

# AirHockey

Final version for course "Introduction to Game Programming 2013"

Niklas Nyholm

Mikko Rautiainen

Rostislav Skudnov

Alexey Sofiev

## Contents

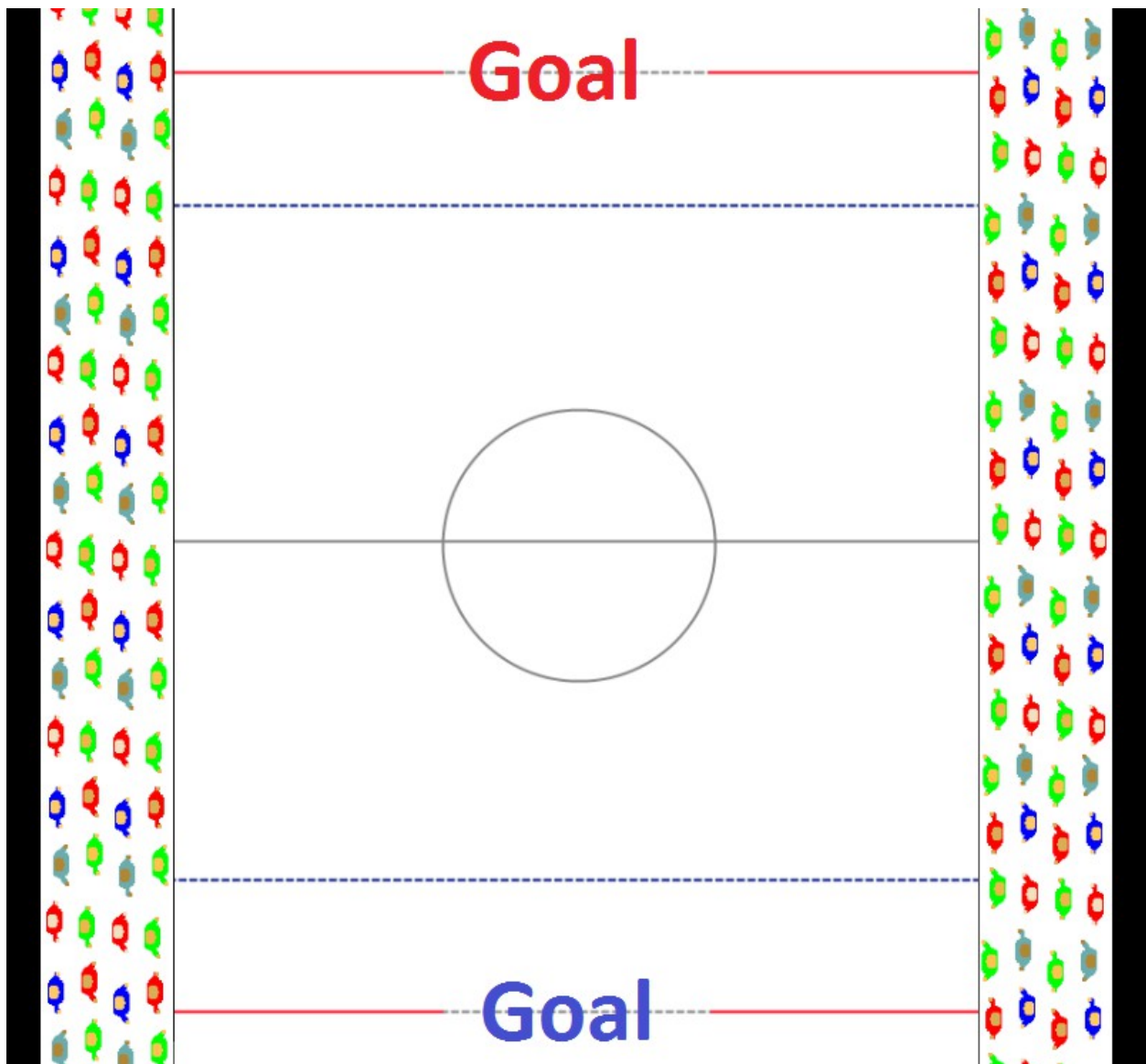
Introduction.....	3
Goal.....	3
Controls.....	4
Features.....	4
3 gamemod.....	4
Multitasking.....	4
Speed.....	4
Sounds.....	5
How to start.....	5
Python-version.....	5
Exe -version.....	5
Structure.....	6
General structure.....	6
Multiplayer.....	8
Basic idea of multiplayer.....	8
Server-client communication.....	8
AI principle, tips and hints.....	9
Local multiplayer.....	10
Contacts.....	10
Extra.....	11
Building executable from source.....	11

