Developing Games in Alice using Workflow

M. Dorooodchi
Cardinal Stritch University
Milwaukee, WI, USA
mdorooodchi@stritch.edu

A. Roudaki
Multimedia University
Kuala Lumpur, Malaysia
roudaky@gmail.com

ABSTRACT
Alice is a user-friendly software tool for teaching concepts of programming by developing games or animations. The pre-programming analysis, however, relies mainly on storyboards. In this work, workflow analysis is used as a complementary method to the analysis part before developing the actual game. Workflow can help the programmer visualize the whole process easier. In other words, workflow makes the analysis and development phase shorter and easier for the developer.

Categories and Subject Descriptors
D.2.2 [Design Tools and Techniques]: Design software using workflow approach

General Terms
Game development, workflow

Keywords
Workflow, Game Programming, Alice

1. What is Workflow?
Workflow is a word that means different things to different people. According to [1], workflow is the computerized facilitation or automation of a business process, in whole or part. Workflow is a reliably repeatable pattern of activities enabled by a systematic organization of resources, defined rules and mass, energy and information flows, into a work process that can be documented and learned. Workflows are always designed to achieve processing intents of some sort, such as physical transformation, service provision, or information processing. Three types of workflow can be distinguished: Sequential, State-Machine, Rules-based [1].

2. Software as Process
A process defines the way activities are organized, managed, measured, supported and improved to reach a goal. It has been shown that processes can also be called software or vice versa [2]. Therefore, computer games also can be considered as processes, and consequently can be modeled using workflow. In fact, workflow analysis makes the visualization of the game process very simple to understand.

3. Development Steps
Alice has been used for teaching computer concepts for the past several years. Its visual interface makes the process of software development much less tedious especially among younger generations. Using workflow with Alice can makes it easier to understand the game development process. There are several steps we should accomplish when we want to create games or animations in Alice based on workflow analysis:

1. Define the game goals, main objects, and scenarios.
2. Perform the workflow analysis on the scenarios, and create the workflow diagram of the game.
3. Create the objects in the scene.
4. Develop the code using workflow diagram.
5. Test the software.

The above steps are explained in details in the following sections using some games as examples.

4. Workflow Analysis
Workflow analysis facilitates the development of the game process. In general, the game is composed of several processes focusing on different objects and aspects of the game. Therefore, the first step after defining the goals of the game is to identify the required processes to achieve these goals. In other words, the game scenario is specified by different processes that in turn contain several activities. Furthermore, each activity can also be identified as another process, and consequently can be broken down into more specific activities if needed.

In our analysis, we first think of the game as one big process with one activity, and then break it into more activities. We call this the expansion of an activity. To expand an activity, we start from the entry points. An entry point is the event which activates or triggers a process. In most cases, entry points are the results of user interactions with the game. In addition, we could have some automatically activated entry points for example by a timer. After identifying the entry points, the activities that are going to be executed should all be listed and sketched in a workflow diagram.

In this way, for every scenario, the workflow model is formed. Later, these individual workflows are combined together based on their interaction and synchronization mechanisms. In other words, the overall game workflow will be formed from these individual workflows. To show how the workflow based game design works, we look at the following examples.

5. Car Dart Game Development
5.1 Game Scenario
In this game, the player controls a car. The main goal is to let the player drive the car into a wall of crates and knock the driver out of the front windshield. The driver flies toward a giant dartboard, and gets stuck in it. After three tries the game ends, and the score
will be shown. The game includes several objects including car, driver, dart, and road. At the beginning of the game, the car is controlled by the computer, but after the first curve the player is allowed to control the car. In this game the camera is connected to the car and car moves on the road. A sample of this game is available from the “alice.org”’s official forum. This game is redesigned using the proposed workflow analysis.

The game starts after counting down from 3 to 1. Figure 1 illustrates the start of the game. Figure 2 shows the car after the first curve, when the player can control the car. In the picture a wall and a giant board is visible. Figure 3 displays the driver after he has flown to the board and is landed on the board.

5.2 Workflow Analysis
After the game scenario is defined, our workflow analysis starts by identifying the main processes needed. The entry points which activate each process are to be determined next. This game includes the following processes:

1. Move Car Forward
2. Move Car Right/Left
3. Driver flies out of the car

These processes can be invoked by the player or when certain event occurs.

In addition, the following three entry points can also be identified:

1. Game Starts
2. Player presses right arrow key on the keyboard
3. Player presses left arrow key on the keyboard

Now, for each of these entry points we can define a list of activities. For example, after player presses the right arrow key, the car should be moved slightly to the right. After defining workflow needed for each of the entry points, we have to combine all of them together. In order to do that, we have to look at the overall game process and design the workflow diagram to include the entry points whenever they have to be available. In this example, the player can either press the right or left button after the car turns at the first curve. The car moves forward toward the end of the road.

Workflow diagram for this game is illustrated in figure 4. It includes two branches; the left branch which shows the entry point for pressing left and right buttons is to move the car to left and right, and the right branch which shows the car movement...
from the beginning to the end when the car reaches the end of the road. This is repeated until there is no driver left to continue the game. Figures 8 and 9 depict the actual code and the way it corresponds to the workflow.

6. Race Ball Game Development

6.1 Game Scenario

This is a racing game in which a ball rolls around the race course. The ball moves very smoothly and is controlled by the arrow keys. If the ball gets close to the walls, it will stop. There is a defined speed limit. This game was also downloaded originally from Alice official forum. Similar to the previous example, we are redesigning the game using the workflow analysis.

This game has a circular field and ball moves on this field. Figure 5 shows the ball close to a boost square while figure 6 illustrates the start of the game.

![Figure 5- Race Ball game](image1.png)

![Figure 6- Race Ball Game](image2.png)

Workflow diagram of this game is shown in figure 7. In this game, the arrow keys are constantly monitored. Upon pressing any of them, the related process will be activated. As the ball moves, its position is also monitored against the walls.

6.2 Workflow Analysis

Our workflow analysis, similar to the previous example, starts by determining the entry points as listed below.

1. Game starts
2. Player presses the Right Button
3. Player presses the Left Button
4. Player presses the Up Button
5. Player presses the Down Button

At the entry point number 1, when the game starts we have two branches. These branches are two parallel workflows. One of them is for checking the entry points number 2 through 5 which are responsible for user movement and setting the movement parameters based on that. The other one leads to the activities needed to be done continuously in order to move the sphere in the world. At this branch several activities are occurring; 1) the speed based on the Forward/Backward parameter is set, 2) the sphere is placed based on Right/Left parameter, and 3) the sphere is moved based on the calculated speed. The camera is moved as the sphere moves. Figures 10 and 11 show how the workflow is mapped to the code inside Alice environment.

7. Results

Our experience indicates that workflow can make it very easy for game developer to understand the game process. Therefore, the entire game analysis, design, and development would be simpler. In addition, other than learning the concept of game development and software development, the learner also learns about workflow
analysis which is very useful in understanding business processes. The developer can also easily recognize the details of game development during the workflow analysis. One of the main reasons for this is because workflow is a visual diagram.

8. References

Figure 8- Mapping the workflow for the Car Dart Game to the implementation in Alice.
Figure 9 - Mapping the entry points to the implementation for the Car Dart Game.

Figure 10 - Mapping the entry points to the Alice implementation for Race Ball Game.
Figure 11- Mapping workflow to the implementation for Race Ball Game.