GNG2101 Final Report

Kinect the Rhythm

Submitted by
Team B3D, Kinect Therapy

Anthony Lai	300012706
Patrick Nguyen	300001909
Andrew Pratheepan	300019807
Marino Vethanayagam	300014892

December 21st, 2018
University of Ottawa

Abstract

Kinect the Rhythm is a video game utilizing the technology of the Xbox Kinect. Structured as a rhythm game, taking inspiration from the likes of Osu and Guitar Hero. The difference with Kinect the Rhythm is that it serves a greater purpose than simply entertainment, but also as a rehabilitation tool for Saint-Vincent Hospital in Ottawa. In this report we will be explaining the long and lengthy process which was followed when developing this game. Furthermore discussing the product and its design process in depth, as well as the inspirations, constraints and issues we encountered throughout development.

The development of this game was separated into several core development components or features. First of which was the actual Kinect integration where we had to configure the Kinect to be workable in Unity 3D. Secondly, the core game was to be built, this is the physical game mechanics and feel. Concurrently as the core game was being built, the user interfaces including the profiles functionality and how-to-play screen were also coded. In the early stages we conferred with the client a lot in order to get valuable feedback to better improve our game for the hospital's specific needs.

Table of Contents

List of Figures

List of Acronyms

Acronym	Definition
KTR	Kinect the Rhythm. The video game developed for St. Vincent's Hospital.
SVH	Saint-Vincent Hospital
SDK	Software Development Kit
API	Application Program Interface
UI	User Interface
UX	User Experience

Introduction

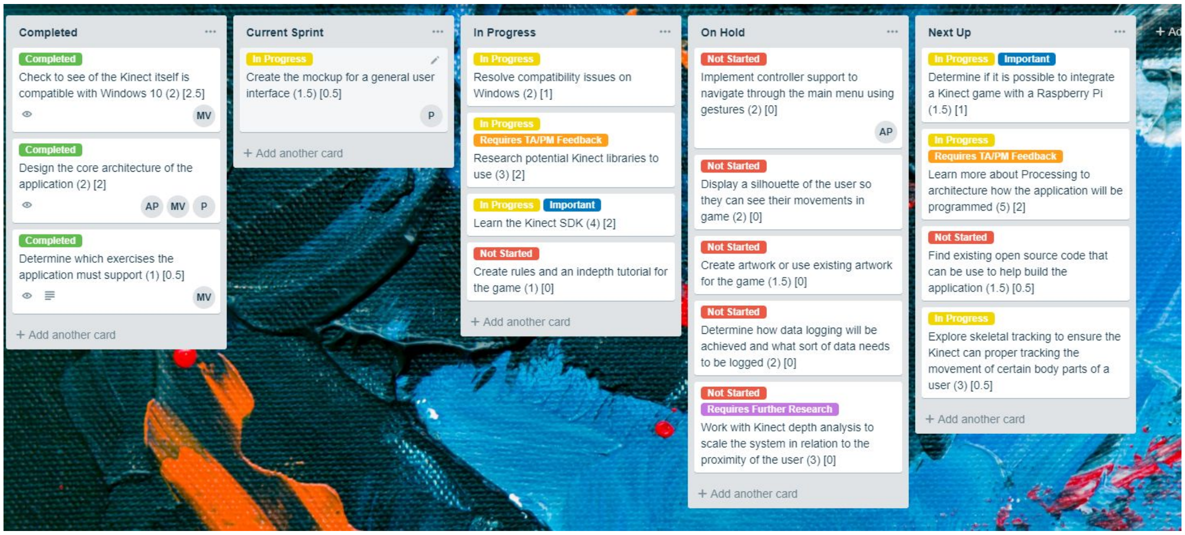
Kinect the Rhythm was born of the need to re-simulate physical therapy at Saint-Vincent Hospital in Ottawa. SVH is the lone long term care hospital in Ottawa and it has a specialized program which aims to help maintain and improve patients functions in their care. They requested for us to build a video game utilizing an XBox 360 Kinect v.1, which they provided, that would help with physical therapy at the hospital. The hospital found that traditional physical therapy was not as effective as it once was, in certain cases. They felt this game could help supplement and assist on to their existing rehabilitation programs by giving it a brand new tool.

KTR was developed primarily as a new platform for physical therapy conducted at SVH. The goal was to include many different types of exercises varying in intensity and difficulty, mainly focused on muscles in the upper body. These included biceps, triceps and shoulders, as they were the most compatible with the Kinect sensors' capabilities.

The success of this project can be measured by the completion of the primary goals which was to develop an intuitive, simplistic and user friendly game which would help increase participation and physical activity for the therapy patients at SVH.

Engineering Design Process

For our design process we decided to use the Agile methodology. Since our project is a software application, the Agile methodology allowed us to easily distribute tasks amongst group members. This is due to the fact that Agile is an iterative design process which means that every member can to work on small components of the product at a time and then we could merge pieces of working components together. To make this methodology successful, we used a free online SCRUM board called Trello. The SCRUM board allows everyone to see what software features still need to be implemented and the estimated time it will take to complete it. This gives the other members a sense of the progress the other member. Below is a screenshot of the SCRUM board we used to divide tasks amongst members:



Completed

- Completed
  Check to see of the Kinect itself is compatible with Windows 10 (2) [2.5]
- Completed
  Design the core architecture of the application (2) [2]
- Completed
  Determine which exercises the application must support (1) [0.5]

Current Sprint

- In Progress
  Create the mockup for a general user interface (1.5) [0.5]

In Progress

- In Progress
  Resolve compatibility issues on Windows (2) [1]
- In Progress
  Requires TA/PM Feedback
  Research potential Kinect libraries to use (3) [2]
- In Progress Important
  Learn the Kinect SDK (4) [2]
- Not Started
  Create rules and an indepth tutorial for the game (1) [0]

On Hold

- Not Started
  Implement controller support to navigate through the main menu using gestures (2) [0]
- Not Started
  Display a silhouette of the user so they can see their movements in game (2) [0]
- Not Started
  Create artwork or use existing artwork for the game (1.5) [0]
- Not Started
  Determine how data logging will be achieved and what sort of data needs to be logged (2) [0]
- Not Started
  Requires Further Research
  Work with Kinect depth analysis to scale the system in relation to the proximity of the user (3) [0]

Next Up

- In Progress Important
  Determine if it is possible to integrate a Kinect game with a Raspberry Pi (1.5) [1]
- In Progress
  Requires TA/PM Feedback
  Learn more about Processing to architecture how the application will be programmed (5) [2]
- Not Started
  Find existing open source code that can be use to help build the application (1.5) [0.5]
- In Progress
  Explore skeletal tracking to ensure the Kinect can proper tracking the movement of certain body parts of a user (3) [0.5]

Figure 1: Scrum Board Example

Need Identification and Product Specification Process

The rehabilitation program uses the kinect sensors to make games/exercises for a general range of patients at St. Vincent Hospital. The generalized target audience tend to have lower motor control and most only have control of their upper body. Due to this, the kinect program **must** have an intuitive user interface that is easy accessible for the patients (it is expected to be designed for a lower range of motion). The *games* should assist in rehabilitation by incorporating/shadowing the exercises listed by the rehab specialist. In the program, there should be minimum movement in terms of patient location (as a large group of patients are wheelchair bound). The program must have some form of progression (for the patients that are able to proceed with harder tasks and wider ranges of motion), this can be either as modes of difficulty (easy, medium, hard, etc.) or by means of data collection accessing the specific user based on their effectiveness (optimal solution).

Metrics

Some of the metrics to be considered are measurable in the real world while others more technological based metrics:.

A. Depth Scaling (m) : Finding user's relative location
B. Point space relative velocities (m/s) : Keeps track of user movement
C. Accuracy of tracking (%)
D. Processing power required for program (GHz)

Conceptual Designs

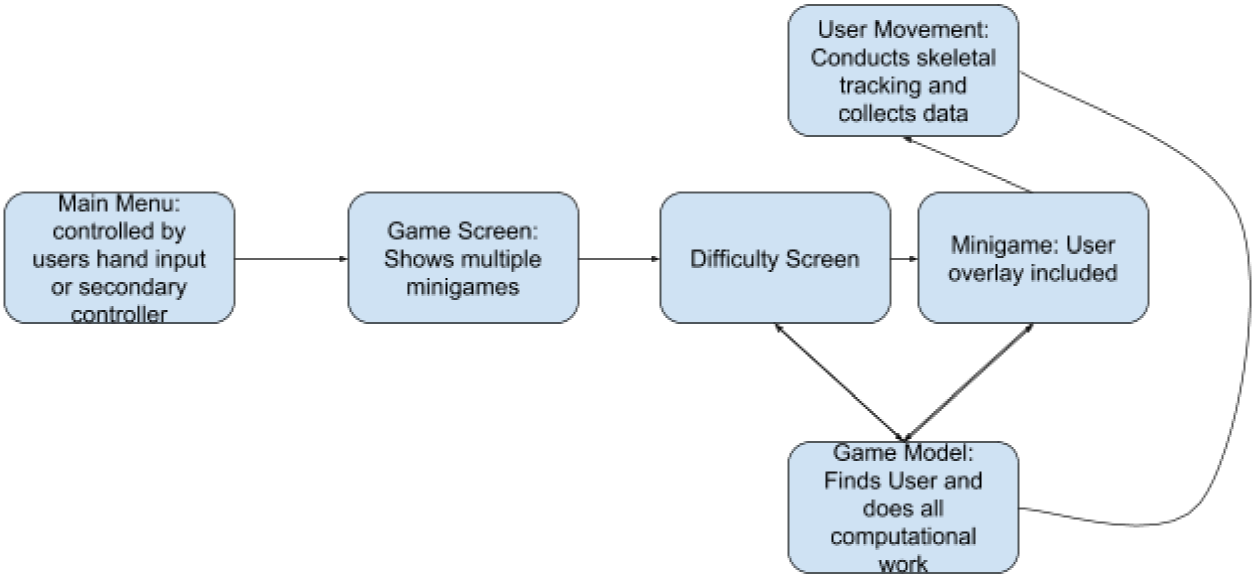

User Movement: Conducts skeletal tracking and collects data
Main Menu: controlled by users hand input or secondary controller
Game Screen: Shows multiple minigames
Difficulty Screen
Minigame: User overlay included
Game Model: Finds User and does all computational work

Figure 2: Conceptual Design of Kinect the Rhythm

Project Planning and Feasibility Study

1. Tasks
    a. Finalize and resolve any remaining compatibility issues on the Windows platform regarding drivers and OS compatibility - Marino
    b. Determine the feasibility of porting the finalized application to a Raspberry Pi through conducting more research - Anthony
    c. Developing a general and self evident user interface to navigating through application/game - Patrick
    d. Implement controller support for the main menu and context menus for users with limited mobility - Andrew
    e. Develop the core functionality of the therapy game(s) - Team
    f. Develop subsequent games with threshold and action settings that can be set and controlled by therapist - Team
    g. Develop data logging and tracking so the hospital can monitor the progression of the patients - Team
2. Milestones

   a. Running Kinect System (Tracking/detection, input/output stream)
   b. Program implementation (basic functional programing)
   c. UI integration and user controller implementation
   d. Implementation of core game
   e. Organized collection of minigames
   f. Progression System
3. Scrum Board
   a. Scrum Master - Andrew
   B. Used to monitor current, past and future tasks and visualize project progress
4. BOM
   a. Kinect System (included) - Free
   b. Raspberry Pi A/B $25-35
   c. Range Modifying lens - Free
   d. Kinect SDK - Free
   e. Kinect Software (Processing) - Free
5. Costs
   a. The only cost associated with this project is the cost of purchasing a Raspberry Pi. The cost is $25 for the model A and $35 for the model B respectively.

   However, any additional modules that may be required will result in an increased net cost.
6. Risks
   a. The client has asked for a low tech and low cost solution. Therefore, we will be attempting to run our therapy game(s) on a Raspberry Pi. However we are unsure if the necessary drivers are available and whether the hardware is sufficient enough to run our game until we can test our application using a Raspberry PI.
   b. Adding too many features may result in us not being able to complete the project in time. Therefore, we will focus on getting the core features in before we start thinking of minor details.

Prototyping, Testing and Customer Validation

Initial Prototype:

- Xbox Kinect system that connects to a windows system with a real time stream of imaging and basic tracking
- Able to display skeletal tracking information in a variety of scenarios
- The ability to show multiple streams of input onto the screen such as color image, depth and 3D vision

Comprehensive Prototype:

- Kinect working with Unity
- Functional rehabilitation game
- Data Logging and User Base
- Safeguards to ensure programs functions according to design
- Intuitive UI and UX

Measuring success of prototype (testing):

The prototype's success will be measured on the detail on the schematics we designed and on the technical accuracy of our program. For a kinect application, the technical accuracy is a measure of how well our program uses the kinect to track the movements of a user and the quality of the model of a user displayed on a screen. Additionally, data logging is another important requirement requested by the client. Therefore, testing the practicality and accuracy of our game's data logging is another good metric of our progress. Overall, building a prototype will give us a physical representation of our ideas for the application allowing the client to give input on the direction we are heading in and if we are on the right track or need to reevaluate our design.

