

ECE532 Project Proposal

Project Title

Interactive Snake Game

Project Team

Tang Tang: 995135438

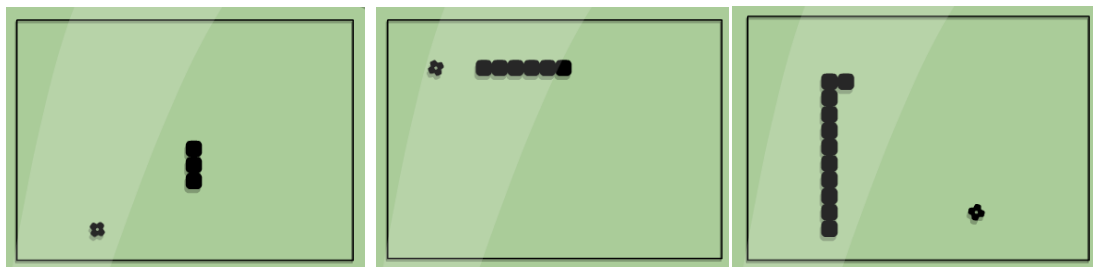
Qiwei Wang: 995380831

Lei Wu: 995386050

Project Description

Background

Snake is a classic single player game. In the game, the player controls a long, thin creature, resembling a snake, to roam around a bordered plane picking up food. Each time the snake eats a piece of food, its tail grows longer. The game ends whenever the snake's head hits the border or itself.



Description

In this project, our group plans to recreate this game with added features. Our goal is to have the player hold and move a physical object (acting as a beacon) to attract the movement of the virtual snake, which is being projected onto a screen. The beacon could be in the form of a piece of coloured cardboard of varying shape, with different color/shape signalling different response from the snake (e.g. varying movement speed). The goal of the player is to physically move the beacon to different positions in front of the screen to guide the snake to its food, while at the same time avoid having the snake hitting itself or the screen border.

Features

- **Major:** Basic Game rules:
 - Snake head moves at constant speed towards the beacon; snake body follows the trace of snake head
 - Score by directing snake head to touch specific object on screen

- Snake grows longer when scored
- Game-over condition: snake head hits itself or any obstacle on the map including screen borders
- **Major:** Real-time tracking of the beacon
- **Major:** Real-time projection of the snake
- **Major:** Detection of the beacon (color, location, etc.)
- **Major:** “Game Start” and “Game Over” Screen
- Automatic detection of screen size, calibration of beacon location
- Video effects for objects on screen (Snake, scoring object, obstacles, beacon, etc.)
- Sound effects for events in the game (game start, scoring, game over, etc.)
- Multiple difficulty levels (snake speed, obstacles on the map, starting snake length etc.)with a menu for selection at “Game Start” screen
- Scoring system (scores obtained and/or play time); display scores at “Game Over” screen
- Game Pause/Resume when beacon out of screen (cannot be detected by camera)

