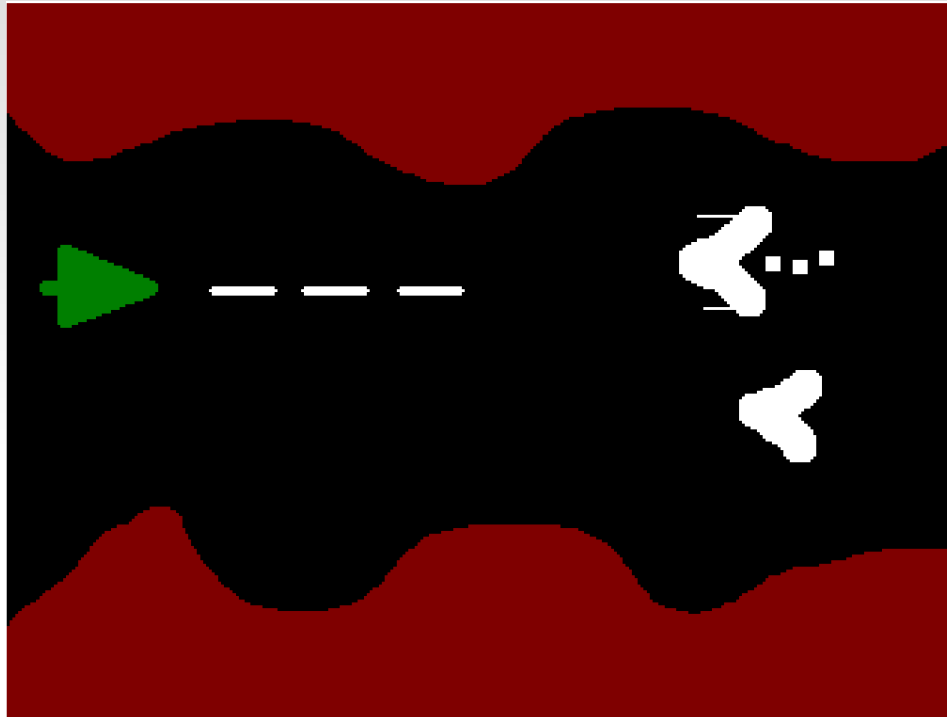
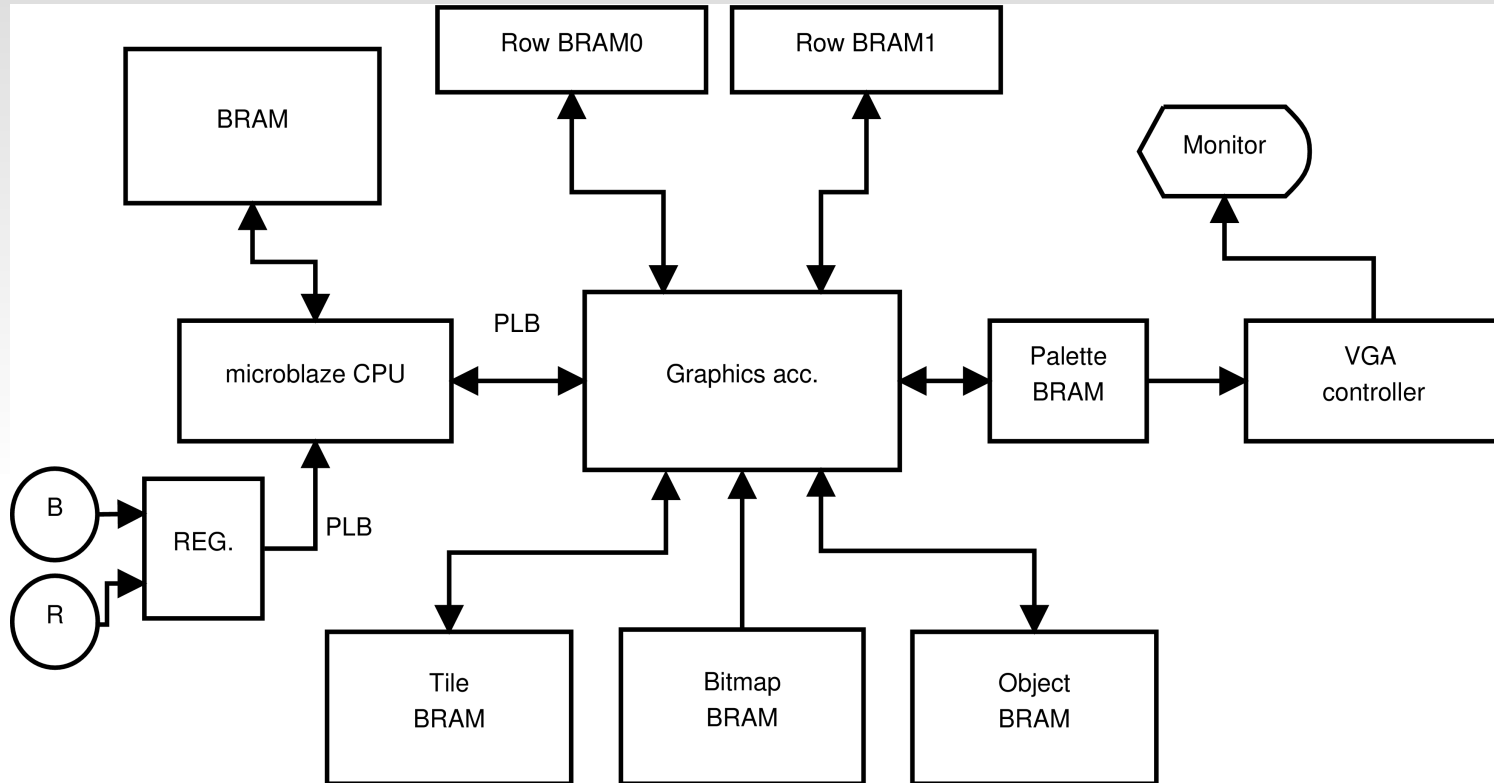