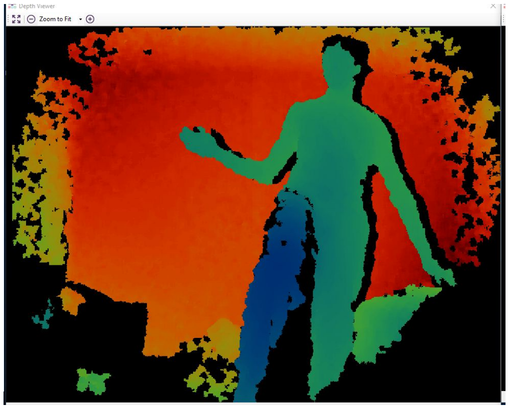


Figure 3: Skeletal Tracking Testing

Customer Validation:

Through the three client meetings with Bocar (the Technologist at SVH), constructive criticism was given. With each iteration of prototypes (shown in the 2nd and 3rd meetings), different standards of testing was instituted to validate progress (based on time constraints and what is reasonable to expect with set times.

For the first client meeting, initial testing involved validating skeletal track and overall making sure the technologies work together (windows machine, kinect sdk and kinect). It was deemed a success as it tracked the movements of a user and the was able to display both the user and the overlay of the skeleton depicting accurate movement detection.

The second deliverable involved a focus prototype showing the kinect interacting with the kinect. With the time constraints, it was unreasonable to expect a fully functional game so it was more so focused on the specifics of the data logging and the possibilities of the project as a whole.

The comprehensive prototype (presented product) was also deemed successful as it brought the whole project together which includes data logging, user database, and a rehabilitation game. Adding on, it was wrapped in a fairly clean UI and intuitive UX. It met the client’s needs.

Business Canvas Model

<table>
<tr>
<td colspan="2" rowspan="2"><b>Key Partners</b>
- St. Vincent Hospital
- University of Ottawa</td>
<td colspan="2"><b>Key Activities</b>
Kinect Development
Database creation and management</td>
<td colspan="2" rowspan="2"><b>Value Proposition</b>
- Better rehabilitation therapy
- Low cost
- Adaptable
- Captivating
- Customized solution
- User data collection
- More sophisticated progression tracking
- More iterative compared to in person (as it works purely with data)
- More nuanced data output for medical professional to view</td>
<td colspan="2"><b>Customer Relationships</b>
- Patients directly using software
-Therapists / technologists incorporating the software into their patient’s routine / rehabilitation</td>
<td colspan="2" rowspan="2"><b>Customer Segments</b>
- Hospitals
- Rehabilitation Centres
- Research Centres</td>
</tr>
<tr>
<td colspan="2"><b>Key Resources</b>
- Kinect
- Unity (various packages included)
- Kinect SDK</td>
<td colspan="2"><b>Channels</b>
- Website (that offers the free software)
- Mobile app (for subscription holders to view users’ profile and data, and do simple modifications of program)
- Sales (in person consultation)</td>
</tr>
<tr>
<td colspan="5"><b>Cost Structure</b>
- Kinect System</td>
<td colspan="5"><b>Revenue Streams</b>
- Free demo/trial version available for free
- license/subscription will allow more advanced features, data collection, and user integration
- Custom solutions/software for customer’s specific needs</td>
</tr>
</table>

Economic Analysis

Variable Costs: Since our end product is a software application, there is no variable cost as software can be distributed online for free through open source code distribution platforms such as github.

Fixed costs: Rent for the building where the software will be developed, tested and maintained. In addition, the cost of software licenses such as windows OS, Unity 3D, etc.

Direct costs: Include the cost of an xbox kinect, windows pc to develop, test and maintain the application.

Indirect costs: Includes the time and labour required to create, test and maintain high quality software that is efficient, modular and maintainable.

Goods sold per year: Subscription fees for premium use of our product, fee for setting up the software (ordering the kinect for the customer and setting it up for them).

Cost of goods: Totals to the cost of the kinect, and the system that would run the program (windows machine).

Gross profit: Based on the number of subscriptions and client variations requested (as the product can be adjusted to the client’s needs).

Operating expenses: Office where developers can work, various services/software used (private repositories, databases, software licenses, etc).

Operating income: Would amount to the gross profit subtracted by the cost of goods and operating expenses.

Breaking Point: Our break even point is when we make profit off of our software, this will most likely happen with our first client as we are not getting paid to design this app. Though it is uncommon for a lot of operating income to be positive in the first year of release, it is possible for this specific case as there is a low fixed cost for the development of the product and production of our product after the first successful iteration (as costs are dramatically lowered regarding development).

