

Vertex Dispenser

Michael Brough

May 24, 2011

Abstract

Vertex Dispenser is a fast-paced abstract strategy game. Wrestle for control of territory on bizarre geometric worlds. Carefully plan your conquests to maximise your colours, then eradicate your enemies with a variety of special attacks.

The **Vertex Dispenser** *campaign* has been designed as an introduction to the game, and is recommended as the primary source for beginners. This document is intended as a detailed reference for those already familiar with the game.

Contents

1 Overview	2
2 Vertices	3
3 Faces	3
4 Players, Robots, Buildings	3
5 Abilities	4
5.1 Blue	4
5.2 Red	5
5.3 Green	5
5.4 Yellow	6
5.5 Cyan	6
5.6 Magenta	7
5.7 White	8
6 Assorted Advice	8
7 Credits	9

1 Overview

In **Vertex Dispenser**, players fight for control of territory on a geometric surface. Territory consists of *vertices*, *edges*, and *faces*. A *vertex* is a coloured node which produces a resource over time. An *edge* connects two vertices, and a *face* is a flat polygon with a perimeter of vertices and edges. Territory belonging to a player is displayed in their team colour, and unclaimed neutral territory is grey. Multiple players may be on the same team and share the same territory.

Each player controls a single unit, which acts on edges either by moving along them (if they are part of the player's territory), capturing them (if they are neutral), or attacking along them (if the other end is an enemy vertex). These actions are mutually exclusive - only one can be applied to any given edge - so they are combined into a unified control scheme: attempting to move into a neutral edge will attack or capture. Players may also press **Shift** to attack without moving - sometimes useful in a situation where you would not want to accidentally move forward - or **Ctrl** to *delete* the vertex in front of them, removing it from their territory - useful for controlling vertex colours.

When an edge is captured, both of its end-vertices are also captured, and the edge is lost when either of those vertices is lost.

When all edges of a face are captured, the face is also captured, and the face is lost when any of those edges is lost. Faces automatically attack enemy vertices adjacent to any of their vertices.

There are seven vertex colours: Blue, Red, Green, Yellow, Cyan, Magenta, White. These are ordered as given, with Blue the lowest-valued colour and White the highest. When a vertex is captured, it takes the lowest colour not on any adjacent vertex (a greedy colouring algorithm). A faint overlay on adjacent neutral vertices shows the colour they will become if captured immediately.

Each player has an energy bar for each colour, which increments at a rate based on the number of vertices of that colour they control. There are diminishing returns; two vertices of the same colour give less than twice as much energy as a single vertex does (the exact formula is $\frac{3\sqrt{2n-1}}{4} + \frac{n}{4}$, where n is the number of vertices of a given colour). (When an energy bar reaches its maximum value, it stops accumulating energy, and it may be used to activate a special ability depending on its colour, resetting the bar to zero. When there is more than one player on a team, energy income is reduced slightly; in the above formula n is replaced by $\frac{5n}{p+4}$, where p is the number of players.)

Each player has two abilities for each colour, chosen before the game starts from four total abilities for each colour. With 4 abilities for each of 7 colours, there are 28 different abilities in total, and $\binom{4}{2}^7 = 279936$ different possible combinations.

2 Vertices

Let v be a vertex, $c(v)$ the number of its colour (ranging from 1 for blue to 7 for white), and $d(v)$ the number of non-neutral edges adjacent to v . The vertex v has a maximum of $\min(1 + c(v) + d(v), 12)$ hit points, which may be reduced by attacks. If the number of hit points ever drops to zero, the vertex is lost, becoming neutral again, and any adjacent edges are also lost.

Some vertices are big and shiny. These have $\min(8 + c(v) + d(v), 12)$ hit points, and generate as much energy as three ordinary vertices.

3 Faces

Most of the time, a face lies passive. At regular intervals, it will check whether any neighbours of its vertices belong to an enemy, and if so, will rise up to attack. When it has risen, if there are still any adjacent enemy vertices, it will pick one at random and fire upon it (doing 2 damage) and then retract back into a passive state.

Sometimes a passive face will rise up a short way and then retract; this is purely a graphical effect and does not affect its behaviour.

Faces may also be toggled off, in which case they are not visible and cannot attack.

4 Players, Robots, Buildings

Some Cyan abilities create *buildings*, which are fixed to a particular vertex. All Magenta abilities create *robots*, which may move between different vertices (though most are restricted to a small area around their origin).