## Introduction

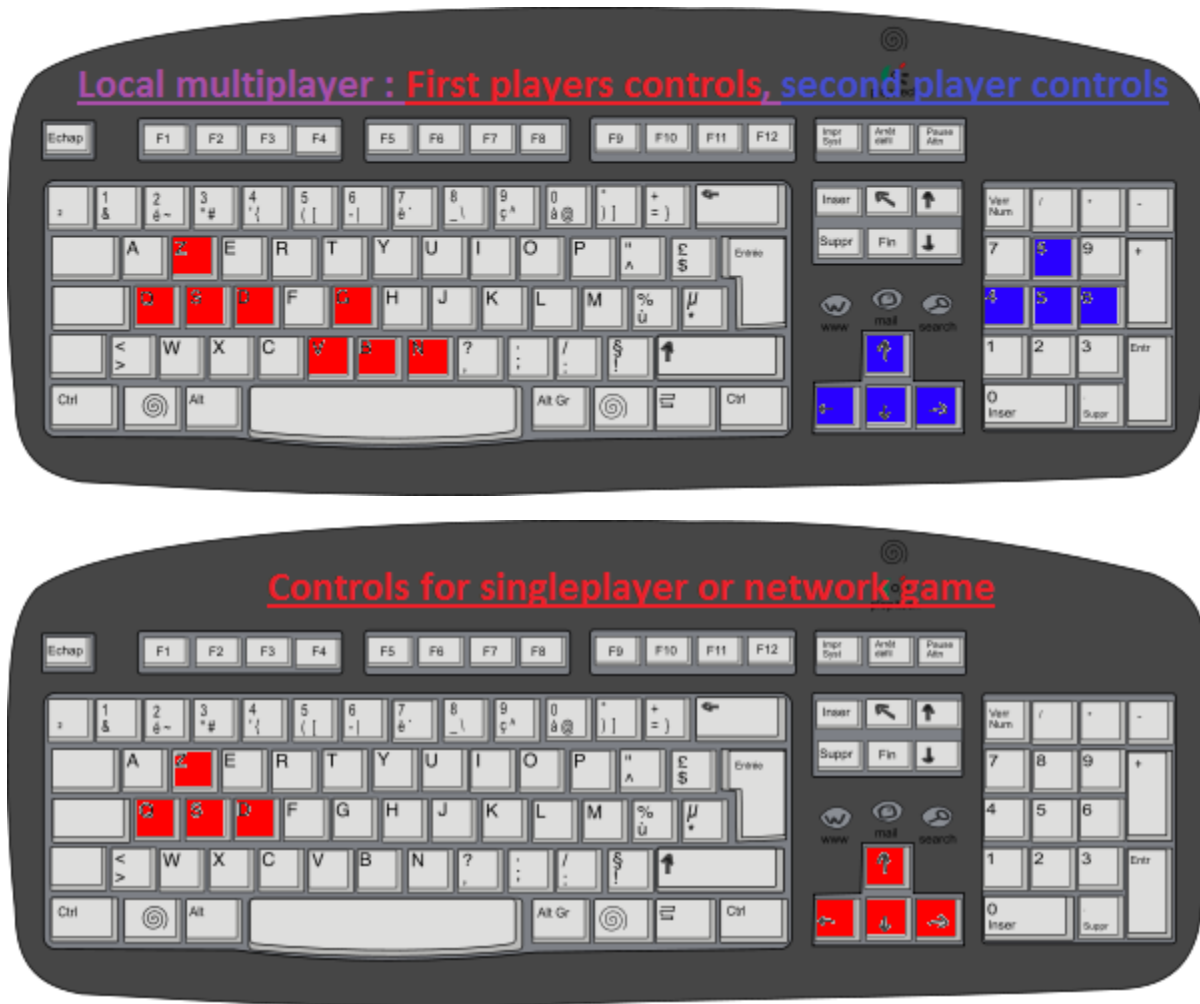
AirHockey is a fast-paced multitasking game, which will provide a pleasure challenge for few minutes. You may play against the AI, against another player on the same computer or over the Internet. Push your multitasking skills to the limits by controlling two paddles at the same time and crush your opponents!

