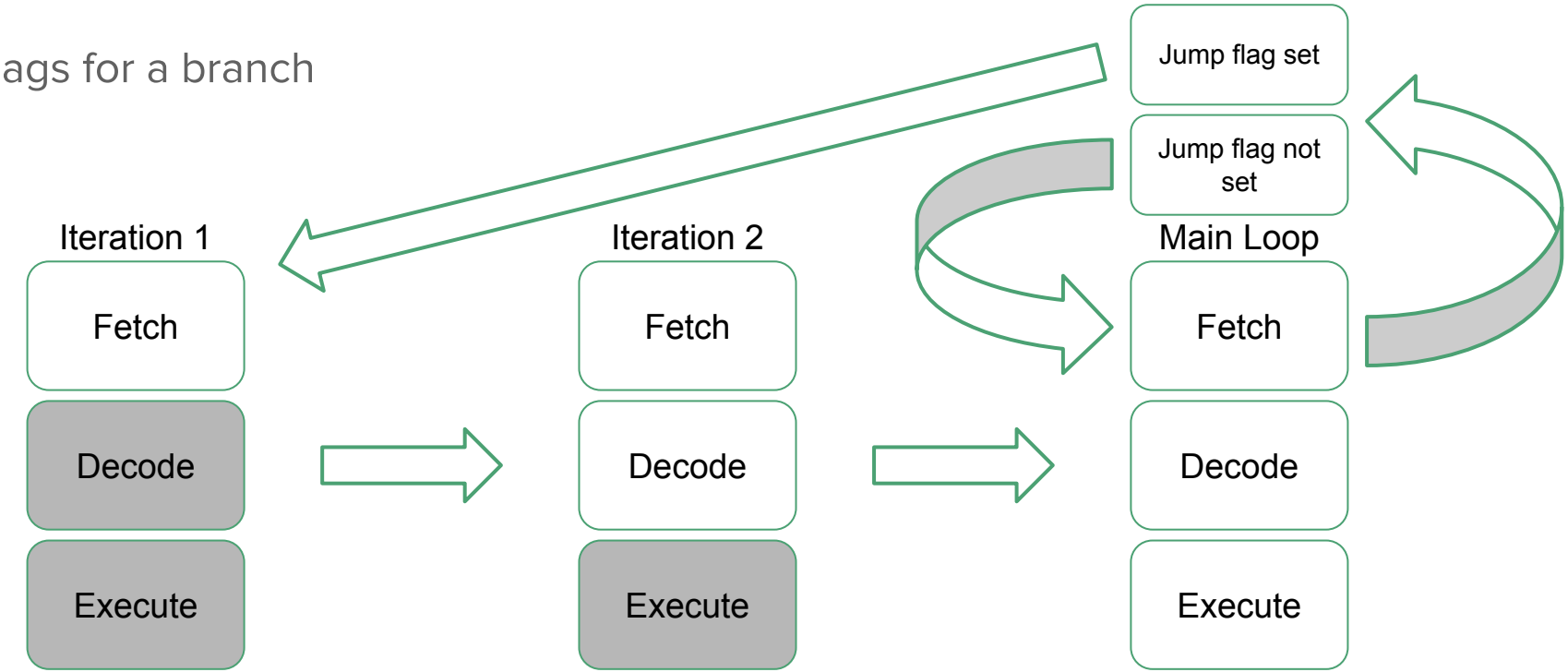
ARM 11 Project

Group 38:
Shashwat Dalal • Marcel Kenlay • Andy Li • Thomas Yung

The Emulator

The Main Executable Loop

- Decode and execute aren't always in use
- Flags for a branch



The Assembler

Data Management

- Lines stored as linked lists
- Function pointers

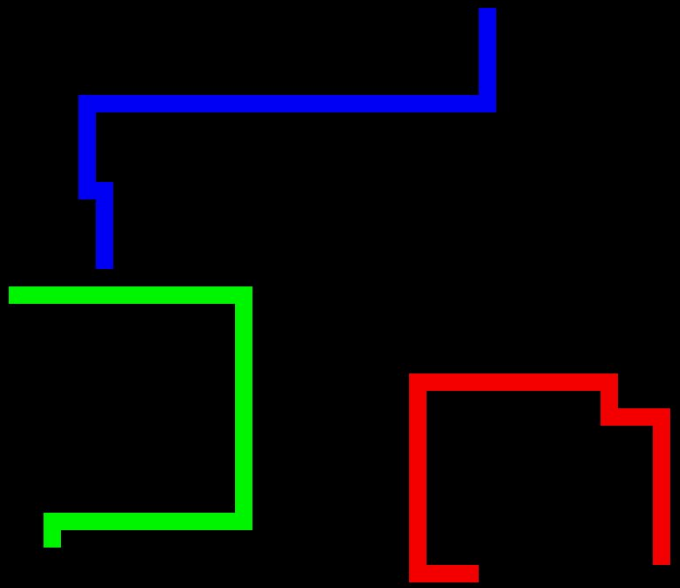
```
llist_t *initLinkedList(void);
void freeLinkedList(llist_t *list);
void llistEnqueue(llist_t *list, void *value);
void *llistPop(llist_t *list);
```

