Project Game Design Document (Initial)

Team 06: Air Hockey for PixelSense

This document contains higher level design documentation about Air Hockey for PixelSense.

Note: Since the project scope changing during development, this document contains the initial scope, and is now mainly for record keeping purposes. Many of the themes and details in this document still apply.

Contents

GDD Section 1 - Game Design

Air Hockey for PixelSense Overview

Target Platform: Microsoft Pixelsense
Other Possible Platforms: Tablets (and other multi-point input platforms)
Targeted Audience: Everyone
Target Purpose: Demonstrations and Showcasing at Events, Research
Genre: e-Sports, Puzzle
View: 2D, Top Down

Description:
Classical air hockey is a game for two competing players trying to score points in the opposing player's goal using a table having a special low-friction playing surface.[Wiki]

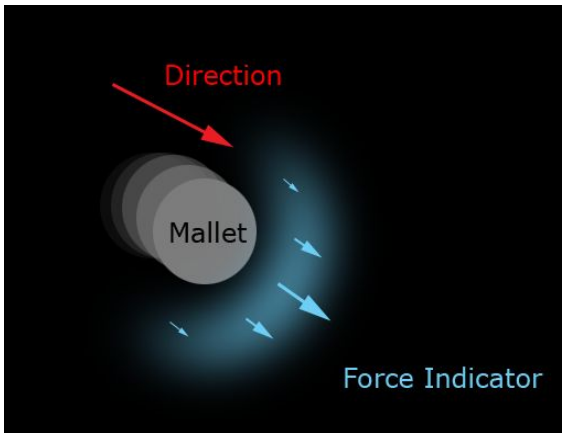
Figure 1: Air Hockey table with 2 red mallets and a red puck.

The players can hold onto the Mallets and must try to hit the puck to have it enter the other player's goal - to which they are protecting/defending.

Air Hockey for PixelSense will adapt most of the same concepts as classical Air Hockey, and will extend and include additional audiovisual themes.

To enable further expansion on the concept of Air Hockey, some basic mechanics that are derived from Classical Air Hockey will be modified to consider hardware limitations of the PixelSense (an example is that direct collisions between solid objects in the game will be avoided when interacting with a player controlled object).

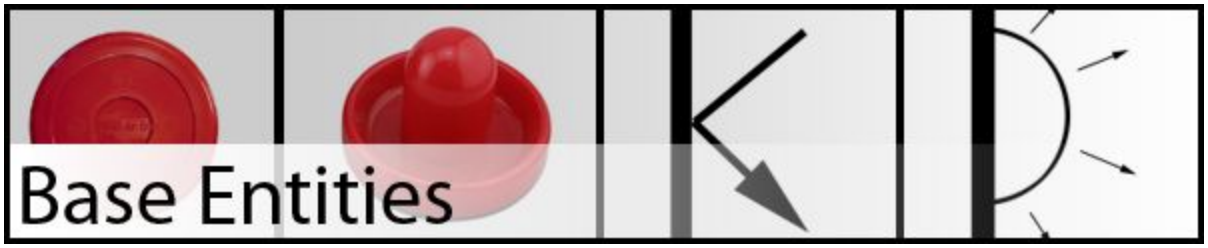
A key difference in this game is the way in which the pucks and mallets interact. Classically it would consist of direct collisions, however in this game the interaction will use a kind of 'magnetic' force to push the puck across the playing area (see figure to the right).



Some mechanics may change depending on theme, however they will only be added on top of

the foundation features if time permits in this project.

1.1 - Base Entities



This section outlines the main base entities in Air Hockey for PixelSense.

1.1a - Puck

Description:
A solid disc that bounces around the playing field and that players hit using their mallets. When a puck hits a player's goal, the opposing player scores a point.
PixelSense:
The puck will not have a physical representation for the PixelSense.

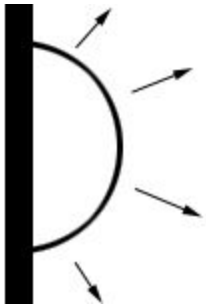
1.1.1 - Mallet (also known as the Goalie/Striker/Paddle)

Description:
A disc with a handle in which the player controls to hit the puck with.
PixelSense:
The Mallet will have a physical object with a marker (Microsoft Tag) which will enable the PixelSense to obtain information such as position and orientation.

1.1.2 - Walls

Description:
Objects in which the Puck and Mallets can collide/interact with.
PixelSense:
This will have no physical representation for the PixelSense. Optionally, it may be possible to have the Puck interact with objects with tags, to enable custom collision objects and shapes.

1.1.3 - Goal / Core

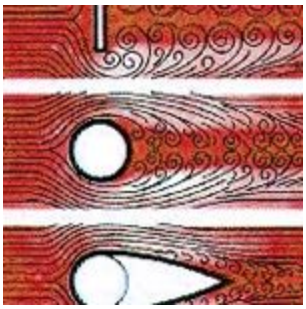


Description:
An edge area or object which the player needs to defend. Defending is using the Mallet to keep the Puck away from one's goal/core.
A core is a free moving entity like a sphere. Pucks that come into contact with cores will destroy it and the other player wins the game.
PixelSense:
The PixelSense will not have any physical object for the goal.

1.2 - Additional Entities

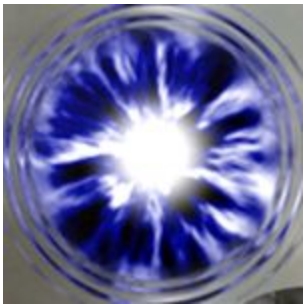
This section outlines entities that would not normally be found in Classical Air Hockey. Typically these entities can appear temporarily throughout gameplay, and disappear after a time.

1.2.1 - Drag Zones



Description:
Drag zones are areas in which as the puck moves through, will have a reduced velocity. Once the puck leaves the area, it accelerates back to almost its original velocity.

1.2.2 - Thrust Zones



Description:
Thrust zones are effectively the opposite of drag zones, speeding up the puck as it passes through the area. Once it leaves, it partially decelerates to almost its original velocity.

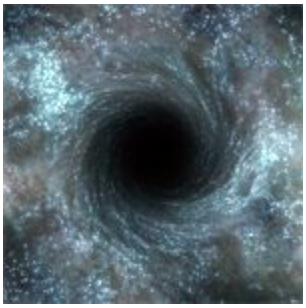
1.2.3 - Jolt Sphere

Description:
These single/small points contain an element of unpredictability. When touched by a puck, these will cause the puck to go in a random direction, at a random speed.

1.2.4 - Wormhole

Description:
These points will draw in a puck when it is in very close proximity, and will release it in a random location on the screen with the same velocity as it had when it entered.
Wormholes will not create exits too close to the goals/cores or mallets.

1.2.5 - Singularity / Gravity Well



Description:
Singularities work by pulling the puck in towards it, like a gravity well (or black hole). Any pucks heading directly towards the center of this will be greatly accelerated while it is pulled in.
Any trapped pucks will be transported to another area on the screen, similar to the exit point for wormholes - except with an increased velocity and random direction.

GDD Section 2 - Themes and Style

This section outlines the skins/themes to potentially be covered in this project. As a targeted minimum, 2.1 or 2.2 can be used.

2.1 - Retro Neon

- This plays upon the classic neon lights theme.
- Highly saturated and bold colours and a glowy look.
- Dark/Black background, which may be reactive (pulses of intensity when puck collides)
- Still uses the vector/forces with the mallets, and not solid collisions with them.

Example playing field image

- Interactions with other objects, such as being affected by the vector force, wall collisions, and simply drifting around will utilize glow trails (such as with the puck, above), or neon spark-style particle systems.

Example of particle system look

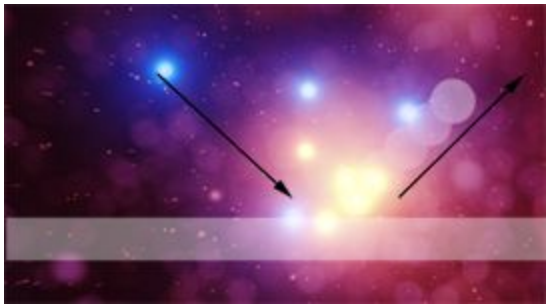
- Audio will consist of sine/sawtooth sampling and synthesized effects

2.2 - Astronomical Void

- Derived from an astronomy theme, which gives stronger context to the interactivity change regarding vector forces rather than solid collisions with the puck.
- Saturated colours, with nebulous clouds and stars in the background
- Possibly add sprites into the background that fly by throughout the game

Example background image theme

- Collisions and interactions will involve rounded, nebulous particle system effects, with an additive brightness effect, simulating exposure mechanics found in telescopes.



Example interaction of a puck bouncing off a wall

- Audio will consist of synthed tracks and sine/electronic audio effects.

2.3 - Echolalia X-Ray

- Inspired by the PixelSense's debug mode where it shows on screen what the PixelSense is 'seeing'.
- Washed out, xray-like theme (With cyan/blue tint)
- Horizontal scanline effect/touch, with noise/haze across the game screen.

Example game screenshot, also showing a hand detected

- Collisions and effects will involve hazey additive blending particle effects and 'static noise/electricity' imagery.

Example effects for puck and wall interaction

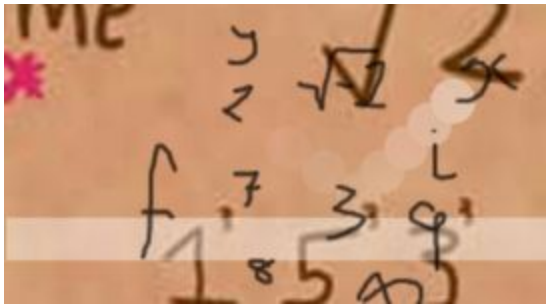
- Audio will involve static and noise, and potentially dissonant/abstract sounds,

2.3 - Numberphile

- Derived from a paper cutout style, combined with a mathematics theme.
- Theme will follow Brady Haran's Numberphile program, using butcher's paper brown, with darker brown markers.
- Also will include ViHart's triangle characters and style for added flavour.

Sample of Numberphile visual style

- Interactions with other objects and walls results in math related symbols as the particles.

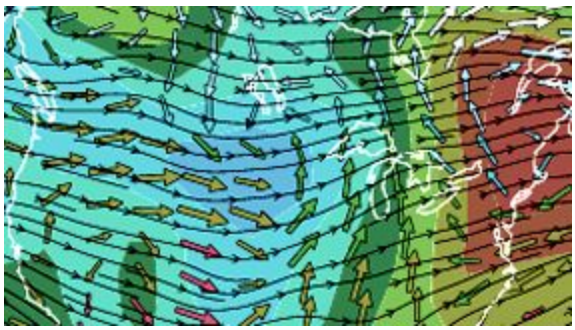


Example of number particles after a wall collision

- Audio will have sounds relating to stationary, paper, and wood.

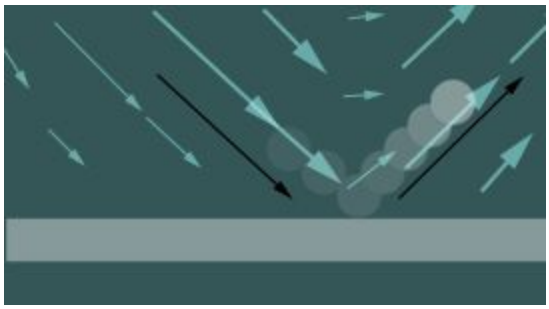
2.3 - Dynamo

- Inspired from vector field maps, like geographical wind direction maps
- Will have an array of dots, which will react to local ‘forces’ (such as the puck, and mallet) and indicate a direction of motion by either a line or an arrow.
- Will not have a coloured background like the example below - and simply will use black as the base.



Example of a vector field representation on a geographical weather map

- Interactions with objects will be depicted by the animation and motion of the grid of arrows.
- Additional arrows which point away from the origin of interaction can be used as particles for added effects.



Indication of how the arrows may point soon after collision with a wall

GDD Section 3 - Controls and Usability

3.1 - Controls

3.1a - Controllers - Touch and Physical Mallet

A physical mallet held by the user. It will have a tag/marker of some sort to enable PixelSense to 'see' the mallet and create a virtual mallet to be tracked.
There could be the option to add additional UI elements that users can touch to activate items/menu's etc.

3.1b - Design to Consider Input Lag

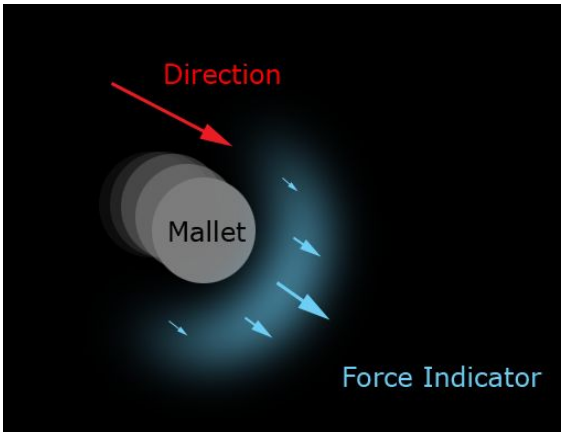
The link below demonstrates the PixelSense showing some input lag for the pucks. Occasionally you can see players moving their mallet to hit he puck only to have the PixelSense respond too slowly.

Microsoft PixelSense Air Hockey Demo

It could be possible to move the puck based on velocity to attempt better synchronisation of the PixelSense display and the physical mallet itself. However this may not be able to accurately calculate where situations where abrupt accelerations and decelerations occur.

3.2 - Radial Pressure/Force and Vectors

One way to enable players to maintain a consistent cognition of the mechanics of the game with the input lag is to have collision points not be of a solid surface nature.



A means of doing so can be to use a radial force emitted from both mallets which affect the velocity of the puck, and have the level of effect on the puck increase exponentially as the distance between the puck and mallet decreases.

This can eliminate the obvious situation where input lag can be seen. In terms of visual feedback, a glowing force can fade in to show registered movement.
This will avoid the situation where direct reflection between the real mallet and the virtual mallet has some identifiable lag.

The image above shows the direction of the mallet movement (red) and an indicator showing the change in force as a result of the moment (blue). This then allows the flexibility to remove a solid indicator for the mallet, and eliminates obvious input lag indicators.

GDD Section 4 - Game States

Default State

Default state for the game can be a screensaver or ‘touch here’ kind of thing. Similar to arcade start screens.

Main Menu

- Select Options/Settings
    - Allows customization of settings such as puck speed, difficulty, level (if any), layout, etc.
- Start Game
    - User clicks to start the game
- About
    - Might include an about section for credit if allowable

In-Game

- Paused/Countdown before game starts
- Playing/Live - where the player plays
- Scored/Goal - when a goal is scored.

Post Game

- High Score/Options to play again, etc...