## Goal

The purpose of the game is to push the black puck into the other player's goal. One player controls the two red hockey paddles, and the other controls the two blue paddles.



## Controls



## Features

### 3 gamemodes

- Network game
- Local Multiplayer
- Singleplayer vs AI

### Multitasking

- 2 paddles to control
- Powerups

### Speed

Your opponent will barely sleep, so hurry up!

### Sounds

Not required for successful gaming, but might help. (whistle at the start.)

## How to start

### Python-version

Run `__init__.py` with python to start the game. Run `server.py` to start a server

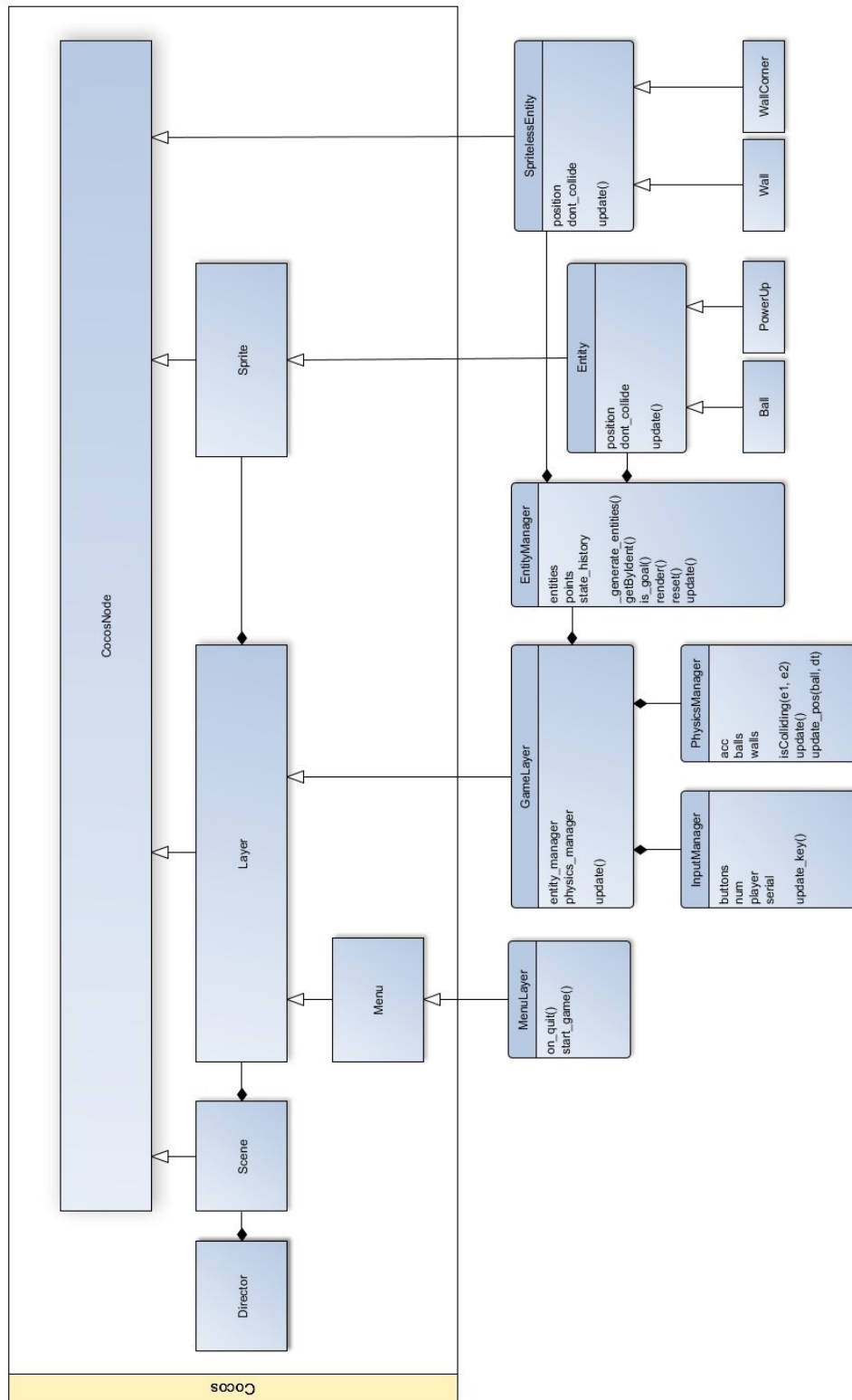
(Cocos2d library required)

### Exe -version

Run `AirHockey.exe` to start the game. Run `server.exe` to start a server

# Structure

## General structure



The game architecture is based on Cocos2d for Python.

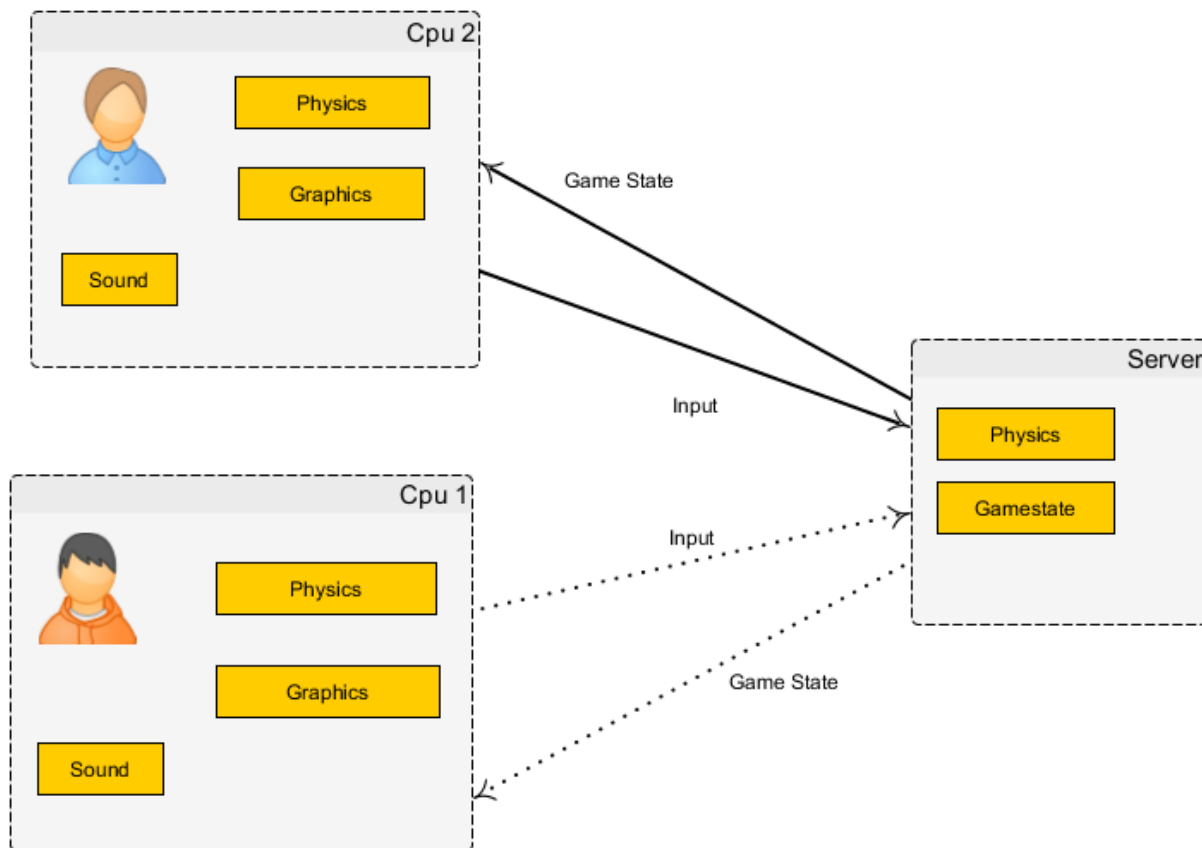
Most of the classes inherit from Cocos2d superclasses. For example the MenuLayer class is an implementation of the Menu class and GameLayer inherits from Cocos2d Layer.