GPIO

Making The State Obvious

```
onloop:
add r2,r2,#1
cmp r2,#0x880000
blt onloop
```

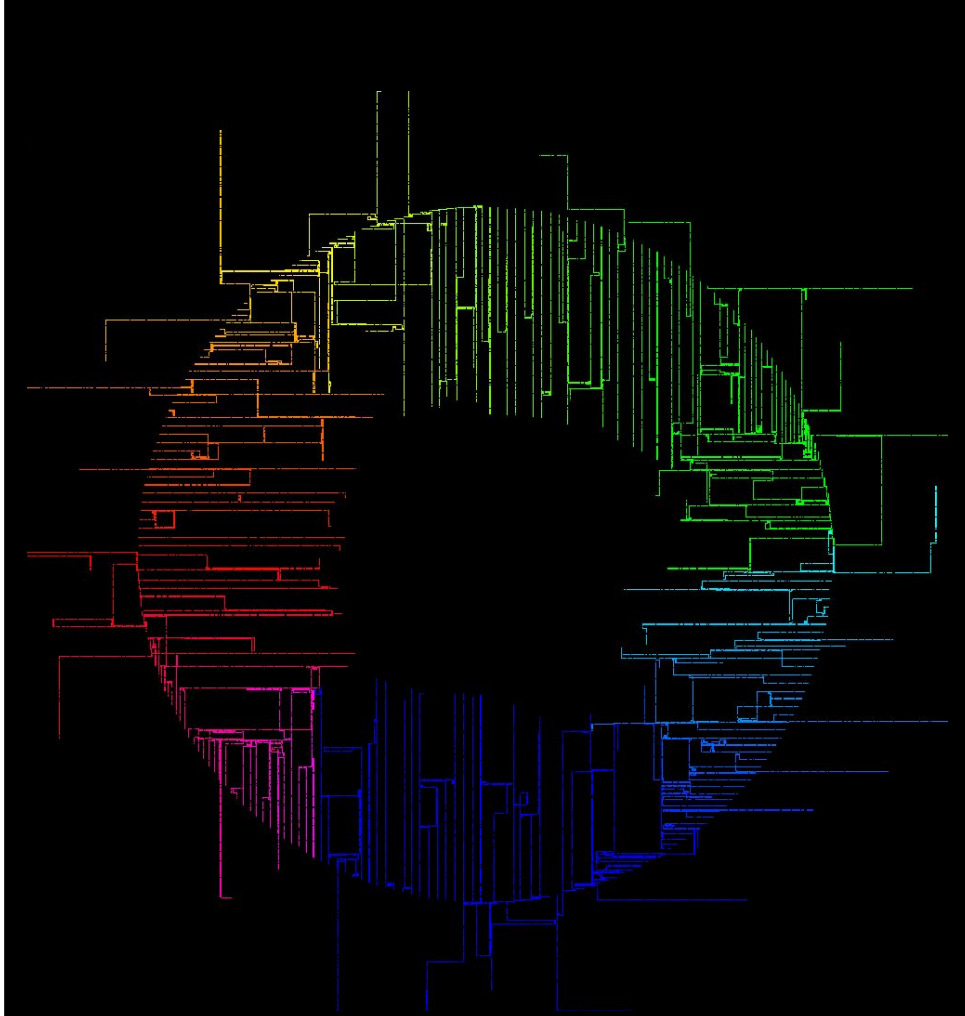
Extension: TRON



Pictured (2017): A game with 3 players

- 2 or more players
- Each player leaves a trail
- Crossing a trail eliminates the crosser
- Last bike standing wins
- Holes optional

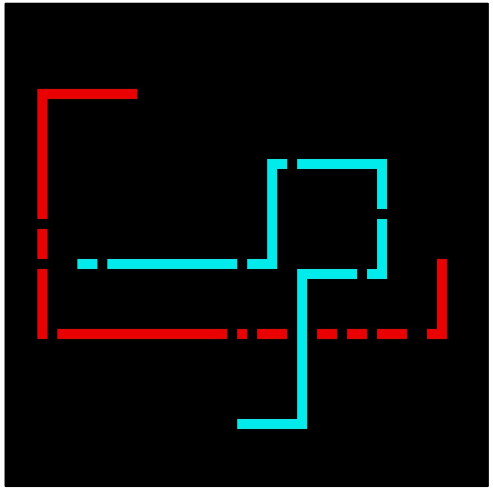
- Game engine
- Graphical output
- AI / Computer Opponent



Pictured (2017): A game with 250 players

The Game Engine

- No explicit main loop but rather an implicit loop
- How to get keyboard input
- The loop moves all players based on their current direction
- Needed to ensure both players are eliminated when a head on collision occurs.



