631 Running Rhino Game Design

By: Running Rhino Team
CSC 631: Multiplayer Game Design and Development
San Francisco State University

Team Members

Julian Refsland:	Server
Joshua Rubin	Server, Content
Dennis Chung	Database, Concept
George Jone	Client
Jan Burianek	Client
Xingya Li	UI
Nicu Listana	Integration, Team Lead

Overview

Running Rhino's concept is comparable to a game called Fun Run (about 10m~50m downloads on Android). Our team wanted to apply the proven game mechanics from Fun Run in order to make it more fun to play and attractive to World of Balance players.

Our focus this semester is to make the gameplay very "sticky". In other words, we want every game to be replayed many times. The game shall take less than 1 minute and 30 seconds to play. We are doing this because our main target is the mobile device, where users have short attention span when compared to other platforms that focus on immersion such as PC or consoles.

In order to accomplish stickiness, we wanted to implement a couple of changes in the gameplay such as making the game control's 1 button, where users can simply tap the screen (or press the spacebar for PC.) to jump and progress through the level. The player does not need to control the way they move, since every player will automatically move towards the finish line as the game starts.

A second way to accomplish stickiness, we want to make room creation really easy for users. We wanted to abstract away game's lobby from the main World of Balance lobby, and create its own Running Rhino lobby. The game shall have its own matchmaking engine where users who decide to play a game join an automatically generated Running Rhino lobby where other users can join to play. Once 4 users join the Running Rhino lobby, they can start the race. Once the race is over, the game will go back to the lobby. If the lobby still has 4 players, the game will start another race, otherwise it will wait for another user.

Finally, stickiness can be achieved by moving forward with what Running Rhino is really good at--being a mini game. Compared to other games in the World of Balance, Running Rhino offers a small experience in order to provide players instant gratification by playing many games at a small span of time.This instant gratification can be achieved by polishing the gameplay even more, by adding powerups that users can use to their advantage, or prevent other players from winning the race, and by rewarding the players medals for each time they place first, second, or third in the race.

High level game design

Types of animals*

Name	Description	Prey	Predator
Southern Eland	a savannah and plains antelope found in East and Southern Africa.	Fruits	Leopard
Kirk's Dik-dik	a small antelope found in eastern and southwestern Africa.	Fruits	Leopard
Kori Buskard	a large bird native to Africa.	Fruits	Leopard
Tree Mouse	a poorly understood climbing mouse from Central Africa.	Fruits	Leopard

*note: We wanted to focus on species on the middle of the food chain in order to make the game more balanced with predator and preys.

Random box power-up contents

Name	Description	Benefit (+/-)
Leopard	A prime predator on top of the food chain	Stops all animals who pass by it Spd = 0
Fruits	Nature's delicacy, favorable to most animals	Stops all animals who pass by it. Spd = 0
Muddy field	Muddy areas can slow down animals who are moving	Slows down all animals who pass by it: Spd -5