If a player, robot or building occupies a vertex, then any damage that would be done to the vertex is done to the unit instead. A player/robot may occupy the same vertex as a building, and in this case the damage is done to both.

Players and robots slowly recover from damage over time; buildings do not.

When a player is killed, they will respawn after 7.5 seconds. During this time they may look around the level and select one of their vertices to spawn on. If all of their vertices are lost, there is nowhere for them to respawn and they are eliminated from the game. If all of their vertices are occupied, they will wait for 3.6 seconds and then attempt to respawn again.

Player attacks usually deal 3 damage, but if you attack immediately after moving forward, you will do a *charge* attack dealing 4 damage.

	Hit Points	Damage	Splash	Recover 1 HP
Player	20	3		0.9s
Attack Robot	20	2		5.4s
Repair Robot	16			5.4s
Capture Robot	18	2		5.4s
Shield Robot	16			5.4s
Fortification	28			
- shield	14			
Laser Turret	22	2		
Roaming Mine		9	3	
Boobytrap		8		
Hunterseeker		10	4	
Guided Bomb		10		

5 Abilities

5.1 Blue

Mighty Vision

For 10.8 seconds, hit point overlays are displayed for all units (including enemies), and vertices and edges out of sight are revealed. Invisible enemies are also revealed, but their hit point overlays are not displayed. Boobytraps directly adjacent to the player are permanently revealed.

Teleportation

The player is teleported away for 3.6 seconds and can navigate the surface to select a destination. During this time, hit point regeneration is increased and timers for other abilities are paused. It is possible to use other abilities while teleporting; their effects appear at the currently selected vertex.

Invisibility

For 6.3 seconds, the player is hidden from the view of enemies. This deceives robots and Laser Turrets as well as other players. It is still possible to determine their location through careful observation, or by using Mighty Vision.

Toggle Faces

Each adjacent toggled face is turned on, and each other adjacent face is toggled off.

5.2 Red

Roaming Mine

Creates a missile that first moves forward, then randomly. It explodes when it hits an enemy unit, or with probability $\frac{1}{2}$ when over enemy territory, dealing 9 damage to its current vertex and 3 to adjacent vertices. It will always move into adjacent enemy territory if there is any, and otherwise will always prefer to move into neutral territory. Does not damage allies.

Local Capture

Captures each vertex and edge of each adjacent face, if they are not already owned by an enemy player.

Vertex Attack

Each of the player's vertices fires upon each adjacent enemy vertex, doing 2 damage.

Boobytrap

Creates a trap on the current vertex. When an enemy attempts to capture the vertex, the trap will activate: the vertex is captured and fires upon all adjacent edges, dealing 8 damage or capturing them. Multiple Boobytraps may be stacked on the same vertex.

5.3 Green

Star Attack

Fires along any adjacent edge that is not yours. When these attacks hit, they deal 3 damage, and fire along any adjacent edge that is not yours, apart from the one they just moved upon, dealing 2 damage.

Protection

A shield is created around the player, providing a buffer of 42 extra hit points. Any damage that would be done to the player is dealt instead to the shield, and it also decays over time.

Pulse Wave

Pushes away adjacent units, stunning enemies for a short time, destroys enemy attacks and missiles, and cancels enemy Air Strikes.

Repair Swarm

Creates fifteen small particles which move randomly on your territory for 8.6 seconds. If one hits a damaged vertex or unit, it will explode, healing 2 hit points.

5.4 Yellow

Hunterseeker

Creates a missile that tracks the nearest enemy unit and moves towards it. If no enemy is within distance 4, it will just continue moving forward. It explodes when it hits an enemy unit, or with probability $\frac{1}{9}$ when over enemy territory, dealing 10 damage to its current vertex and 4 damage to adjacent vertices. It damages allies as well as enemies.

Close Gaps

Captures every edge within distance 3 which you already own both ends of.

Air Strike

Sets a timer which waits for 8.1 seconds, then attacks 25 random nearby edges. If both ends of the edge are neutral or friendly, it will be captured; if one end is owned by an enemy, the other end will be captured and a 2 damage attack will be fired along the edge; if both ends are enemy, they are each dealt 1 damage.

Recolour

Changes the colour of the current vertex to the lowest not on itself or any adjacent vertex.