Space game



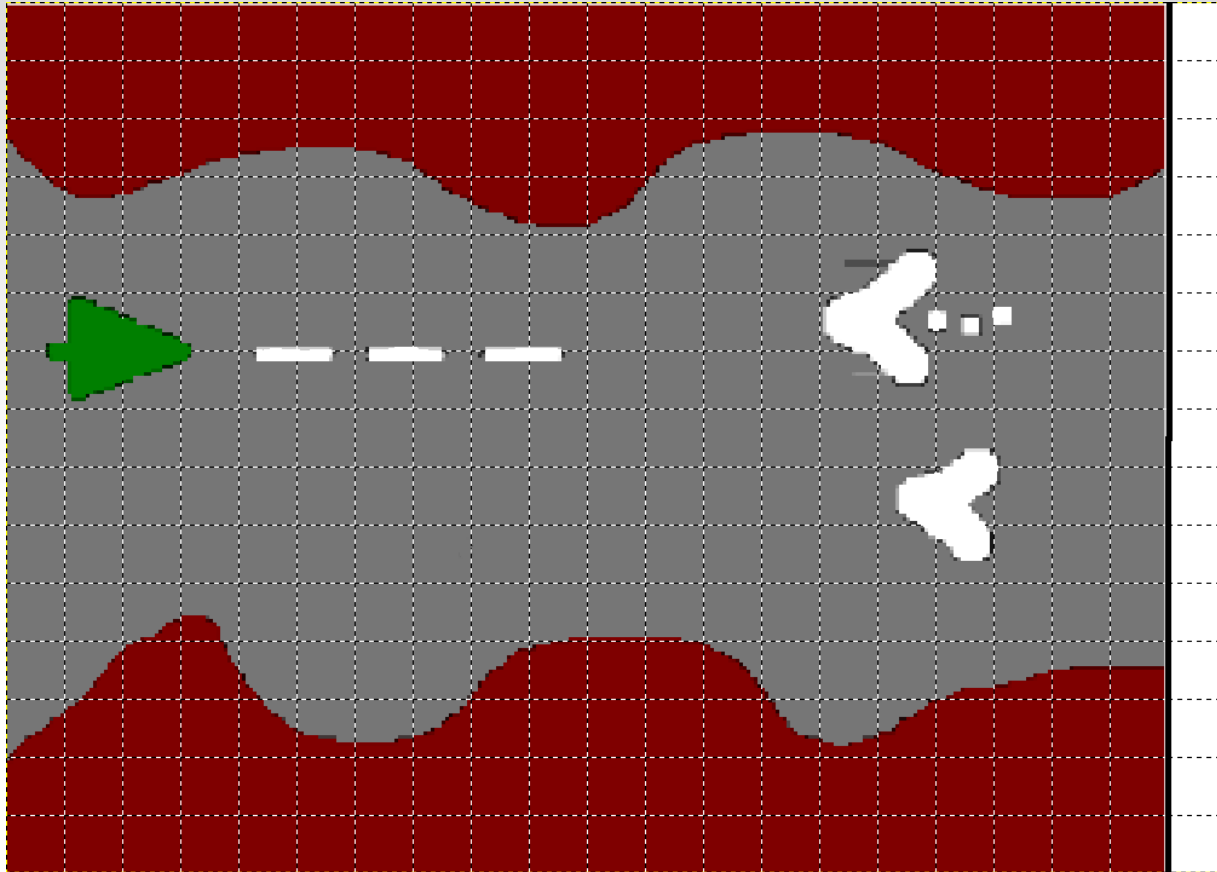
- The idea of the project is to construct a space shoot 'em up style game.
- The player controls a spaceship and has to avoid enemy spaceships and obstacles.

Hardware



- The graphics accelerator does all the drawing to screen.
- The CPU calculates position of background and foreground objects.

Background



- The background consists of tiles placed in a 21x15 grid. Each tile is 16x16 px.
- The grid is one tile wider to allow background scrolling.
- 16x16 px bitmaps are kept in one RAM. What bitmap to put in each tile is stored in another RAM.

Specification

- To control the spaceship a rotary encoder and a pushbutton is used. The controls are read after every screen refresh.
- Screen resolution is 320x240pixel (640x480) @ 60Hz.
- Separate memories and logic for background and foreground.
- Line buffers are used for foreground object rendering.

Time plan

| Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 |
|--|---|---|--|---|--|--|
| Philip & Jonas make project proposal and presentation | Jonas construct skeleton for hardware architecture | Philip construct interface and registers for buttons | Philip & Jonas start with interrupt routine for cpu | Philip calculate positions for objects | Philip finishing collision handling | Philip & Jonas write report. |
| | Philip generate bitmaps for graphics memory | Jonas implement row buffering | Philip Uppdate graph. memories | Jonas algorithms for choosing background and objects | Philip & Jonas testing of whole system | |
| | Philip & Jonas Start construct graph. Acc. possible to draw background | testing | Jonas Calculate scrollvalue | Philip start on collision handling | Jonas start on report | |
| | testing of graph. Acc. | | testing | testing | | |