RMIT INTERNATIONAL UNIVERSITY VIETNAM

ISYS2102 – Software Engineering 2

Instructor: Kevin Jackson

GAME DESIGN DOCUMENT

Project: ZOO-A1

Game name: Angry Park

Team name: Skoorti

Nguyen Phuong Duy – s3258157

Le Anh Duy – s3298336

Nguyen Quoc Hoang Tan – s3260657

Semester A - 2012

1. Game description

1.1. Audience

Angry Park is design for everyone above 6 years old.

1.2. Technology

- Operating System: Fedora Linux 16, Windows 7, MacOS X
- Programming language: Java - version 7
- IDE: Netbeans 7.1
- Version control: Git
- Framework: Spring
- Design tools: Photoshop, Gimp

1.3. Variables

1.3.1. Difficulty levels

Easy, Medium and Hard

1.3.2. Animals

There are 2 types of animals in the zoo:

- Natural animals: the same as those in normal zoo, such as elephant, lion, giraffe, etc.
- Mutant animals: super-powered animals that can break their cage and attack visitors. The higher level the scientist, the more natural the mutant animals are produced. That means if the scientists are well-qualified, they can produce animals with lower risk of attacking people.

1.3.3. Types of visitors

- Individual: visitor comes to the zoo and hangs around to visit animals
- Group: visitors come in group and always go together around the zoo
- VIP: visitors with more budget than normal visitors and higher speed of moving so that they can escape from the animals.

1.3.4. Types of staff

Zoo keepers, cleaners, vegetarians, scientists, security guards, clowns. Each of them has specific qualification, which determines the efficiency of their work.

1.3.5. Police and taxation

At any time, if there is some visitor killed by animals inside the zoo, player must pay a certain amount of tax based on difficulty level. Otherwise he will get caught by police and the game ends.

1.4. Building options

- Cages/tanks: There are various types of cages with different price levels. An animal only matches with a certain group of cages. For example, fishes must be raised in water tanks, while cage of elephants is not the same as that of birds.
- Restaurants: Hanging around the zoo, visitors are able to stop at any restaurant opened in the zoo to eat and drink, depending on their hungry and thirsty level. There are different choices for food and drink with a variety of prices.
- Toilets: Visitors have bladder level, which requires them to enter the toilet if it goes down to a certain level. A certain amount of money is charged whenever visitors enter a toilet.
- Car park: Player is allowed to buy car park in order to attract visitors in group.
- Building promotion: Buildings can be promoted to upper levels if they reach a certain number of views. Building promotion costs user a predefined amount of money.
- Selling buildings: There is an option to sell some buildings if player is running out of money. The remaining value of buildings will be determined based on the time it has been built.

1.5. Technical design

Map: The game map is divided into units called “tile”. Each tile will be represented by a certain type of building. Tile in map is also one unit in the x-y coordinate, which is stored in 2-dimentional array.

Visitors: There are several properties in class Person to express person’s personalities: age, gender, moods (satisfaction, thirsty, hungry and bladder). Each person is managed by a thread, thus his/her moods are easily controlled by the game AI.

Animals: Each animal is also managed in one thread so that all of its activities are separated from others’ and not dependent on any procedure. Many threads of animals are run in parallel.

1.6. User interface

Angry Park is built on 2D graphics and simulated in 3D.

Background music would make the game more attractive to the player.

1.7. Gameplay

Before starting the game, player is required to set target money. If the budget goes up to that goal, player wins the game.

The player loses when he runs out of money to pay the staff, animal food and tax.

2. Product backlog

ID	Description
Sprint 1: Simple game engine	
1	Set up development tools, version control, team environment and communication
2	Create initial game design and mechanism
3	Design the game technically based on game design (class diagram)
4	Find suitable images for game objects (map, buildings, people, animals, …)
5	Generate 3 maps with different terrains in 3 places: the US, Africa and Asia
6	Design different kinds of people with personalities (visitors, staff)
7	Design different kinds of buildings (cages, restaurants, toilets…)
8	Design animals with various features
9	Implement the control panel displaying budget, score, statuses of the zoo …
10	Auto-generate visitors after a random period of time (individual and VIP)
11	Implement AI motion for visitor
12	Implement placing buildings on the map (except car park)
13	Implement buying animals
14	Implement hiring different types of staff (clowns, scientists, zoo keeper)
15	Implement buying food and feeding animals
16	Implement visitors can stop and watch animals – changing their satisfaction level
17	Implement animals can escape from their cages and kill visitors by erupting fire
18	Auto-generate new mutant animals produced by scientists
19	Implement 2 levels: easy and hard
20	Implement visitors buy food and drink in restaurant and go to the toilet
21	Implement visitors get out of the zoo (out of money, low level of satisfaction)
22	Implement win and lose conditions
Sprint 2: More advanced features	
23	Implement selling staff, buildings and animals
24	Implement building promotion (getting more income)
25	Implement placing car park on the map
26	Implement group of visitors
27	Implement water fishes with suitable tanks
28	Implement qualification of staff
29	Implement interaction between visitors and clowns
30	Implement animals could be sick and need help from vegetarian
31	Implement medium level
32	Implement random map generation
33	Improve the AI motion of visitors
34	Implement different interaction between visitors and animals when being chased
35	Implement saving and loading game
36	Implement salary promotion request by staff