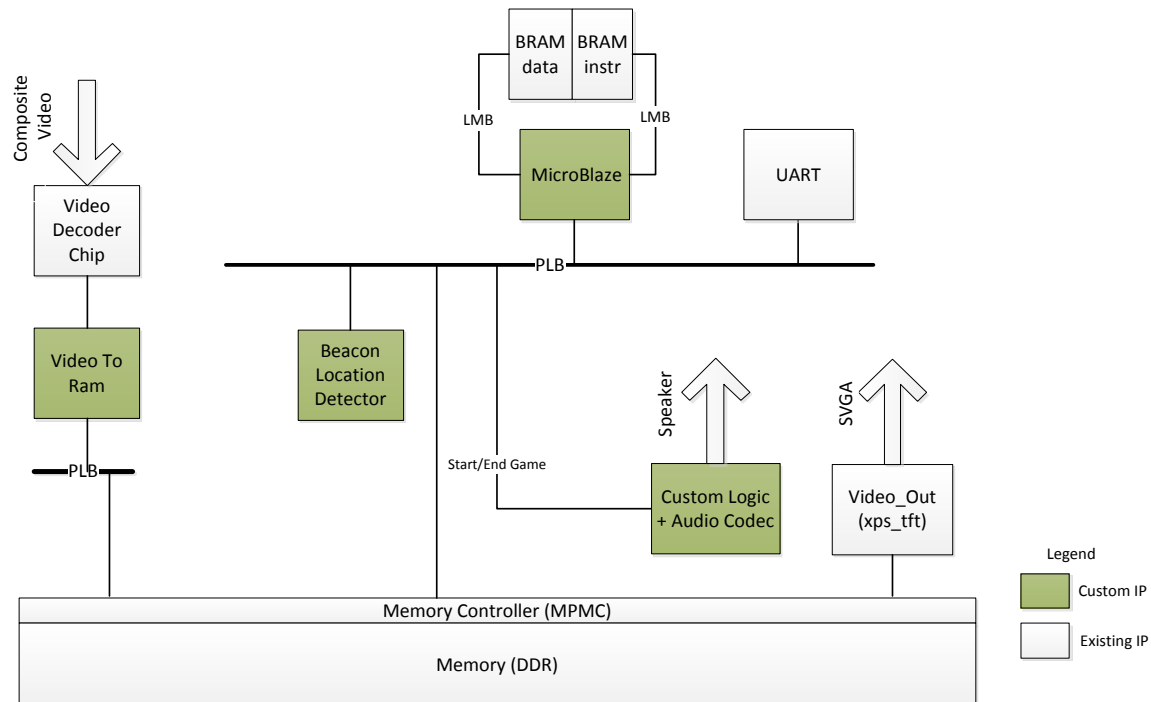
Specifications

- Hardware implementation on Xilinx XUP Virtex-II Pro Development System with video decoder board from Digilent
- Camera with composite output for monitoring the movement of the beacon
- VGA output to monitor or projector
- Audio output using standard 3.5mm jack
- Game runs smoothly (at least 30fps)
- System powered by standard 110V 60Hz power source

Acceptance Criteria

- Deadline: final demonstration on April 7th
- All major features implemented and tested

Block Diagram



Description of IP

- Video Decoder (VDEC1) - Converts the frame by frame composite video signal captured by the camera into digital signal. Digital signal is then fed to Video to Ram block.
- Video to Ram [Custom] – Stores digital video signal into DDRAM through MPMC.
- Memory Controller (MPMC) – Multiport memory controller that regulates Memory Read/Write operations. In total 4 ports are used.
 - One port is used by Video to Ram block to write captured incoming video signal to memory.
 - One port is used by Beacon Location Detector to read the captured incoming video signal from memory.
 - One port is used by the MicroBlaze processor to write the generated outgoing video signal to memory.
 - One port is used by Video_Out block to read outgoing video signal from memory for VGA output.
- Video_Out (XPS TFT) - Hardware display controller to handle the signal to VGA output.
- Beacon Location Detector [Custom] – Reads incoming video signal from memory, processes the signal to identify the relative location of the beacon, stores the processed result in register/memory.
- Custom Logic + Audio Codec [Custom] – Handles the signal to audio output.
- MicroBlaze – stores game logic (implemented in C or assembly). It is responsible to generate the outgoing video frame based on snake position and beacon location. Information about beacon

location is obtained from the result calculated by Beacon Location Detector. Information about snake position is obtained from previously stored game states. Game-over conditions are constantly checked. The controller also communicates with MPMC to store the generated video signal to memory, and with Audio Codec to generate appropriate audio outputs.

- VGA Output - Output to a monitor or a projector.

Resource Requirement

- Xilinx XUP Virtex-II Pro Development System with video decoder board from Digilent
- VGA monitor/projector
- Video Camera with composite output; a real-time display of captured content is preferred
- Speaker

Milestones

Feb. 2, 2011	Lab Test.
Feb. 9, 2011	VGA video output module completed, able to output coloured screen to monitor; Basic functions of memory controller (MPMC) modules completed and tested, able to read/write data from MicroBlaze processor.
Feb. 16, 2011	Able to load data from PC to on-board memory, implemented “Game Start” and “Game Over” screens; Start working on Video Decoding module and Video to Ram custom logic.
Feb. 23, 2011	<i>(Reading week)</i>
Mar. 2, 2011	Able to store captured frames to memory with 2 channels going parallel for read and write; display captured frame on VGA output, i.e. show what’s captured by camera on monitor; Beacon Location Detection module completed; able to display location of beacon on VGA output; Start C program for snake game on PC.
Mar. 9, 2011	A preliminary game, “Game Start” screen -> a moving snake -> “Game Over” screen, is playable; able to detect game-over condition (snake head hits screen border or hits other parts of the snake).
Mar. 16, 2011	Audio Codec module completed and tested; able to load sound data to memory and play it; audio effects added to the game.
Mar. 23, 2011	Optional features such as auto calibration (software), multiple difficulty levels (snake moving speed and/or obstacles on map), game pause and resume (game pause and display pause message if cannot find beacon on screen; game resumes when beacon is back) are added to the game; a working game with some bugs is completed.
Mar. 30, 2011	ECE Design Fair; All features completed and tested by now. The rest of the term is for additional bug fixes and report writing.
Apr. 4, 2011	Individual and group reports due.
Apr. 6, 2011	Final demonstration.