Graphical Output

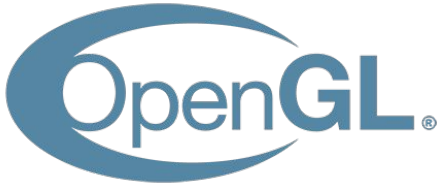
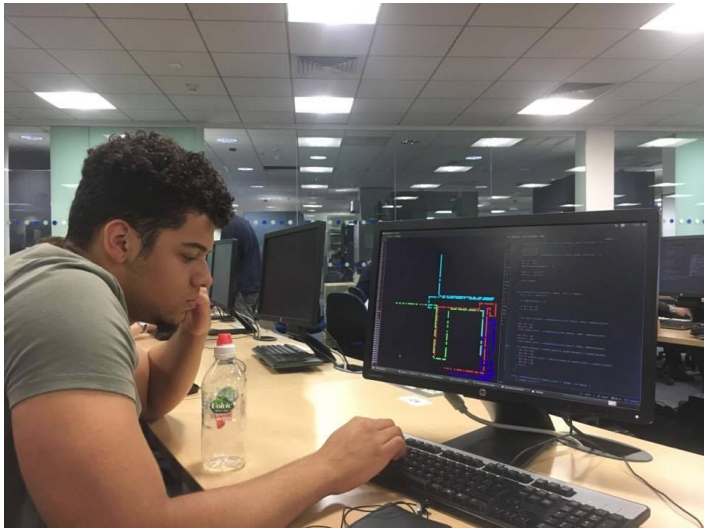


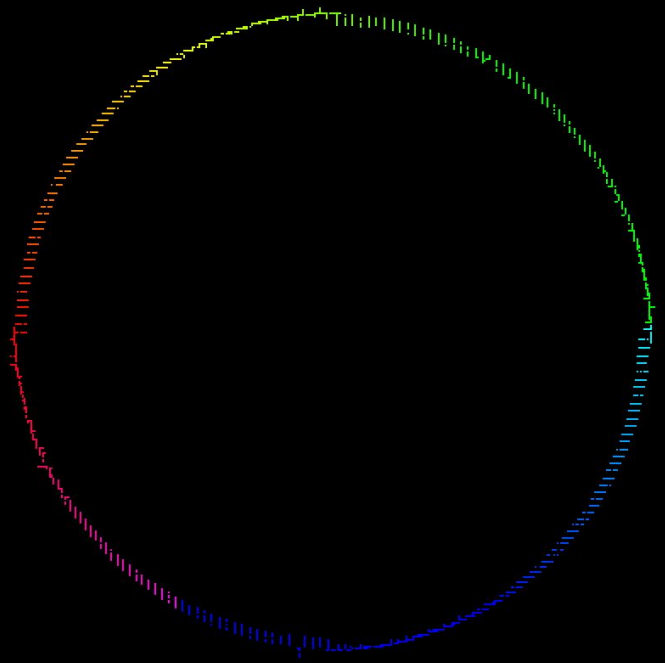
Image: www.opengl.org

- Rendered Matrix
- Multiple Player Colour Distribution
- Refresh Rate vs Keyboard Polling Rate



AI / Computer Opponent

- General Dumb Algorithm
  - Polls board in a 2 block radius for imminent collisions
  - Collision imminent
    - Make a random valid move that doesn’t kill the player
  - Otherwise
    - Go straight
    - Has a configurable chance to turn randomly



