

Integrated Pathfinding and Player Analysis for Touch-Driven Games

Tommy Thompson and Michele Vinciguerra

University of Derby
Derby, United Kingdom

Abstract

This paper introduces a software framework, currently in prototype stage, designed to permit automatic integration of pathfinding navigation in isometric camera-driven games. This framework is built with an emphasis towards touchscreen-driven gameplay on tablets and smartphones. This enables pathfinding to be adopted not by characters in-game, but also permits a range of traditional and experimental control schemes to utilise this feature. In addition, the framework provides for a range of metrics to be established based on user activity; enabling quality assurance to be conducted based on raw-user input and pre-processed data.

Introduction

The work detailed in this paper is an extension of previous research documented in (Maycock and Thompson 2013). In this body of work a range of touch controls were developed that enable the *Ms. Pac-Man vs. Ghosts* framework (Rohlfshagen, Robles, and Lucas 2011), to be playable on an Android device. The purpose of this study was to assess the impact that touch controls would have on players enjoyment, addressing a number of key issues when adopting touchscreens in games:

Maintaining Tacit Controls: The challenge of transferring traditional controls, such as joystick or directional pads (d-pads) to a touchscreen while ensuring it works as expected using a different interface.

The Lack of Tactility: Use of touchscreens and motion controls often removes tactile feedback from the control interface. Meaning the designer must consider how the game reinforces what impact users have upon the game world

Screen Occlusion: Players will often cover parts of the screen as they interact with the game. It becomes important to ensure control schemes do not interfere with the gameplay.

Action Translation: Often actions that are translated from gamepad/joystick to game do not translate well to touch

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

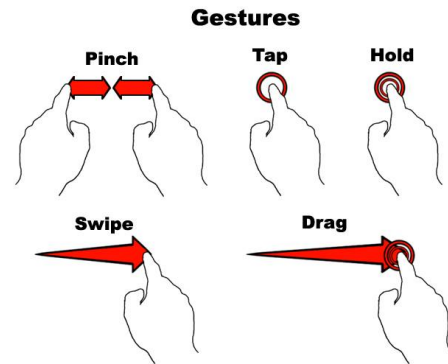


Figure 1: A collection of gestures commonly exhibited in touchscreen applications.

devices (e.g. swiping to change direction). As such, alternate offerings should be provided that change how users approach the game, while maintaining the level of challenge. We adopted pathfinding algorithm such as A* as an alternative, allowing players to tap the screen to pathfind towards a location.

This study found not only that players were receptive to non-traditional forms of control that adopted pathfinding, but that it changed how players perceived the game and the ways in which they could play it. In this paper, we discuss preliminary work by the authors in creating a plugin for the popular game engine *Unity3D* which not only provides the touch mechanics, but also largely automates the pathfinding integration into games reliant upon isometric cameras for use in the touch controls. In addition, in order to enable this work to be assessed on a far broader scale, we provide extensive set of features to log user activity for data mining purposes.

Framework Overview

Our framework is developed using the popular game engine *Unity3D* to allow developers to integrate pathfinding with a suite of touch gestures shown in Figure 1. The framework can be seen in Figure 2, where and consists of three components: the Gesture Manager, Navigation System and Log System.

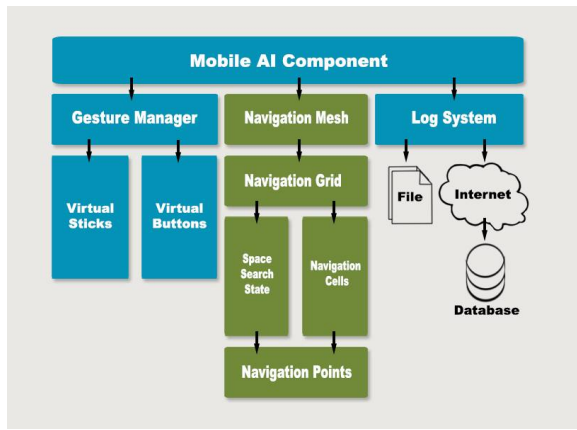


Figure 2: An overview of our Mobile AI framework. The gesture manager can recognise the gestures mentioned in Figure 1, apply navigation if required and log the interaction both locally and online.

