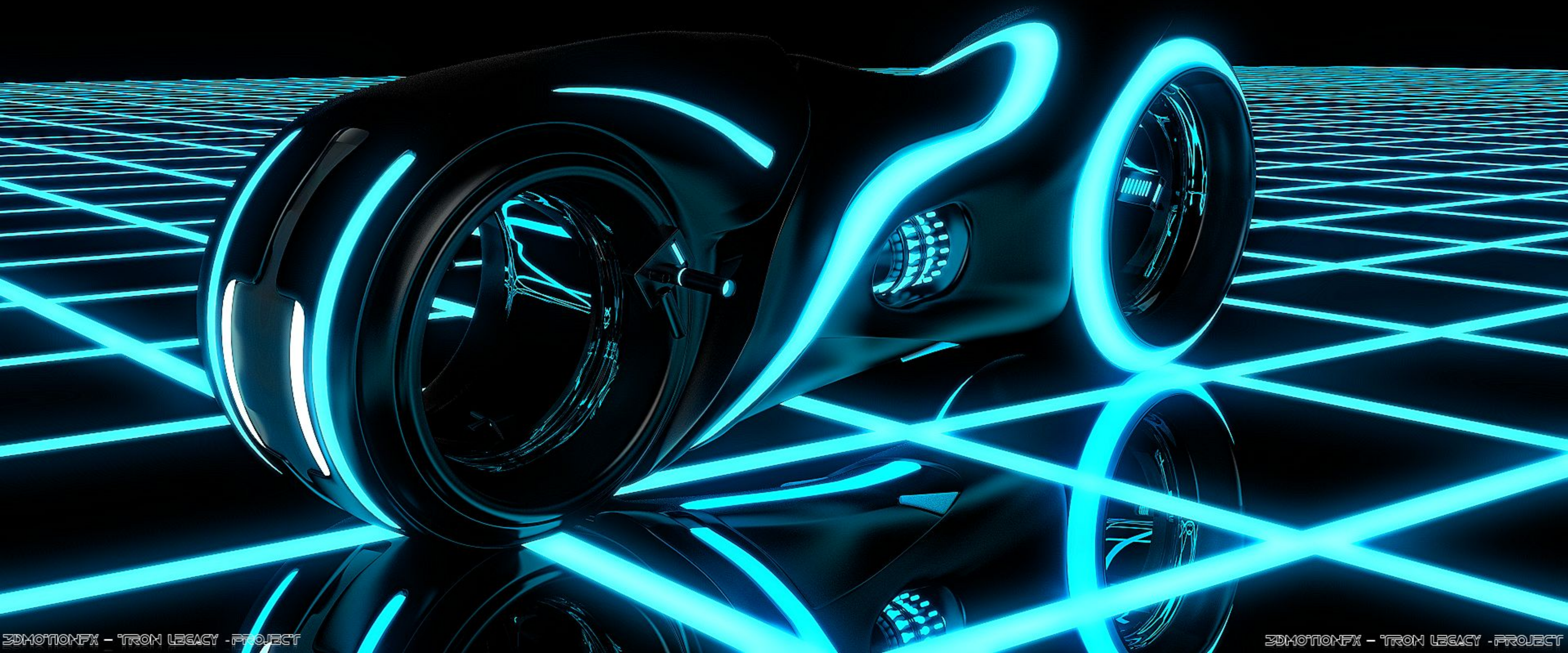
REVAMPING TRON

Naomi Chiu | Jessie Potter | Alex Frye


3DMOTIONFX – TRON LEGACY -PROJECT
3DMOTIONFX – TRON LEGACY -PROJECT

Key Questions

- What are your thoughts on our current plans moving forward?
- Do you have any suggestions to clarify our code? What needs more commenting?
- Do any of you have experience with minimax?
- Do you think a aggressive bot that seeks to cut off opponents or a defensive bot that seeks to survive the longest would be a more formidable player?

Where are we going?