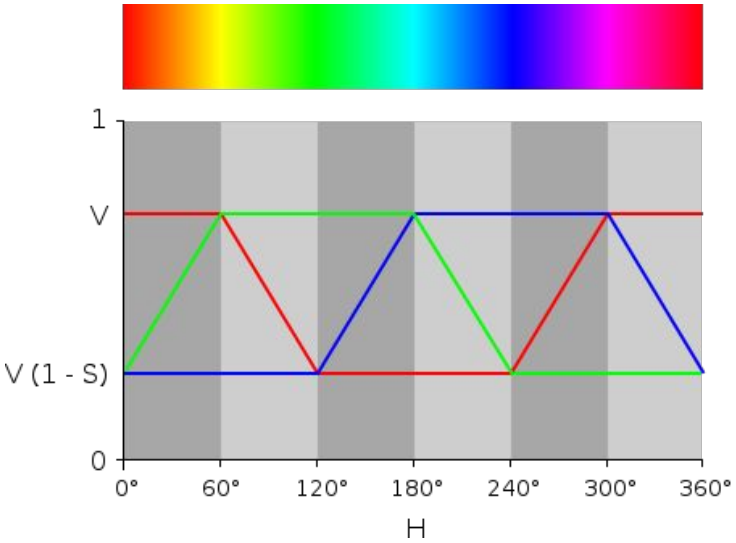
Player Placement

- Given the board dimension *B*, numPlayers *P*, and playerNum *N*:
- To arrange n players in a circle of radius $\frac{3}{8}B$ with center at $(\frac{B}{2}, \frac{B}{2})$
- Each player is assigned to its position:

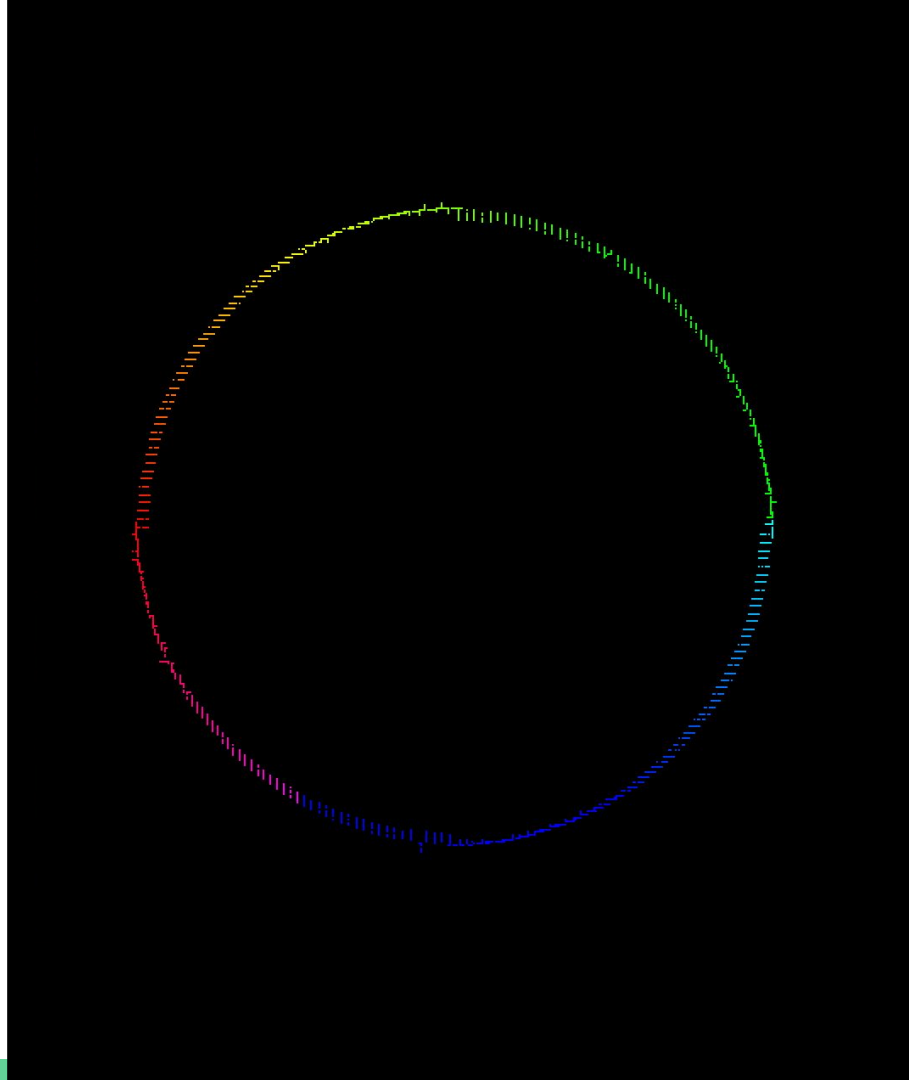
$$(\frac{B}{2} - \frac{3}{8} \cdot B \cdot \cos((N-1) \cdot \frac{2\pi}{P}), \frac{B}{2} - \frac{3}{8} \cdot B \cdot \sin((N-1) \cdot \frac{2\pi}{P}))$$

Colour Assignment

- Given the numPlayers *P*, and playerNum *N*:
- Using Hue Saturation Value representation to project the 1D value of *N* into the 3D space of RGB:



https://en.wikipedia.org/wiki/File:HSV-RGB-comparison.svg



Reflection

What went well...

- Sharing work
- Helping each other
- Efficiency

... and what didn't

- Initial work distribution
- WebPA
- Some time management

Thanks for listening!
We will now show you the game.