

ICEBREAKERS++

DESIGN DOC

Written by Keano Raubun

28/03/2014

GAME SUMMARY

Peripherals: PingPong++ & Xbox Controllers

Players: 3-6

Icebreakers++ is a multiplayer game played using the PingPong++ and several Xbox controllers, with two players standing on opposite ends of the table, whilst the others stand to the sides. The game itself is projected on the surface of a table tennis table, and can detect the impact of a ping pong ball hitting its surface.

In the game, players are divided into two teams; taking the role of a penguin or as an ice castle. The goal of the penguins is to blow up both ice castles by gliding around on the ice and collecting and dropping off bombs, while the castles defend themselves by throwing ping pong balls on the table's surface, trying to break the ice or hit the penguins directly.

One of the two castle players needs to tap the ball inside of the circle in order to start the game. This happens only when at least one player has joined using the controller. When the circle is tapped, a countdown begins and the penguins spawn on the map. The game ends if all castles are destroyed, or the penguins have no remaining lives. After the game ends, players are given their score and can return to the starting screen.



Pre-game UI & player positioning. The blue players are castles, while the others are penguins.

PLAYER CHARACTERS

Players take control of either a single penguin or one of the two castles.

PENGUINS

Main Objective: Collect bombs and destroy the castle.

Secondary: Collect pick-ups to increase their score and be the best player.

- Penguins start with three lives.
- Penguins die when they fall into water, get hit by a ping pong ball, or blown up by a bomb.
- Penguins will crack the tiles they are on if they wait too long, so they have to keep moving.
- Penguins can collect items/power ups to increase their score.
 - o Collecting 100 points will earn them an extra life.

- Movement
 - o Penguins glide slightly over the ice
 - o Penguins can bounce into one another
 - o Penguins bounce off the edges of the map

CASTLES

Main Objective: Defeat all penguins and defend the castle.

Secondary: Destroy pick-ups to hinder the penguins.

Castle players are able to defend the castle by throwing ping pong balls onto the table's surface, killing penguins or destroying ice in the process. Only the opposite half of the playing field relative to the castle player with the ball can be damaged at a time, to ensure that the game remains difficult for the Castles.

Castles have 5 hit points, which can only be depleted if a bomb is planted and detonated, or when struck by a ping pong ball.

GAMEPLAY ELEMENTS

ICE

The floor of the game consists of breakable ice tiles. The tiles have 2HP and can be damaged by being hit by a ping pong ball or a bomb explosion. Tiles regenerate after 20 seconds of not receiving damage.

BOMBS

Bombs will spawn from the edges of the playing field. Penguins need to pick up these bombs and carry them to the castle, but will move much slower when doing so. Bombs are automatically dropped by Penguins when they are in a one-tile radius of the castle. Damaging the castle award penguins with 100 points.

Castle players can blow these up if they hit them with a single ball, destroying several ice tiles and penguins within range.

FISH

Occasionally, fish will be thrown onto the ice field which Penguins can pick up to earn 50 points. If a penguin reaches 100 points, they will gain an extra life.

TECHNOLOGY & INTEGRATION

The PingPong++ uses 8 sensors under the table to triangulate the ball position using the vibrations in the wood. The sensors are attached to an Arduino. A program running on this Arduino sends the ball position data to the computer running the game using a network connection. To make sure that the ball position corresponds to the same position on screen the sensors and display will have to be calibrated.

We used the input from the ping pong balls to have two teams interact with the game in different ways, as well as using the balls to navigate the game's menus. By having the castle players use the ball input instead of an alternative controller, it results in the game having a higher skill ceiling and playing speed due to a lack of artificial constraints that would need to be set in place due to the precision of using a mouse or controller.

RISKS & DIFFICULTIES

- ***Table is currently not set-up, and as such we have not yet been able to test the table. Not getting the table to work or taking too long to get the table to work is the biggest risk we have.***

Restoring table functionality took us several weeks, due to sensors being broken and requiring replacement. In the end, we managed to restore functionality to only one half of the table, since replacement parts would arrive past the project deadline.

- ***The lack of a suitable projector to project onto the surface of the table will prevent us from testing.***

The green room had a short throw beamer that projected perfectly onto the surface of the table tennis table.

- ***When the code to receive and decode the ball position messages was added to the project it caused Unity to crash for unknown reasons. This also caused the port used to receive the messages to be blocked. A return of this bug would create a big risk, as we don't know what causes it.***

This issue was fixed during the first two weeks by rewriting the code and using the source code provided by the original developers of the PingPong++.

- ***The surface of the table is dark blue and as such colors will appear different and some will not be visible at all.***

This was resolved by having high-contrast colors projected onto the surface of the table, and tweaking to find the ones that were most clear.

KNOWN FAULTS & IMPROVEMENTS

- Half of the table is not working due to broken circuitry. Parts are already underway.
- Color reproduction is off because of the blue surface of the table and low brightness of the beamer. A white surface would improve colors.
- When the table is moved during gameplay the hit detection will seem wrong, this could possibly be remedied by attaching the beamer to the table.
- Circuitry is a big mess of wires and ICs are not in sockets so they can be easily replaced. Having the PCB that MIT uses would have probably saved a lot of time.

TESTING & ITERATIONS

Because of the extra time spent on getting the table to function, and ultimately ending up with only a functional half, the amount of iterations is fairly limited; mostly due to the game being fairly simple in nature as well as the integration of the technology. However, there have been several changes as the result of tests conducted and player feedback.

1. Regenerating ice tiles and the ability to jump over water was added after penguin movement became nearly impossible as the game progressed and pick-ups started to fall into water.
2. To increase the challenge of playing a penguin, we added a stun mechanic where penguins become immobilized when bumping into walls or other players, making it easier for Castle players to hit the penguins.
3. Originally, the playing field was composed of more tiles. After witnessing the occasional inaccuracy of the hit detection, we decided to increase the size of the tiles to decrease apparent errors.
4. The castle used to be positioned in the center of the map, but now there are two castles located at each end of the table to reflect the castle player's positioning and making it more difficult for penguin players to damage the castle.
5. The ice tiles have had their HP decreased from 4 to 2, to make the field break sooner and making it easier for castle players to kill penguins.