Keyboard and Mouse

The ability button is a single button, but use of the ability and how the use is reflected as a control mechanism is ability dependent.

## Artificial Intelligence

# References

1. *Geometry Wars: Galaxies* is a multidirectional shooter video game created by Bizarre Creations and Kuju Entertainment, which was released on Nintendo's Wii and DS consoles in November 2007.

The soundtrack was composed by Chris Chudley from Audioantics who created the music for all of the Geometry Wars series.

<http://en.wikipedia.org/wiki/Geometry_Wars:_Galaxies>

[http://web.archive.org/web/20101213144851/http://bizarrecreations.com/games/geometry\_wars\_galaxies/](http://web.archive.org/web/20101213144851/http:/bizarrecreations.com/games/geometry_wars_galaxies/)

1. The kamikaze concept and movement/control style is inspired by the boost ability in *Super Stardust HD* developed by the Finnish company Housemarque for the PlayStation 3.

<http://en.wikipedia.org/wiki/Super_Stardust_HD>

<http://www.housemarque.com/games/super-stardust-hd/>

1. The Homing Missile ability is inspired by Samus’ tilt side-B special attack in *Super Smash Bros. Melee* developed by HAL Laboratory for the Nintendo GameCube.

<http://en.wikipedia.org/wiki/Super_Smash_Bros._Melee>

1. The continuous chomp eater concept and movement is inspired by the Fireball Frenzy mode in *PixelJunk Racers* developed by Q-Games for the PlayStation 3.

Three player video of Fireball Frenzy mode from *PixelJunk Racers*: <http://www.youtube.com/watch?v=QBFE9uQQg64>  
Explanation of Fireball Frenzy mode from the official Q-Games PixelJunk website:

<http://pixeljunk.jp/library/Racers/modes.php?page=modes04>

<http://en.wikipedia.org/wiki/PixelJunk_Racers>

# Task List and Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Estimated Time | Actual Time | Task | Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |