|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Flappy  Penguin | Bildschirmfoto%202016-06-24%20um%2011.34.39.png | | Bildschirmfoto%202016-06-24%20um%2011.39.31.png | Bildschirmfoto%202016-06-24%20um%2011.32.49.png | |
| HCI-Abschlussprojekt  Gernot Brunner - Stefan Cimander - Martin Hartmond - Thomas Weber |

# Projektidee

## Spielkonzept

Der Spieler spielt einen Pinguin, der unter der Wasseroberfläche verschiedenen Hindernissen wie zum Beispiel Eisschollen ausweichen muss. Analog zum bekannten Spiel „Flappy Bird“ kann der Pinguin lediglich nach oben gesteuert werden und sinkt anschließend von alleine wieder ab. Die Hindernisse bewegen sich von der rechten Seite in den sichtbaren Bereich hinein, bei einem Zusammenstoß des Pinguins mit einem Hindernis ist das Spiel zu Ende.

Als zusätzliches Spielelement stehen dem Spieler nur eine begrenzte Anzahl an Luftblasen zur Verfügung, die mit der Zeit vom Pinguin verbraucht werden. Auch Luftblasen erscheinen zufällig im Spiel und lassen sich aufsammeln. Außerdem kann der Spieler den Vorrat an Luftblasen wieder auffüllen, indem er den Pinguin mit dem Kopf über die Wasseroberfläche manövriert. Sind alle Luftblasen aufgebraucht ist das Spiel ebenfalls vorbei.

Der Spieler bekommt Punkte für die zurückgelegte Strecke und überwundenen Hindernisse. Das Spiel ist als open-end Spiel konzipiert, einziges Ziel ist es, so weit wie möglich zu tauchen und eine maximale Punktzahl zu erreichen.

## Erweiterungen

Um dem Spieler eine interessantere Herausforderung zu geben, die ihn auch körperlich etwas fordert, wird der Spieler mit einem Atemsensor mit dem Computer verbunden. Die Idee ist, dass bei jedem Einatmen des Spielers eine Luftblase verbraucht wird, falls sich der Kopf des Pinguins unterhalb der Wasseroberfläche befindet. Ist der Kopf oberhalb der Wasseroberfläche, kann der Spieler durch Einatmen eine Luftblase zurückbekommen. Diese Integration des Spielers erweitert das Spiel um eine Vielzahl an Strategien, beispielsweise kann der Spieler versuchen, möglichst lange die Luft anzuhalten, um Luftblasen zu sparen. Eine andere Strategie wäre etwa ein häufiges Auftauchen.

Um die Steuerung intuitiver zu gestalten, wird ein Pinguinkuscheltier mit einem Drucksensor versehen. Drückt der Spieler das Kuscheltier gibt der Sensor das Signal zum Auftauchen, das im Prototyp noch mit der Computertastatur ausgelöst wird. Diese Art der Steuerung macht zum einen mehr Spaß als das Drücken einer Taste auf der Tastatur und passt zum anderen besser in den Kontext des Spiels. Für Kinder als Zielgruppe des Spiels bietet ein Kuscheltier eine wesentlich greifbarere Möglichkeit der Interaktion und haptischen Wahrnehmung.

Den Spieler zusätzlich zu motivieren ist das Ziel der nächsten Erweiterung. Mithilfe des Q-Learning Algorithmus soll das Spiel mit zunehmendem Spielverlauf lernen, wann der Spieler eine Luftblase gebrauchen könnte. Diese erscheinen dann nicht mehr zufällig im Spiel, sondern werden zu sinnvollen Zeitpunkten generiert. Auf diese Weise kann das Spiel zum einen erschwert und zum anderen dem Spieler in einer ausweglosen Situation weitergeholfen werden.

# Technische Umsetzung

## Grundlegende Implementierung als Browsergame

Das Spiel wird plattformunabhängig als Browsergame umgesetzt. Dazu werden die für die Web-Programmierung üblichen Technologien HTML, CSS und JavaScript verwendet. Das Framework create.js stellt grundlegende Spielkomponenten zur Verfügung und wird unter anderem für das Laden und Animieren von Sprites verwendet.

Der Prototyp des Spiels wird zunächst über die Computertastatur gesteuert. Zur Integration des Atemsensors und eventuell eines alternativen Gegenstands zur Spielsteuerung wird ein Arduino Uno Mikrocontroller verwendet. Dieser kommuniziert wiederum mit einem Node.js Webserver mit der Spieloberfläche.

## Brustgurt mit Atemsensor

Der Atemsensor ist ein vom Lehrstuhl zur Verfügung gestellter Stretch-Sensor und wird auf einen elastischen Brustgurt befestigt. Atmet der Spieler ein, vergrößert sich sein Brustumfang und der Brustgurt wird fester gespannt. Der Sensor erkennt und misst den Kräfteunterschied. Der Spieler trägt während dem Spiel den Brustgurt, der an den

## Alltagsobjekt zur Steuerung des Pinguins

## Reinforcement Learning

# Projektplan

## Aufgabenverteilung

Gernot Brunner Konstruktion und Integration des Atemsensors, Plüschtier-Controller

Stefan Cimander Projektplan, Dokumentation, Präsentationsfolien

Martin Hartmond Implementierung, Spiellogik

Thomas Weber Implementierung, graphische Oberfläche, Spiellogik

## Projektmeilensteine

23. Juni 2016 Demonstration eines ersten Prototyps

27. Juni 2016 Integration des Atemsensors und Anpassung der Spiellogik

01. Juli 2016 Intelligentes Platzieren von Luftblasen

06. Juli 2016 Drücken eines Plüschtiers als alternative Steuerung

11. Juli 2016 Abschlusspräsentation und Abgabe der Dokumentation und Projekttagebücher

## Risiken

Zeitrahmen Die genannten Meilensteine erlauben eine einfache Fortschrittskontrolle. Auf diese Weise sollen Zeitprobleme vermieden und das Projekt zum rechtzeitigen Abschluss gebracht werden.

Atmungserkennung Es existiert ein nicht unwesentliches Risiko, dass die Erkennung der Atmung des Spielers technisch nicht so umgesetzt werden kann, wie es für das Spiel notwendig wäre. Aus diesem Grund wird neben der Spieloberfläche als zweiter wichtiger Schwerpunkt zu Beginn des Projekts mit der Konstruktion des Brustgurts und dem Atemsensor begonnen. Dem Risiko wird somit frühzeitig begegnet, mögliche Probleme werden durch regelmäßiges manuelles Testen sofort erkannt.

Projektteam Der Ausfall eines oder mehrerer Teammitglieder, beispielsweise aufgrund von Krankheit, gefährdet prinzipiell den Projekterfolg. Erfahrungen aus vergangenen Projektarbeiten und Präsentationen während dem Semester haben allerdings gezeigt, dass unser Team flexibel auf solche Ausfälle reagieren konnte, weshalb dieses Risiko als nicht zu hoch bewertet wird.