Last Rhino

Game Design Document

##### Copyright notice / author information / boring legal stuff nobody likes

PLEASE NOTE:

You can copy this into your Drive by going to **“File” >> “Make a copy…”**

You do not need to request permission to edit this document if you make a private copy.

##### 

## *Index* Index

1. [Index](#kix.tn44c9jbklu6)
2. [Game Design](#kix.vh7mr3wp9gub)
   1. [Summary](#kix.y0z7a3kyjoqj)
   2. [Gameplay](#kix.q89snwy3vllo)
   3. [Mindset](#kix.3j0sjjj4v19l)
3. [Technical](#kix.nw2pvrargt85)
   1. [Screens](#kix.5baplgyo85eh)
   2. [Controls](#kix.4u55u65bonjw)
   3. [Mechanics](#kix.vrkne3rhovuh)
4. [Level Design](#kix.mun66rmbl34h)
   1. [Themes](#kix.feu141lpqdpj)
      1. Ambience
      2. Objects
         1. Falling Evil balls
   2. [Game Flow](#kix.lrp1xr74olld)
5. [Development](#kix.k64sfnylstdx)
   1. [Abstract Classes](#kix.fwz93u72nlt2)
   2. [Derived Classes](#kix.9mr6xhrqo7o4)
6. [Graphics](#kix.7fniby1a9lf0)
   1. [Style Attributes](#kix.3ui65x9w2r9w)
   2. [Graphics Needed](#kix.cdq39znqma9y)
7. [Sounds/Music](#kix.p0zdtpym1l8)
   1. [Style Attributes](#kix.g829e5h0mi8e)
   2. [Sounds Needed](#kix.1k2k1v9771k6)
   3. [Music Needed](#kix.coz0pe9ii0ek)
8. [Schedule](#kix.ur637w5onhwq)

Game Design

## Summary

You are the last living rhino. Evade the falling balls survive as long as you can.

## Gameplay

Evade the falling balls survive as long as you can. You can dash with the space button

## Mindset

Player should drink a lot of coffee and try his best to evade the falling evil balls of death

Technical

## Screens

1. Title Screen
   1. Start button
2. One level

*(example)*

## Controls

Press space button to dash

## Mechanics

No gravity is used here only spawn mechanism and angle rotations

Level Design

*(Note : These sections can safely be skipped if they’re not relevant, or you’d rather go about it another way. For most games, at least one of them should be useful. But I’ll understand if you don’t want to use them. It’ll only hurt my feelings a little bit.)*

## Themes

1. Mount Everest
   1. Mood
      1. Light,Bright and Colorfull
   2. Objects

*I Falling Evil Balls*

* + - 1. Club ball happy
      2. Angry razor ball
      3. Laughing guy ball

*(example)*

## Game Flow

1. Player starts near mount everest
2. Move right and left
3. Falling evil balls try to get you
4. You have 3 lifes
5. Balls have different damage so watch out

Development

## Derived Classes / Component Compositions

1. BasePlayer
   1. PlayerMain
2. BaseEnemy
   1. EnemyAngry
   2. EnemyHappy
   3. EnemyGuy
3. BaseObject
   1. ObjectHeart
   2. ObjectMountEverest
   3. ObjectSky
   4. ObjectGround

Graphics

## Hand drawn Graphics

## Graphics Needed

1. Characters
   1. Ball-like
      1. Angry-ball
      2. Happy-ball
      3. Guy-ball
   2. Player
      1. Rhino (idle, running)
2. Blocks
   1. Ground
   2. Sky
   3. Mount Everest
3. Ambient
   1. Grass
4. Other
   1. Heart icon

Sounds/Music

## Style Attributes

Again, consistency is key. Define that consistency here. What kind of instruments do you want to use in your music? Any particular tempo, key? Influences, genre? Mood?

Stylistically, what kind of sound effects are you looking for? Do you want to exaggerate actions with lengthy, cartoony sounds (e.g. mario’s jump), or use just enough to let the player know something happened (e.g. mega man’s landing)? Going for realism? You can use the music style as a bit of a reference too.  
   
 Remember, auditory feedback should stand out from the music and other sound effects so the player hears it well. Volume, panning, and frequency/pitch are all important aspects to consider in both music *and* sounds - so plan accordingly!

## Sounds Needed

1. Effects
   1. Falling balls sound 3 different
2. Feedback
   1. “mooo!” (death)
   2. Shocked “moomoo!” (attacked)

*(example)*

## Music Needed

1. Whacky
2. Exciting “flute” track

Schedule

1. develop base classes
   1. base entity
      1. base player
      2. base enemy
      3. base block
   2. base app state
      1. game world
      2. menu world
2. develop player and basic block classes
   1. physics / collisions
3. find some smooth controls/physics
4. develop other derived classes
   1. blocks
      1. moving
      2. falling
      3. breaking
      4. cloud
   2. enemies
      1. soldier
      2. rat
      3. etc.
5. design levels
   1. introduce motion/jumping
   2. introduce throwing
   3. mind the pacing, let the player play between lessons
6. design sounds
7. design music