Heartbeat Horror
Mitchell Neal and Jacob Brown
Team Polsky

maneal@uchicago.edu
jacobpeterbrown@uchicago.edu
Background

**Problem**
Games are lacking in their ability to detect the quality of a player’s game experience, and to respond to it.

**Idea**
A game that senses the player’s involuntary responses during the game, and is able to change the gameplay experience accordingly in real time.

---

**Our Idea: Your Heartbeat is Your Health Bar**

- The game senses the player’s heart rate, and the player loses if their HR gets too high.
- The game can change based on what it detects scares the player most.
Basic Description

- In the game, you are taken through a series of different scenes. In this live demo, we will only be showing the first scene.
- This game represents a proof-of-concept of the technology, so movement through the scene is scripted; the only motion you control is head motion.
- On the menu screen, we estimate a baseline heart rate. In the game, if your heart rate rises 40% above this baseline heart rate, you lose the game.
- We have 3 pieces of software involved in monitoring your heart rate. 1) an app on the watch that reads heartbeat data. 2) a background app on the phone that takes the data from the watch and broadcasts it to the Unity game. 3) a unity plugin that listens for HR data.
Live Demo!
Volunteers?
Background Research: Hierarchies of Fears

Domains of Fear

The Chapman University Survey of American Fears addressed 11 major domains:

- **Immigration/Demographic Changes**: Illegal immigration, whites no longer being majority in US
- **Environment**: Global warming, overpopulation, pollution
- **Relationships**: Significant other cheating, others talking about you behind your back
- **Technology**: Artificial intelligence, robots, cyber-terrorism
- **Natural Disasters**: Earthquakes, droughts, floods, hurricanes
- **Illness and Death**: High medical bills, becoming ill, dying
- **Economic**: Not having enough money for the future, becoming unemployed
- **Government**: Government corruption, Obamacare, drones, gun control
- **Man-Made Disasters**: Bio-warfare, terrorism, nuclear attacks
- **Personal Fears**: Tight spaces, public speaking, clowns, vaccines
- **Crime**: Murder, rape, theft, burglary, fraud, identity theft

The Chapman University Survey of American Fears 2016
### Top 10 Fears

<table>
<thead>
<tr>
<th>Fear</th>
<th>Fear Domain</th>
<th>% Afraid or Very Afraid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corrupt government officials</td>
<td>Government</td>
<td>60.6</td>
</tr>
<tr>
<td>Terrorist Attack</td>
<td>Manmade Disasters</td>
<td>41</td>
</tr>
<tr>
<td>Not having enough money for the future</td>
<td>Economic</td>
<td>39.9</td>
</tr>
<tr>
<td>Terrorism</td>
<td>Crime</td>
<td>38.5</td>
</tr>
<tr>
<td>Government restrictions on firearms and ammunition</td>
<td>Government</td>
<td>38.5</td>
</tr>
<tr>
<td>People I love dying</td>
<td>Illness and Death</td>
<td>38.1</td>
</tr>
<tr>
<td>Economic/financial collapse</td>
<td>Economic</td>
<td>37.5</td>
</tr>
<tr>
<td>Identity theft</td>
<td>Crime</td>
<td>37.1</td>
</tr>
<tr>
<td>People I love becoming seriously ill</td>
<td>Illness and Death</td>
<td>35.9</td>
</tr>
<tr>
<td>The Affordable Health Care Act/Obamacare</td>
<td>Government</td>
<td>35.5</td>
</tr>
</tbody>
</table>

### Top 10 actionable fears

<table>
<thead>
<tr>
<th>Fear</th>
<th>% of Americans who say they fear...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Speaking</td>
<td>20.6</td>
</tr>
<tr>
<td>Heights</td>
<td>20.0</td>
</tr>
<tr>
<td>Bugs, Snakes, Other Animals</td>
<td>20.0</td>
</tr>
<tr>
<td>Drowning</td>
<td>19.8</td>
</tr>
<tr>
<td>Blood/Needles</td>
<td>18.7</td>
</tr>
<tr>
<td>Claustrophobia</td>
<td>18.1</td>
</tr>
<tr>
<td>Flying</td>
<td>18.1</td>
</tr>
<tr>
<td>Strangers</td>
<td>16.9</td>
</tr>
<tr>
<td>Zombies</td>
<td>15.0</td>
</tr>
<tr>
<td>Darkness</td>
<td>13.6</td>
</tr>
<tr>
<td>Clowns</td>
<td>11.5</td>
</tr>
<tr>
<td>Ghosts</td>
<td>9.9</td>
</tr>
</tbody>
</table>

Notice: Most of these are not actionable in a simple video game!
Scenes Not Shown in the Live Demo

Underwater Sea Monsters Scene:
Plays on fear of water/drowning, vague shapes drift in and out of peripheral vision
Scenes Not Shown in the Live Demo

Hospital Scene:
Fear of blood/hospitals, darkness. Long hallway directs player’s vision forward, leaving the periphery for surprises.
Scenes Not Shown in the Live Demo

Urban Rooftop Scene:
Rooftop setting plays on fear of heights
Fear Manager

- Object in the scene that changes gameplay based on monitored response
  - Responses considered: your heartbeat, the speed of your head movement, and what objects your eyesight lingers on.
- In general, it uses these responses to determine whether you are scared of individual objects in the game world.
  - Once deciding you are scared of an object, it will then increase or decrease the frequency of this object in the game world, depending upon whether the code is set to “easy” or “hard” mode.
- The algorithm that determines if you are scared of any specific object is based upon the behavior of a neuron firing an impulse.
Fear Manager Code Snippets

```csharp
if (Application.loadedLevelName == "forest")
{
    Spherecast(forward, hit);
}

world Spherecast(Vector3 forward, RaycastHit hit)
{
    CharacterController charCtrl = GetComponent<CharacterController>();

    if (Physics.Spherecast(forward, charCtrl.height / 2, transform, out hit, 1))
    {
        if (hit.collider.gameObject.CompareTag("littleSpider"))
        {
            GameObject.Find("InputManager").GetComponent<InputManager>().checkForClick();
            GameObject.Find("Heartbeat Manager").GetComponent<FearManager>().scaredOfLittleSpiders += .5f;
            hit.gameObject.tag = hit.collider.gameObject;
        }
        if (hit.collider.gameObject.CompareTag("bigSpider"))
        {
            GameObject.Find("InputManager").GetComponent<InputManager>().checkForClick();
            GameObject.Find("Heartbeat Manager").GetComponent<FearManager>().scaredOfBigSpiders += .5f;
            hit.gameObject.tag = hit.collider.gameObject;
        }
        if (hit.collider.gameObject.CompareTag("flatlegs"))
        {
            GameObject.Find("InputManager").GetComponent<InputManager>().checkForClick();
            GameObject.Find("Heartbeat Manager").GetComponent<FearManager>().scaredOfFlatlegs += .5f;
            hit.gameObject.tag = hit.collider.gameObject;
        }
    }
}

RaycastHit hit = new RaycastHit();
comparison2 = comparison;
comparison = forward;
forward = GameObject.Find("Reticule").GetComponent<Rigidbody>().transform.TransformDirection(Vector3.forward);

if (Math.Abs(forward.x - comparison2.x) > 5 || Math.Abs(forward.y - comparison2.y) > 5)
{
    GameObject.Find("Heartbeat Manager").GetComponent<FearManager>().scaredOfSpiders += .5f;
}
```

if (heartBeat > restingHeartRate * 1.2)
{
    RenderSettings.fogColor = Color.red;
    RenderSettings.fogDensity = .8f;
    AudioSource audio = this.GetComponent<AudioSource>();
    audio.Play();
}

if (Application.loadedLevelName == "forest")
{
    // getting players x coordinate location
    float x = GameObject.Find("Guy with flashlight").transform.position.x;

    // print text that they are scared of spiders
    scaredOfSpiders += 1;

    if (x < 58)
    {
        scaredOfLittleSpiders += .5f;
    }
    if (x > 58 && x < 68)
    {
        scaredOfBigSpidersScares += .5f;
    }
    if (x > 68 && x < 74)
    {
        scaredofFogScares += .5f;
    }
    if (x > 74 && x < 97)
    {
        scaredOfFogScares += .5f;
    }
    if (x > 98 && x < 120)
    {
        scaredOfFogScares += .5f;
    }
    if (x > 120)
    {
        scaredofFogScares += .5f;
    }
```
Android Wear App/Unity Plugin Code Snippets

**Wear App Pseudocode**

```java
    private GoogleApiClient googleClient;
    private SensorManager mSensorManager;
    private Sensor mSensor;

    protected void onCreate(Bundle savedInstanceState) {
        Initialize(mSensorManager);
        Initialize(googleClient);
    }

    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_HEART_RATE) {
            putDataMapRequest.putDataMapRequest = putDataMapRequest.create();
            putDataMapRequest.getDataMap().putString("message", event.hrdata());
            putDataMapRequest.setDataRequest().setUrgent();
            PendingResult<DataApi.DataItemResult> pendingResult = Wearable.DataApi.putDataItem(googleClient, putDataRequest);
        }
    }
}
```

**Mobile Listener/Forwarder Pseudocode**

```java
public class GoodIntentions extends Service implements GoogleApi.Client.ConnectionCallbacks, DataApi.DataListener, GoogleApi.Client.OnConnectionFailedListener{
    private GoogleApiClient googleClient;

    public void onCreate() {
        // Pseudocode:
        Initialize(googleClient);
        googleClient.connect();
    }

    @Override
    public void onDataChanged(DataEventBuffer dataEvents) {
        // Get data from data event:
        DataItem item = event.getDataItem();
        DataMap dataMapItem = DataMapItem.fromDataItem(item);
        String hr = dataMapItem.getDataMap().getString("message");
        // Create intent to broadcast HR data to Unity App
        Intent intent = new Intent();
        intent.putExtra(Intent.EXTRA_TEXT, hr);
        sendBroadcast(intent);
    }
}
```

**Unity Listener Pseudocode**

```java
public class Receiver extends BroadcastReceiver {
    private static Receiver instance;
    public static String text = "";

    @Override
    public void onReceive(Context context, Intent intent) {
        String message = intent.getStringExtra(Intent.EXTRA_TEXT);
        if (message != null) {
            text = message;
        }
    }
}
```
Challenges

● To access deeper Android features such as messaging and listener services, we needed to create a Unity plugin in Android Studio

Sense HR  >  Send Data from Watch  >  Listen for Data on Phone  >  Send to Unity Plugin  >  Use in Unity

● What we wanted to do here isn’t very well documented
● We found a few startups devoted to sensing biometric data for VR games
Next Steps

- Improve our game to include more player-directed movement and control
- Adding the ability to monitor verbal data
- More play testing
- Do even more with involuntary player feedback

More Advanced Scenes We Have Planned Are...
Future Additions to the Game

- Level where zombies can ‘smell fear’ and only attack if you’re afraid
- Plays on fear from anticipated danger
Future Additions to the Game

- Crossing a narrow bridge
- Camera shakes if your heart rate is too high, simulating impairment from fear
Thanks!

Questions?