# Title

The name “Vortex” comes from a game published in Ahoy magazine, I see this as a sequel of sorts with a similar concept but multiplayer. Need a good name for a sequel (Beyond the Vortex?).

# Gameplay - General

The game plays similarly to Zone Ranger and Raid on Bungeling Bay in that it is a top-down view, 2D scrolling game with movement in any direction. Fly by pushing the joystick in that direction and shoot by pressing the button. Powerups can be triggered with the keyboard.

An important gameplay element is the ability to destroy some of the blocks in the map (similar to Fort Apocalypse or the original Vortex). This allows the players to blast through planetoids and remove obstacles (asteroids, protective files etc.)

The object of the game is primarily survival, gathering energy orbs (mostly contained within planetoids) and other powerups placed within the map. A larger-scale goal of defeating the enemy, scoring points along the way.

To encourage multiplayer, it should be impossible to ‘win the game’ alone. Teamwork such as some players blasting away tiles while others fend off the enemy ships should be common tactics.

Enemy ships are controlled by the server, which will generally chase and attack the closest player, though different behaviours could be programmed. They could also potentially rebuild areas that the human players have destroyed.

The game should be fun and fast-paced!

# The Map

The map will be 100x100 screens (10,000 screens!), this is large enough to be technically impressive but not so large that players and enemy ships can’t find each other.

The map data would be generated by the server on first run, if there is no pre-existing map. It should be serialized to disk periodically, and re-read on startup if it exists.

Elements in the map would be planetoids with a powerup in the middle, asteroid belts, an auto-generated “maze” with a significant powerup in the middle… etc.

# Networking

The currently visible screen, as extracted from the map, is streamed in a single packet 20 times per second to each client C64. This is admittedly heavy bandwidth-wise, but it drastically simplifies the logic, and allows all map mods, animations, and things like open-closed doors to be controlled entirely by the server.

At startup, the character set and sprite data could be sent from the server.

# Raster Time

Currently looking at a round-robin approach to managing the raster time to balance everything that has to be done with smooth game play.

i.e.

All frames: Poll the joystick and update direction/velocity.  
Frame #1: Send an update to the server with current player actions (takes most of a screen)  
Frame #2: Check for and receive packets from the server (takes most of a screen).  
Frame #3: If a packet was received in previous frame, copy it to the screen buffer and scroll the screen (take most of a screen).

# Sound

TBD.

# Gameplay - Details