Gesture Manager

The gesture manager provides ‘hooks’ that map each interaction shown in Figure 1 and provide delegates for developers to choose how to respond to a given gesture. These hooks ensure that developers need not consider whether the device in question is running on iOS, Android or Windows.

Each gesture is capable of drawing information to the screen to aid users in understanding how their input has been recognised. For example, the game may draw how the swipe has been recognised from the player. The Gesture Manager can utilise features in both the navigation and logging systems: allowing for interesting control-schemes to be built, as well as useful data gathering based upon how players use these controls.

Navigation System

The navigation system works by interfacing directly with the navigation mesh built into the *Unity3D* engine. In a given environment, we can build a quick pathfinding system that is not only useful for the Gesture Manager, but also agents controlling non-player characters (NPCs). The plugin builds a navigation grid on initialisation by conducting ray-traces against the map that is then stored in memory for future use.

The plugin provides access to all features of the navigation system to allow players to build their own pathfinding algorithms. At present, we provide an implementation of A* by default. Should players touch a given point in the grid, it will use the selected pathfinding algorithm to provide the fastest path using the navigation mesh. This allows us to quickly replicate touch-driven pathfinding established in (Maycock and Thompson 2013).

Designers can also establish which items in the world are factored by the pathfinding system. This allows for variations both in gameplay and difficulty. For example, the designer may opt for NPCs to be either included or ignored in the navigation search. This would naturally have an impact on the resulting gameplay.

Logging System

The logging system allows for information to be recorded about each play session as well as data on a per-gesture basis. This information is then stored in JSON format to allow for ease of use at a later date. Each hook within the Gesture Manager allows for data to be submitted to the logging system. This is with the intent of allowing flexibility on what information is recorded depending on the game being developed. For example: logging the frequency or success of gestures that are recognised by the game, or recording when players attempt to interact with the game in a manner that is not supported. This allows us to understand whether there is an issue with how existing controls are built, but also whether the existing controls match players expectations.

This logging system will retain this information locally and is easily accessible for developers. However, given that we expect this tool to be adopted in mobile game development, the framework also provides means to upload this data to a web server. Allowing for developers to continue to conduct player analysis after a game has been released into the wild.

Current & Future Work

The authors are conducting a preliminary investigation into the effectiveness of this framework using a clone of *Pac-Man* that reproduces the control methods originally adopted in (Maycock and Thompson 2013): a traditional D-pad controller, swipe gestures and touch methods that adopt A* pathfinding¹. This implementation is not an accurate reproduction of the original game. However it does retain the primary and secondary objectives of pill consumption and ghost evasion. Given we are no longer constrained by the *Ms. Pac-Man vs. Ghosts* framework, we can record metrics such as the frequency of gestures, whether they translate into useful gameplay actions and where the player touches the screen relative to the player. This tells us useful information about how the player interacts with the game.

We aim to summarise the results of this initial experiment once a sufficient number of candidates have tested the prototype. Having taken on board both feedback from the log system as well as qualitative feedback on the effectiveness of the touch controls, we will move towards building small yet complete games that are publically available via app stores. This carries the potential for a much larger case study to be conducted based upon a broader collection of participants.

References

- Maycock, S., and Thompson, T. 2013. Enhancing touch-driven navigation using informed search in *ms. pac-man*. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, 1–2. IEEE.
- Rohlfshagen, P.; Robles, D.; and Lucas, S. M. 2011. *Ms. pac-man vs ghosts league*.

¹A short video of this prototype in action can be found at <http://www.youtube.com/watch?v=q9EvB5qDIEQ>