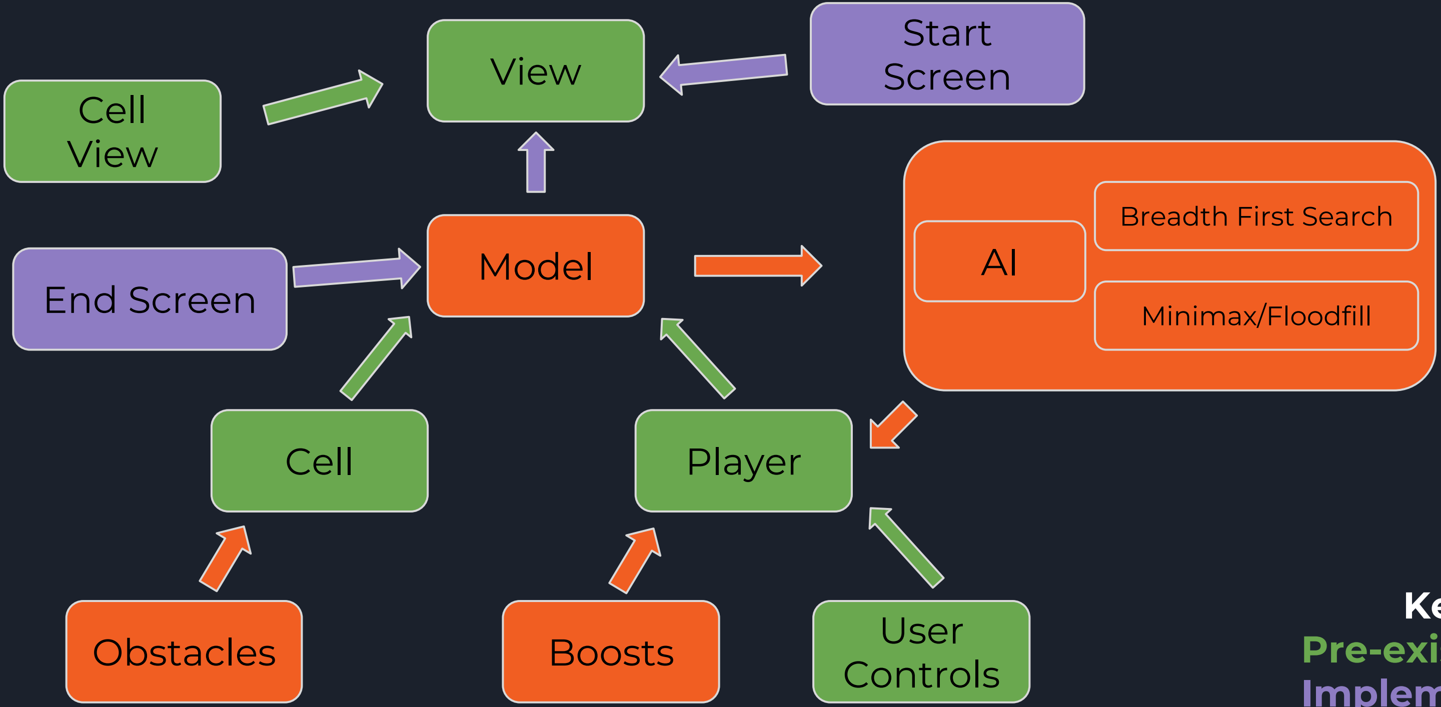
- Focusing on developing an semi-competitive AI
- Hoping to implement with player v. player v. CPU free-for-all
    - Stretch goal: make user interface so number and combinations of players can be customized
- May try to create game maps to increase variety and difficulty

What have we done so far?

- Improved game packaging with a start and end screen now
- Divided up classes into separate files for better readability instead of one ginormous file
- Cleaned up unnecessary code and compiled classes for concision
- Did research on possible AI types (A* and Breadth First) and attempted implementation

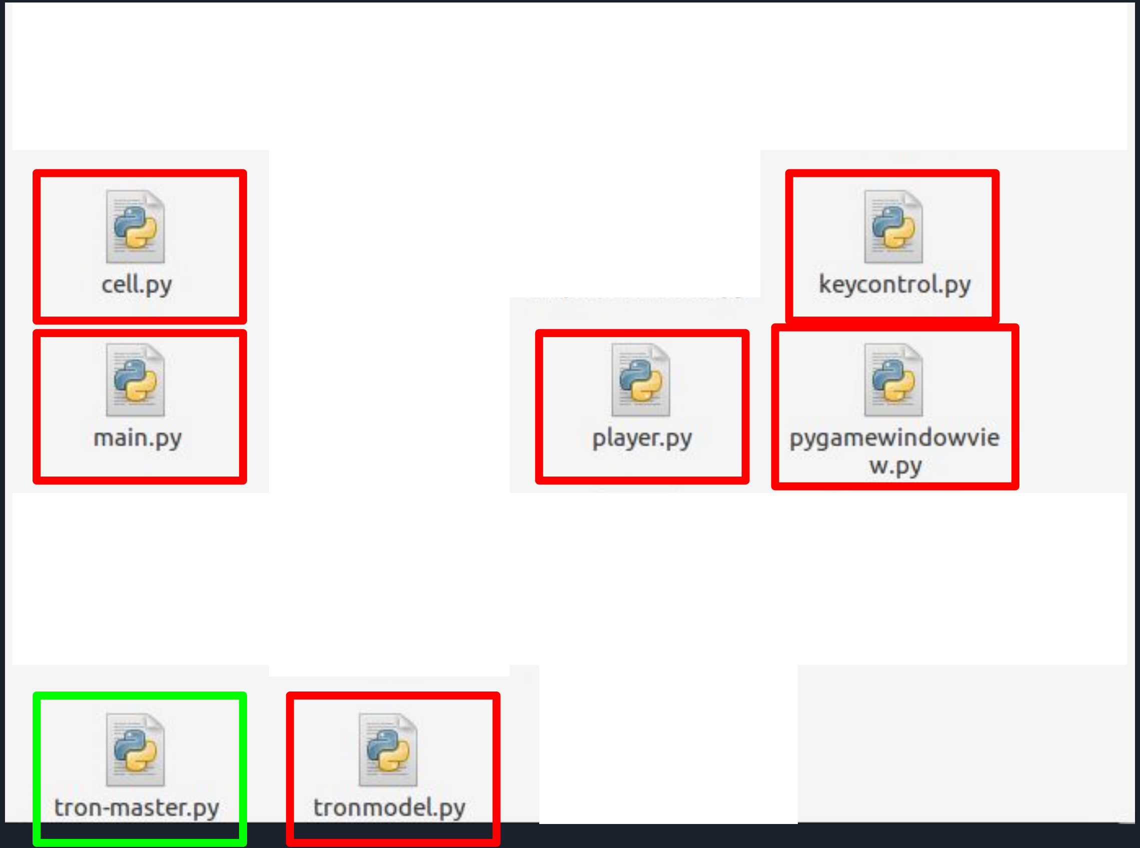
DEMO TIME

Updated System Architecture Diagram

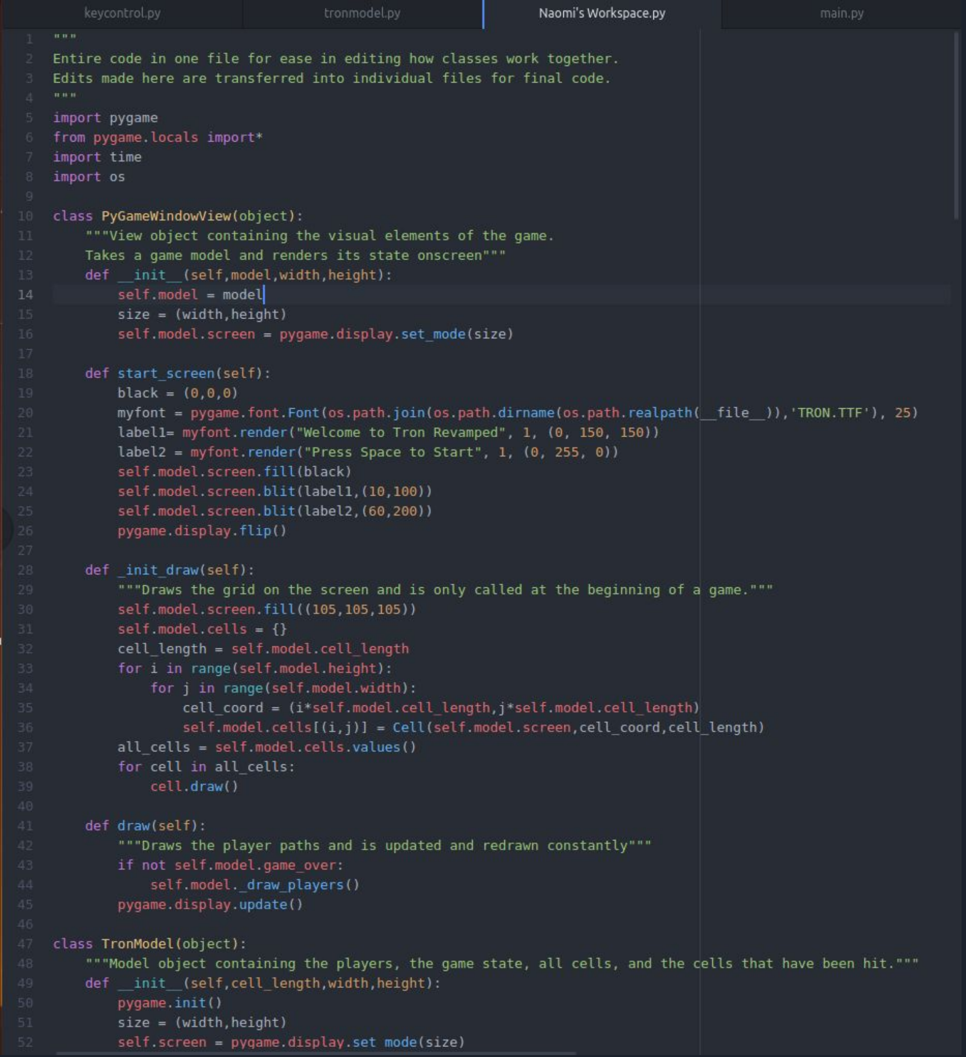


Key
Pre-existing
Implemented
In Progress

- tron-master.py was broken into separate files for each class.
- Using file imports to import classes to reference in other classes



Our code for our model class for our MVC



```python
"""
Entire code in one file for ease in editing how classes work together.
Edits made here are transferred into individual files for final code.
"""
import pygame
from pygame.locals import*
import time
import os

class PyGameWindowView(object):
    """View object containing the visual elements of the game.
    Takes a game model and renders its state onscreen"""
    def __init__(self,model,width,height):
        self.model = model
        size = (width,height)
        self.model.screen = pygame.display.set_mode(size)

    def start_screen(self):
        black = (0,0,0)
        myfont = pygame.font.Font(os.path.join(os.path.dirname(os.path.realpath(__file__)),'TRON.TTF'), 25)
        label1= myfont.render("Welcome to Tron Revamped", 1, (0, 150, 150))
        label2 = myfont.render("Press Space to Start", 1, (0, 255, 0))
        self.model.screen.fill(black)
        self.model.screen.blit(label1,(10,100))
        self.model.screen.blit(label2,(60,200))
        pygame.display.flip()

    def _init_draw(self):
        """Draws the grid on the screen and is only called at the beginning of a game."""
        self.model.screen.fill((105,105,105))
        self.model.cells = {}