The GameLayer class is where the gameplay happens. It has an EntityManager, an InputManager and a PhysicsManager. There are two types of entities: Entity and SpritelessEntity. Entity inherits from Cocos2d class Sprite, while SpritelessEntity is only a CocosNode. Entity class is parent to Balls and PowerUps. SpritelessEntity is used for walls, but also for all Entities server-side since no visual representation is required.

The Cocos2d Director class handles switches between different Scenes and Layers.

## Multiplayer

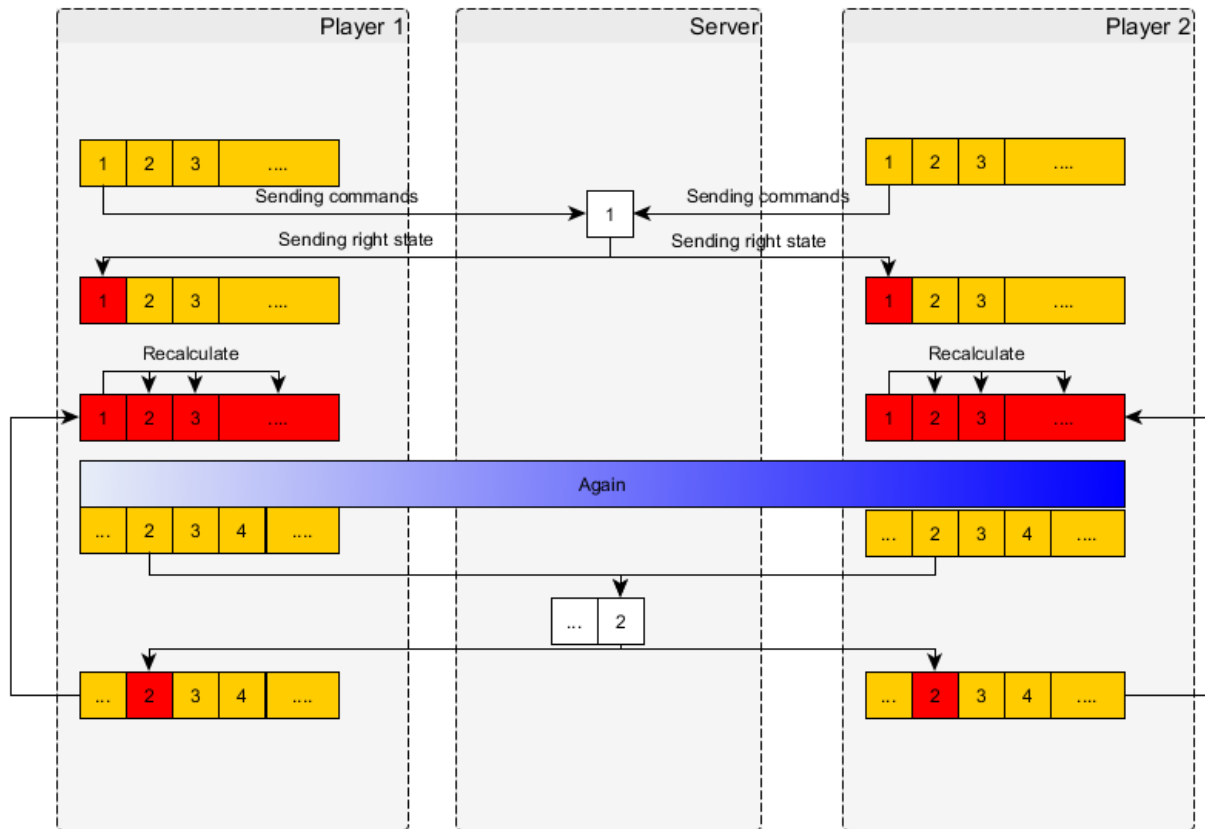
### Basic idea of multiplayer



### Server-client communication

Each client side takes inside an input and generate gamestate, which has all information of this game inside. However, this client side has no information about opponent's actions, so all input is sent to server, who calculates the real situation and forwards it back to players.

There is a small delay, so following picture demonstrates how we handle it.



Each computer calculates on constant speed coming frame number. Then for example we send information from state 1 to server. At moment when we receive the response of real situation at state 1, we are already at later state, so we recalculate all basing on the fixed situation of state 1. Then we select next state, and repeat all this operation.

### AI principle, tips and hints

Compared to previous in case of singleplayer mode, we don't start any server communication. Player's commands are recorded to gamestate and are modified by adding AI actions.

AI has 3 difficulty levels 0-2

- Level 0 is just going after the ball
- Level 1 is clever aggression.
- Level 2 will put 1 ball to be a goalkeeper, while second one chases the ball.