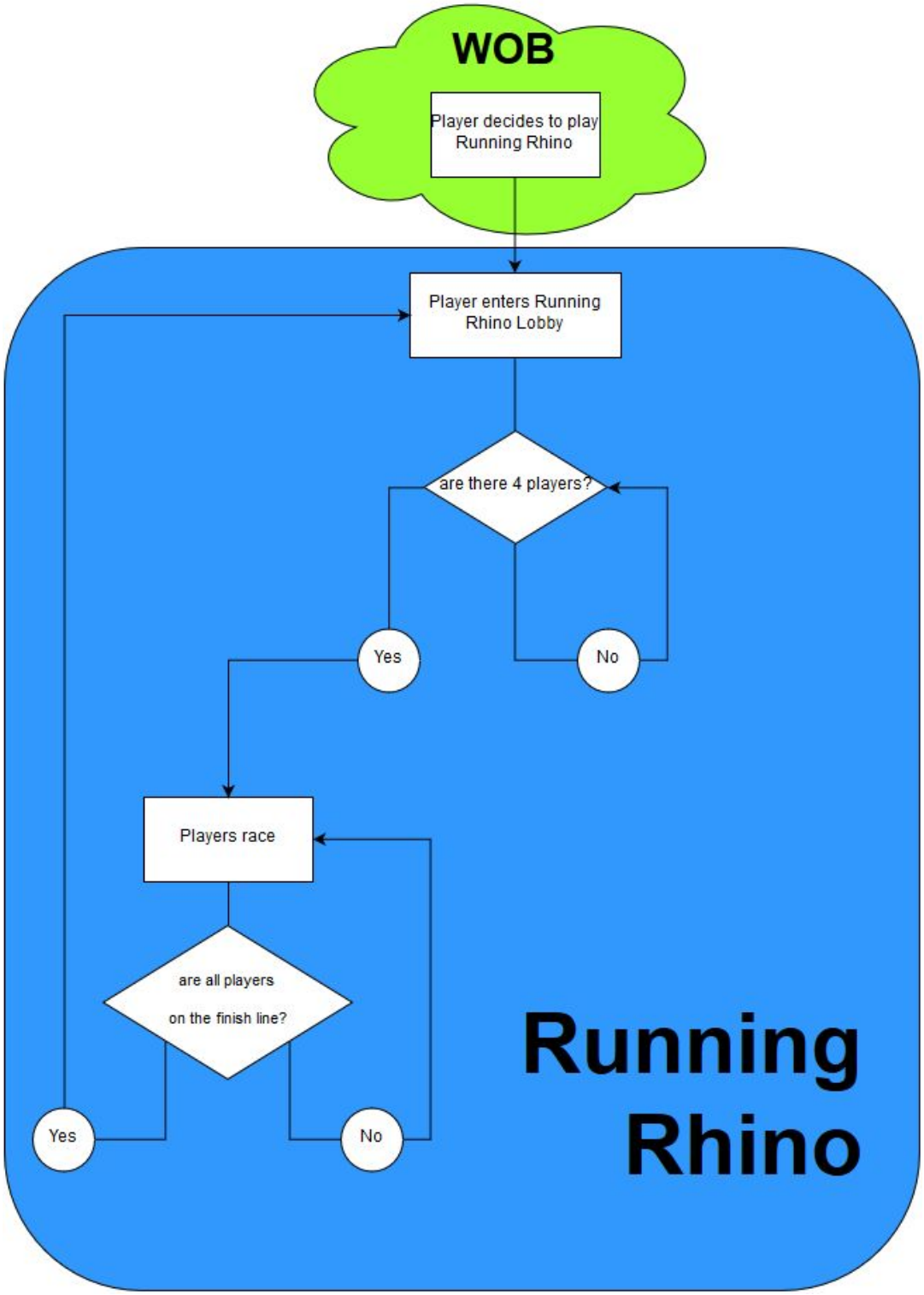
Jungle traps	Traps set up by hunters to catch animals	Stops the player who steps in front of it by 1 second. Spd = 0 for 1s
Monsoon	Climate change can prevent animals from survival	Stops the players from moving besides the player who got the powerup Spd = 0
Hunter	Hunts down the animal right in front of you	Stops the player in front of the user for 1.5 seconds Spd = 0 for 1.5s

Level Design

Level locations:

Name	Description	Location
Grassy Savannah	An area filled with grass that is home to many types of animals. Weather is usually tropical, and can be wet. Can be occasionally visited by hunters looking for animals to hunt and have monsoons.	Africa

Game client core-loop



Implementation timeline

Week 1, Concept presentation

- Talk about the game design concep
- Propose changes

Week 2, Familiarizing with code

- Clone github repository
- Read code, for each part ie: server, client
- Run the client using localhost or remote server
    - DB: local MySQL, libra, or smurf
    - Server: localhost over local network, remote server
- Take notes on items that need to be implemented, and where to implement them
- Prepare for midterm interview

Week 3, Debugging last semester's code

- Test code, then debug, then test again
- Ensure quality and code standards

Week 4, Code implementation: player controller, global currency, lobby

- Client:
    - Write character controller code to make it 1 button
    - Write parallax background code
- Server:
    - Integrate with global currency
    - Make the server handle 4 concurrent clients
- Database:
    - Save global currency in the database
    - Create Data Access Object to access read,write, etc. for global currency
- UI
    - Implement the Running Rhino lobby
- Integration
    - Handle merge conflicts
    - Monitor pull requests

Week 5, Code implementation: powerups, maps, lobby

- Client:
    - Re-design the map to make it 1 button friendly
    - Implement powerups
- Server:

        - Implement server-side powerups
- Database:
    - Help UI with the lobby
- UI
    - Finish the lobby
- Integration
    - Handle merge conflicts
    - Monitor pull requests

Week 6, Testing and Debugging

- Client:
    - Debug character controller code
- Server:
    - Debug global currency code
    - Debug lobby code
- Database:
    - Debug global currency DAO
    - Help test lobby code
- UI
    - Debug lobby code
- Integration
    - Test powerup code
    - Test lobby code
    - Test powerup code
    - Handle merge conflicts
    - Monitor pull requests

Week 7, Testing and Debugging pt. 2

- Client:
    - Test and refine the map with multiple clients
- Server:
    - Test lobby code
    - Test powerup code
- Database:
    - Help test lobby code
    - Help test lobby code
- UI
    - Test lobby code
- Integration
    - Test powerup code
    - Test lobby code
    - Test powerup code
    - Handle merge conflicts

    - Monitor pull requests

Week 8 - Refining and final touches

- All:
    - Prepare a testable level for the launch party
    - Testable using local db and server
    - Refine the level
    - Final debugging