Project Name:

TrueTris

Description:

A low bandwidth port of Tetris with multiplayer and role-playing game style modes in addition to the staple 15 level standard, 2 minute ultra, and 40 line sprint modes.

Competitive Analysis:

Tetris is obviously already a huge franchise that has stayed relevant for almost 40 years now, so there is a lot of competition from existing Tetris titles. The two most up-to-date titles to compete with being Puyo-Puyo Tetris (2014), which experimented with the fusion of standard Tetris mechanics with a popular Japanese game similar to Tetris called Puyo-Puyo, and Tetris monsters, a mobile game released only in Japan that added a similar RPG-like system to Tetris, however, with the limit of mobile device capabilities. Tetris monsters is a mobile game that fell quickly after its release. Monsters was already difficult to research for the sake of this proposal, therefore, this clone is meant to build on Tetris monsters. I plan to rewrite Monster’s RPG style mechanics to the point where said battle mode is refined to a point where it is comparable to Puyo-Puyo Tetris which was popular enough to find releases on major home consoles like the PS4, Nintendo Switch, and Xbox One.

Structural Plan:

Each class gets own .py file in final deliverable (for TP1, the code is still concise enough to be written in one .py)

Each game mode will have its own class with a different run functions. Main Game.py will implement all of these game-mode classes and run the overall program.

For the actual Tetris game, there will exist two objects: Matrix and Piece. Matrix manages play grid in addition to Piece queue and held piece. Piece manages its location in the Matrix, in addition to its shape and rotation state.

Enemies for RPG mode will be individual classes inherited by the RPG mode class.

Socket usage will be determined for multiplayer mode

Algorithmic Plan:

The hardest part of this project is giving the player the ability to choose modes and have the game run differently depending on the current mode. This is going to be done with each mode being a class with a run function. A main Game.py will manage all of these modes and run the correct class. Most likely going to need a Game.py, menu.py, ~~RPG.py, multi.py,~~ and standard.py (sprint.py, ultra.py will inherit standard due to similar properties) [EDIT: we may have every game mode inherit standard as standard is the basis for all Tetris, other modes merely edit conditions]

Timeline:

4/18 – port standard Tetris to pygame logic.

4/25 – have working multi and 1 level of RPG mode, all standard Tetris modes should work

5/3 – “completed” game with 2-3 levels of RPG mode and all working modes

Version Control:

I have no clue how to use git nor do I have time to learn at the moment. Using Gdrive as version control uploading all TP files to drive arrive deliverable due dates.

Module List:

* Pygame

TP2 UPDATE

Design Changes:

* Project split up into multiple files for better readability
* Continued following Tetris Guidelines more closely documented here:

<https://www.dropbox.com/s/g55gwls0h2muqzn/tetris_guideline_docs_2009.zip?dl=0&file_subpath=%2F2009+Tetris+Design+Guideline.pdf>

* Utilizing Data() struct for universally accessible variables. Mostly reserved for player interactive conditions such as the falling piece dubbed data.piece, the board dubbed data.board, the queue of next pieces data.pieceQueue, etc.
* Collections implemented for deque structure. Much more efficient than a list and considered the optimal implementation for queues within python language. Deque = double ended queue
* Matrix updated for view of rows 22-40 as per Tetris Guidelines
* Pieces kick correctly as per SRS guidelines
* [NOT YET VISUALLY IMPLEMENTED]: Scoring works. For the time being it is printed out in the shell

TP3 UPDATE

Design Changes/Additions:

* Menu completely integrated
* Basic Tetris modes functional with high score saving capabilities (Standard, Sprint, Ultra)
* You can now see everything that you need to: the 6 pieces in the “next” queue, the currently held piece, combo counter, etc.
* Battle mode fully implemented with functional menus and one boss battle: Morpho Knight
* 2 Active skills added: -4 Lines and Unsealer
* The UI is fancier now because backgrounds actually exist and are animated!
* Help mode integrated
* Credits roll integrated
* Music and sounds fully implemented + player ability to modify in-game volume
* Project name update: “Tris”
* A lot of spriting got done.
* Removed plans regarding sockets.