Due to fact that few found it rather hard to beat even Level 0 AI, we provide a spoiler, of how to win AI.

Best working strategy is just to stop the ball, and push it behind AI balls. This is not necessary a win, but most likely at least vs lvl 0 AI. Lvl 1 AI can be beaten even with 1 ball, just go to the middle and then straightly top.



## Local multiplayer

Similarly to singleplayer mode, here is no communication with server. Input from both players are taken, resulting in gamestate, which is showed at the screen.

## Contacts

Nyholm, Niklas

nyholmniklas@gmail.com

Rautiainen, Mikko

m\_rautiainen@hotmail.com

Skudnov, Rostislav

rostislav.skudnov@cs.helsinki.fi

Sofiev, Alexey

alexey.sofiev@helsinki.fi

## Extra

### Building executable from source

If you don't want to use prebuilt option or python version, then here is our guide of how to compile:

1. copy src/ to new folder compile/
2. comment compile/\_\_\_init\_\_.py lines 3-7
3. comment compile/server.py lines 5-7
4. copy libs:  
autobahn-0.6.4/autobahn -> compile/autobahn/  
cocos2d-0.5.5/cocos -> compile/cocos/  
pyglet-1.1.4/pyglet -> compile/pyglet/  
twisted-13.1.0/twisted -> compile/twisted/  
zope.interface-3.8.0/src/zope -> compile/zope/
5. copy files:  
compile/autobahn/resource.py -> compile/resource.py
6. uncomment compile/setup.py line 4 '\_\_\_init\_\_.py'
7. in compile folder: python setup.py py2exe
8. copy resources:  
compile/res/ -> compile/dist/res/
9. rename compile/\_\_\_init\_\_.exe -> compile/AirHockey.exe
10. copy compile/dist/ -> dist/
11. comment compile/setup.py line 4 '\_\_\_init\_\_.py', uncomment line 5 'server.py'
12. in compile folder: python setup.py py2exe
13. copy compile/dist/server.exe to dist/server.exe

Now dist/ has distribution version.

"access denied" on py2exe running: try again