**Title: Parabolic Game Suite**

**Category: PyGame Game DELUXE SUITE**

**Repository: UNABLE TO CREATE REPOSITORY CURRENTLY**

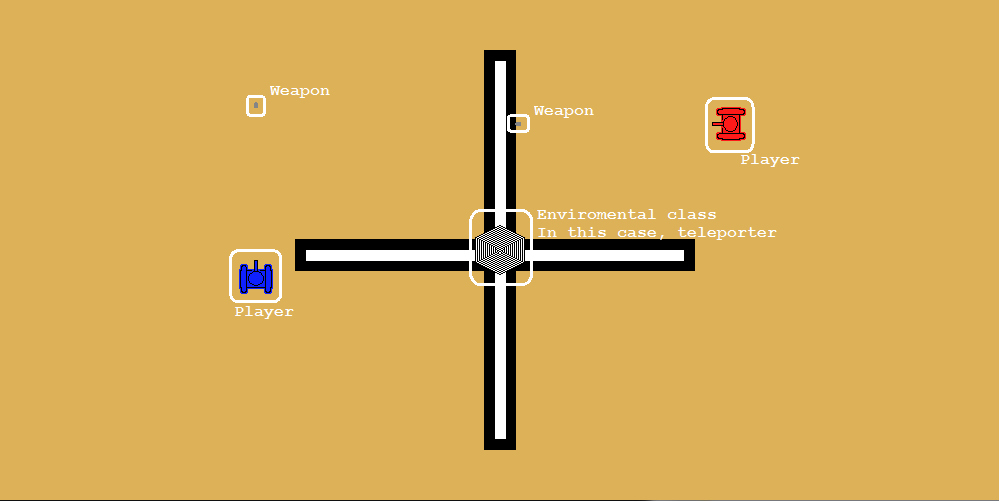
**Summary: A suite of several games, including GALAXE, an old fashioned space invaders game, Defender, a game in which the player flies through a canyon that continually gets smaller while shooting at aliens and collecting powerups, Pong, which is kind of obvious, Tanks, which is a multiplayer game in which two tanks drive around and shoot at each other, and more as the project progresses, will all be included on one disc. The disc will be an installer only, it will not be necessary to have the disc in to play the games. The programs will most likely be ran separately, but there is a possibility that a menu will be added as the project goes on. Most of these games will have a HighScore table that appears on the screen when the game is over. This can be accomplished via pickling and unpickling. All these games will provide for a variety of fun activities for users.**

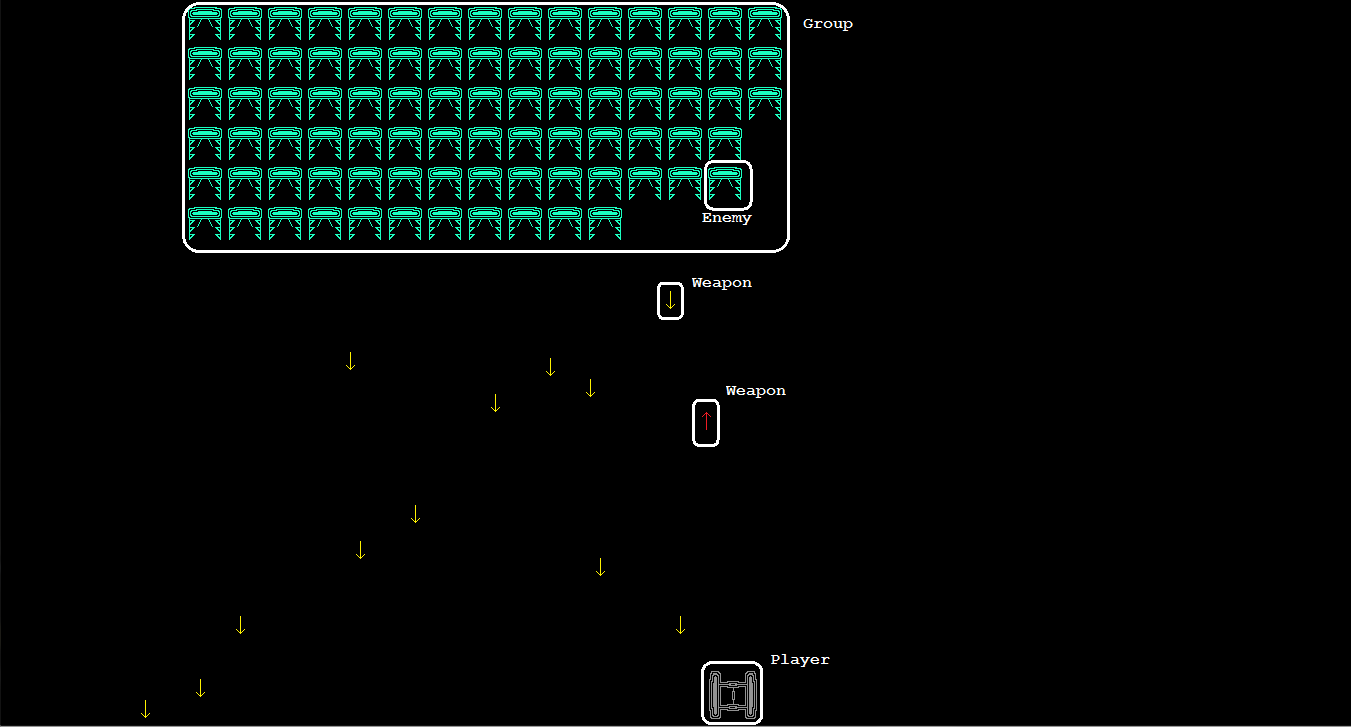
**Also on the disc will be a Python 3.1.2 32-bit download for Windows, a PyGame download for Windows, and a LiveWires download for Windows. As the project develops, it will most likely expand to Linux operating systems as well, and possibly even Mac OSX. All of these downloads and all of the games will be automatically installed via an autorun program. The autorun program will be written as an exe or a batch file (for Windows) that will run the .msi installers and put the game files onto the computer. For Linux, the program will be a .sh file. The media files will be in a file on the hard drive so that there won’t be too many files in the folder.**

**Implementation: For each game, there will be several classes necessary, most of them based off a games.Sprite class and in some games a *game* class will be made to create all sprites necessary and start the games.screen.mainloop.**

**For each game, everything that moves will be a class. Some games will have a game class to tie everything together. Most games will have a class for the player’s object that checks for keyboard input in the update method. Other classes can be sorted into enemy, environment, weapon, and group categories. Enemies will mostly be immobile objects that fire at the player. Environment classes will be a variety of things, from powerups to walls to teleporters. Most will have a function to kick out anything that is overlapping, like a wall. Weapon classes would be almost identical for all games. They would have a dx or dy so they would only move in one direction. When there is something in weapon.overlapping\_sprites the object will have its die method invoked and it will be destroyed. Then the projectile will be destroyed. Group classes will create a bunch of one class and make them move.**

**Some examples of games and classes are provided at the bottom of the proposal.**

**Teamwork: The size of the Parabolic project alone allows a multitude of students to all work on different parts of the project without ever interacting. There are many games which are independent of each other, and many classes in all of the games, which don’t overlap for the most part. If there are too many people working on the games, then there are still the installation features to work on. In event of minor clashes, the students involved and the Project Leader would settle the dispute. In the event of a major clash, everyone on the team would be called for a tribunal deciding what to do on the matter. As team size increases, the number of people on one or more features can be expanded to lessen the work load per person.**

****