Strategy Tactics: Final Report

Cal Pierog, Ryan Riegel, Paul Scott

May 13, 2004

1 Overview

Strategy Tactics is a tactical RPG in which the player must fight off hoards of belligerent peasants on a quest to reclaim some stolen source code. The game incorporates OpenGL graphics and several of the design principals we learned in class, which greatly accelerated the development process. The only risk we faced that proved to be too difficult for the scope of or project was animation; otherwise, our game functions essentially as envisioned.

2 Gameplay

2.1 General

The game occurs on rendered 3D battle sites containing the player's units and enemies. Play is broken into turns; every round, each unit gets to take two actions in a row, such as moving up to an enemy and attacking, or casting two spells. If the player manages to defeat all enemies, the player moves onto the next battle; if, instead, the enemies defeat the player, the game is over.

2.2 Game Elements

2.2.1 Battle Site

This the game's play field. It is essentially just a rectangular grid of tiles, which can vary in height. Height is significant because most unit actions (see below) are limited in how great a height difference they can target/affect. The site is drawn as a 3D model, which is related to the tile heights, but neither is dependent on the other; thus, very intricate models could be used for the site, even if the height map is simple.

2.2.2 Units

These are the knights, archers, etc. that participate in the battle. Each unit occupies exactly one tile and only one unit can occupy a tile at a time. Units are drawn as billboarded bitmaps and have different images for when they are alive or dead. Each unit has a list of actions (see below), from which the player or AI can choose what to do. Units have the following stats:

Stat	*Descrpition*
HP	How much damage the unit can take before dying
SP	How much magic the unit can cast before running out
Speed	How many actions a unit can take per round

Units also three attributes, Body, Mind, and Spirit. These affect how effective a unit is at casting a certain spell, as well as how effective spells are when cast upon the unit.

2.2.3 Actions

There are the true heart of Strategy Tactics, encompassing everything from movement to basic attacks to powerful spells. Each action has an associated SP cost (which can be 0); a unit cannot do an action unless it has sufficient SP. Also, each action has a targeting range, effect range, and effect type. These three are modeled as abstract classes with various subclasses for the different options as follows:

Range Class	*Description*
Move	Unobstructed path from start to all targets must exist
Blast	Targets all tiles between min and max range
All	Targets all units (or all units on one side)

Effect Class	*Description*
Move	Relocate the unit to the specified destination
Attack	Deal some amount of damage to target
Heal	Restore some amount of HP to target

There are a few other factors not listed above for brevity, such as the range's maximum height difference and the effect's associated attribute. A more specific example of a normal attack action might be as follows:

Targeting Range	`Blast(Min Range = 1, Max Range = 1, Max dHeight = 2)`
Effect Range	`Blast(Min Range = 0, Max Range = 0, Max dHeight = 0)`
Effect	`Attack(Base Damage = 10, Multiplier = 2, Type = Body, Resistable = false)`

This action's targeting range includes only the unit's adjacent tiles, but allows for some height difference. The effect range is only the targeted tile, and the effect is to deal `10 + 2 * Body` damage to the target.

2.2.4 Selectors

These are great pillars of light raining down from the heavens used to indicate which unit is in focus or which tile is targeted by an action. They track to the mouse's position when the player is issuing commands, making it easier to see what you're doing.

2.3 Player Controls

2.3.1 Camera Adjustment

The player can always adjust the camera's viewing angle by right-clicking on the game window and dragging. Normally, this causes the camera to rotate about some focus point, but holding SHIFT while right-click dragging will force the camera's position to remain unchanged and instead moves the focus point. Also, the player can adjust the camera's zoom with `+` and `-`, and can use `w`, `a`, `s`, `d`, `c`, and `v` (i.e., Half-Life movement controls) to adjust the camera's position.

2.3.2 Viewing Status and Terrain Information

The player can view information on a units and tiles by mousing over them and left-clicking. A Windows-style menu is displayed with submenus containing terrain information and unit status, if a unit is present.

2.3.3 Issuing Commands

The player can issue commands to the active unit (drawn with a yellow sphere floating above it's sprite) by mousing over it and left-clicking. In addition to the status and terrain information, a submenu containing the unit's actions is displayed. Selecting an action from this list will cause that action's targetable range to be displayed (targetable tiles are drawn with cyan spheres intersecting the ground and targetable units are tinted cyan). Mousing over tiles in the targetable range results in the action's area of effect to be displayed (affected tiles are drawn with red spheres floating a bit above the ground and affected units are tinted red). Left-clicking in the targetable range asks the user to either confirm or cancel his or her decision; confirm executes the action at the selected tile and cancel exits targeting mode.

2.3.4 Alternate Keyboard Controls

It is possible to issues commands using only the keyboard. The player can `i`, `j`, `k`, and `l` to move the tile selector, and can use the number keys to select the nth action in the unit's action list. Also, SPACE and ESCAPE can be used to confirm and cancel, respectively.

2.3.5 Debugging Controls

There are several commands left in by the developers for the purposes of collecting debug information. These are as follows:

Key	*Effect*
`z`	Toggle the display of the height map
`x`	Toggle the display of axes
`q`	Print out the camera's position and angles

2.4 Graphics

The game uses OpenGL to render the battle site, units, and various selectors. Some simple lighting is implemented, as are textures and some translucency. Graphics are borrowed from Xenogears and various online depositories.

2.5 Levels and Storyline

There are four battles in the game, each of which is introduced by a dialog. The battle sites, in sequence, are a tower on a hill, a crossroads, a field with hay bales, and a town. The general gist of the story is that a peasant stole the LSD source code from the player's team, and now they must go on a peasant killing rampage in order to get it back.

2.6 Scoring

Score is based entirely on how much damage was dealt by the party. Note that dealing damage to other party members still counts towards your score...

3 Bugs and Possible Improvements

While the game is playable and fun, it's not bug free. We've spotted the following:

- There are numerous graphical issues. For instance, at some camera angles it is possible to get units' heads to stick through walls, and the transparency of selectors does not quite work right when two selectors overlap.
- The game slows to a crawl when many tiles are indicated as targetable or effected, possibly because the sphere indicators have too many polygons.
- The AI is in need of some revision. We've observed it go into what appeared to be an infinite loop, but it may have simply been an exponential explosion in possibilities for a unit with too high of a speed stat (the AI uses a complete search to find the most damaging move at the moment; it's wasteful, but it's still pretty quick, most of the time).

Also, there are a few areas where our game could be improved:

- We didn't even touch on many of the dimensions of tactical RPGs. We could have added items, status effects, leveling up and gaining skills, a world map, and, of course, chocobos.
- More battles would make the game more fulfilling, and more enemy types might have increased the challenge.
- Graphics could have been improved in general. Better character sprites could have been chosen and the battle sites could have been made more detailed.
- Animation, sounds, and music could have been added.
- The AI could have been made smarter by using a better heuristic than dealing as much damage as possible.