5.5 Cyan

Guided Bomb

Creates a missile which you control directly with the movement keys, while your main vehicle is fixed in place. It lasts for 3.6 seconds or until it hits an enemy unit, and then explodes, uncapturing all nearby edges and faces (including your own), but leaving vertices unharmed. If it directly hits an enemy unit, it also deals 10 damage and stuns it.

Fortification

Constructs a Fortification building, which protects its vertex and anything on it. For every three adjacent vertices, a small shield is created; these rotate around the central one and similarly protect anything currently beneath them.

Fortifications take half damage from most special abilities, but full damage from direct attacks by players, robots, and faces.

Doppelgänger

Creates a temporary duplicate player which responds to input simultaneously with the player's main vehicle. The Doppelgänger also duplicates the effects of abilities of lower level than Cyan. Lasts for 6.3 (5.4) seconds.

Laser Turret

Constructs a building which attacks the nearest enemy unit it has a line of sight to, dealing 2 damage. The delay between attacks increases linearly with the target's distance.

5.6 Magenta

Attack Robot

Creates a robot which can simultaneously attack all adjacent enemy vertices, dealing 2 damage. An Attack Robot may not move further than distance 2 from its origin. It will attempt to preserve itself when damaged, by staying back at its origin and away from danger.

Repair Robot

Creates a robot which moves about and repairs damaged vertices. If any adjacent vertices are damaged, it will move to one of these, otherwise it moves randomly, preferring not to move into danger and not to move back to where it just moved from. It does not repair buildings or units; only vertices.

Capture Robot

Creates a robot which captures territory in an area. A Capture Robot may not move further than distance 2 from its initial vertex. It prioritises capturing edges both of whose end-vertices are already captured, and will not capture a vertex which is threatened by an enemy. It will attack if it has no option to capture territory or escape to its origin.

Shield Robot

Creates a robot which blocks any enemy attacks directed towards its shield, which covers one adjacent edge. A Shield Robot may not move further than distance 2 from its initial vertex. It attempts to protect threatened territory while minimising danger to itself.

Shield Robots are not aware of invisible enemies.

5.7 White

Damage All

Deals 2-5 damage to all enemy objects, and 1-2 damage to all unoccupied enemy vertices (damage dealt is proportional to the target's current hit points).

Expand Edges

Captures each edge of which one end-vertex is yours and the other is neutral.

Time Freeze

Suspends all activity on the level except the movement of allied players, for 4.5 seconds.

Prism Colours

Fills your energy bars for all colours below White.

6 Assorted Advice

- Don't worry if you find vertex colouring difficult. It's possible to mostly ignore it and just use lower-level powers, and if you practise on the **Colour Puzzle** levels eventually you will master it. Also, in a team game, all players share the same territory, so one player can take care of colouring while others attack!
- If you want to handicap a stronger player in multiplayer, playing in teams with AIs of different levels works well; e.g. an experienced player allied with *Easy* vs. a beginner allied with *Hard*. (This also works for a single-player challenge: give yourself *Easy* or *Medium* allies against a team of *Hard*.)
- Having different numbers of players on a team is not recommended; this is not expected to be balanced at all. (Playing co-op against a superior number of AIs can be a good challenge for experienced players though.)
- It is recommended that beginners play with just two teams.
- The *sphere* and *plane* levels are recommended for beginners. Levels like *torus* or *node_cube* are more difficult to make progress on, because in general there is a smaller boundary between the territories of opposing players.
- Players have a *level* in multiplayer games, which is intended as a crude estimate of how skilled they are. It is increased by winning multiplayer games, with more experience awarded for beating players of higher level, and it increments for each single-player campaign completed.

7 Credits

Design, programming, graphics, sound, basically everything:

Michael Brough

Additional music and audio design:

Arshan Gailus

Playtesting, advice, encouragement, and general awesomeness:

Rhys Abraham, Andrew Brockert, Tara Brough, Terry Cavanagh, Kate Farb-Johnson, Tom Fraser, Ricky Haggett, James Harrison, Sophie Houlden, Darius Kazemi, Ed Key, Stephen Lavelle, Kelvin Lomborg, Scott Macmillan, Brendan McArdle, Luther Patenge, John Polson, Glen Robertson, Duncan Robertson, Jonathan Rogers, Seth Rosen, K. Adam White, Jonathan Whiting, Chris Wilson.

Libraries used:

Boost, GLEW, libogg, libvorbis, OpenGL, SDL. Fixed point arithmetic closely based on Markus Trenkwalder's code (<http://www.trenki.net/>).