        cell_length = self.model.cell_length
        for i in range(self.model.height):
            for j in range(self.model.width):
                cell_coord = (i*self.model.cell_length,j*self.model.cell_length)
                self.model.cells[(i,j)] = Cell(self.model.screen,cell_coord,cell_length)
        all_cells = self.model.cells.values()
        for cell in all_cells:
            cell.draw()

    def draw(self):
        """Draws the player paths and is updated and redrawn constantly"""
        if not self.model.game_over:
            self.model._draw_players()
        pygame.display.update()

class TronModel(object):
    """Model object containing the players, the game state, all cells, and the cells that have been hit."""
    def __init__(self,cell_length,width,height):
        pygame.init()
        size = (width,height)
        self.screen = pygame.display.set_mode(size)
```

Where do we hope to go?

- After asking some experienced game makers, we found that our best bet for AI for this game would be through Minimax and Floodfill
- We are beginning to build our own AI and are hoping to get it running ASAP
- AI is our primary concern but we still have additional features in gameplay in consideration to implement

Key Questions

- What are your thoughts on our current plans moving forward?
- Do you have any suggestions to clarify our code? What needs more commenting?
- Do any of you have experience with minimax?
- Do you think a aggressive bot that seeks to cut off opponents or a defensive bot that seeks to survive the longest would be a more formidable player?