|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 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. Ziel des Spiels ist es, einen weiteren, vom Computer gesteuerten Pinguin einzuholen. Die Hindernisse bewegen sich dabei von der rechten Seite in den sichtbaren Bereich hinein, bei einem Zusammenstoß eines Pinguins mit einem Hindernis wird er für einen Moment lang aufgehalten. So ist es möglich, durch geschicktes Ausweichen den Abstand zum anderen Pinguin zu verringern, falls dieser einen Eisblock berührt und kurzzeitig gestoppt wird. Gelingt es hingegen dem verfolgten Pinguin den sichtbaren Bereich des Bildschirms zu verlassen, so hat der Spieler verloren.

Als zusätzliches Spielelement stehen dem Spieler nur eine begrenzte Anzahl an Luftblasen zur Verfügung, die mit der Zeit vom Pinguin verbraucht werden. 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 verloren.

## Erweiterungen

Um dem Spieler eine interessantere Herausforderung zu geben, die ihn auch körperlich etwas miteinbezieht, 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 der Atmung 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 der verfolgte Pinguin mit zunehmendem Spielverlauf lernen, wie er es am besten vermeidet, mit Eisblöcken zusammenzustoßen. Dies erhöht nach ausreichendem Lernen den Schwierigkeitsgrad des Spiels und fordert somit den Spieler zunehmend. Das Spiel bietet dem Spieler somit neue Herausforderungen und wird nicht so schnell zu einfach und damit langweilig.

# 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 misst und signalisiert den Kräfteunterschied. Der Spieler trägt während dem Spiel den Brustgurt, der an den Arduino angeschlossen ist. Mithilfe des Node.js Webservers werden das entsprechend erkannte Ein- und Ausatmen an den Game Client im Browser gesendet. Auf diese Weise ist es möglich, dem Spieler eine Luftblase abzuziehen, falls er unter Wasser einatmet und ihm eine Luftblase zurückzugeben, falls er über der Wasseroberfläche einatmet.

## Alltagsobjekt zur Steuerung des Pinguins

Zur Steuerung der Spielfigur soll ein Kuscheltier, passend in Form eines Pinguins, verwendet werden. Dazu wird in das Kuscheltier ein Flex-Sensor eingenäht, der beim Zusammendrücken seinen Widerstand verändert. Dieser ist ebenfalls mit dem Arduino Mikrocontroller verbunden.

## Reinforcement Learning

Um den verfolgten Pinguin mit fortgeschrittenem Spielverlauf ein intelligenteres Ausweichverhalten beizubringen, wird der in der Vorlesung behandelte Q-Learning Algorithmus verwendet. Dabei werden dem Algorithmus die aktuelle Position des Pinguins auf einer diskreten Skala und festgelegte Hindernis-Abtastpunkte vor dem Pinguin als Spielstatus übermittelt. Belohnt wird der Algorithmus bei erfolgreichem Ausweichen, bestraft bei einem Zusammenstoß mit einem Hindernis.

# Projektplan

## Aufgabenverteilung

Gernot Brunner Konstruktion und Integration des Atemsensors, Plüschtier-Controller

Stefan Cimander Projektplan, Dokumentation, Videodreh und -schnitt

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

06. Juli 2016 Drücken eines Plüschtiers als alternative Steuerung

11. Juli 2016 Implementierung des Q-Learning-Algorithmus

14. 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.