Assumptions: We are assuming that we have a small team of developers working on maintaining and augmenting the quality of the product. We also assume that the consumer is paying for our software and services, monthly as our product will be subscription based.

Even though our project is unlike most other as it is purely software, there are still many costs to account for. Costs such as hosting services, software licensing, and cost of any hardware we are developing for. Of course the typically costs of labour and development space are required for the development and continued maintenance of our product will be accounted more. The difference for our software product is that a subscription model is much more easily applied, making our product profitable continuously.

Final Solution

Main Menu

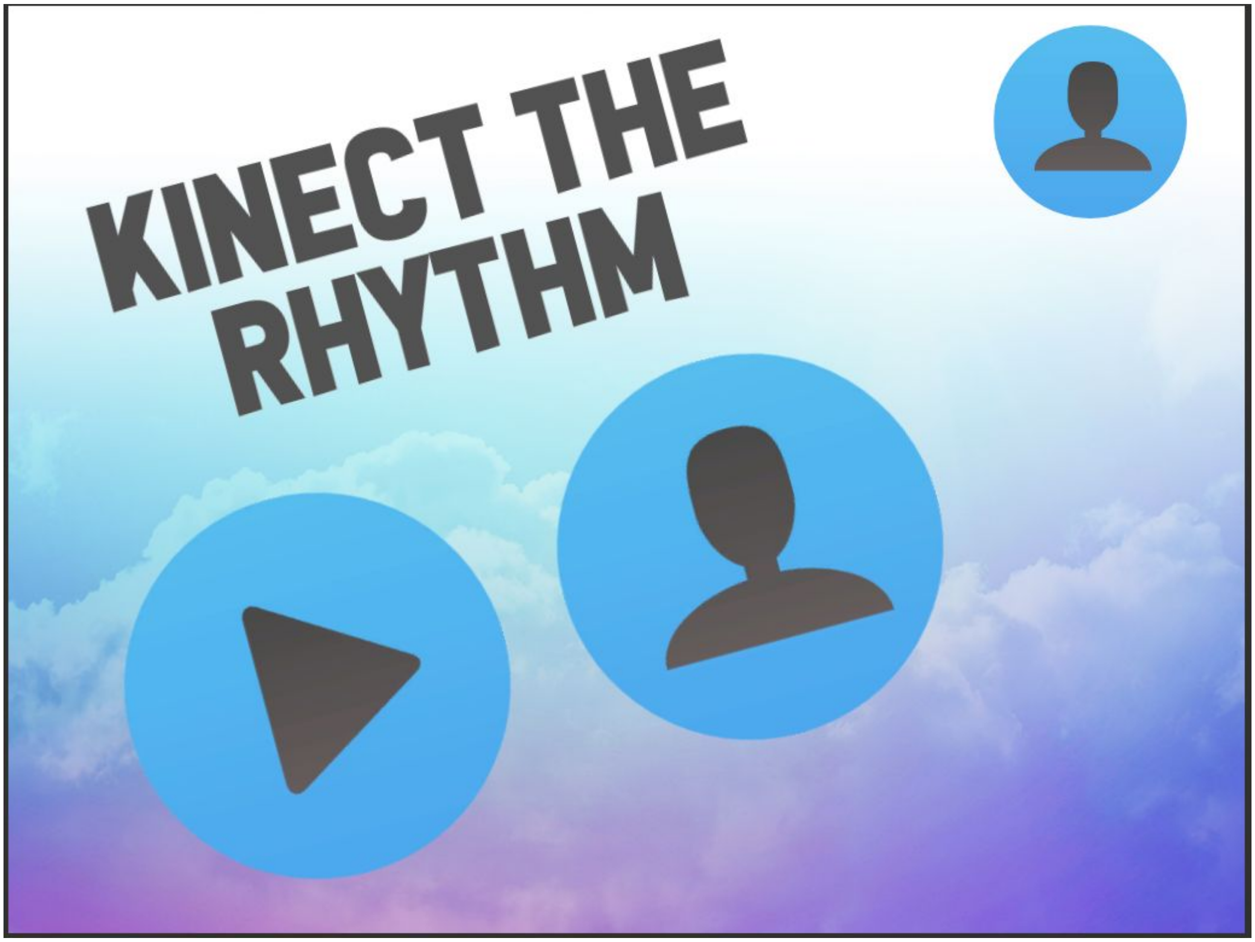


Figure 4: Main Menu

This it the starting main menu screen. The play button will begin the bicep curl game and the profile icon will allow you to access, add, edit or remove patient profiles. The image on the top right shows the profile picture of the currently selected patient.

Profile Selection

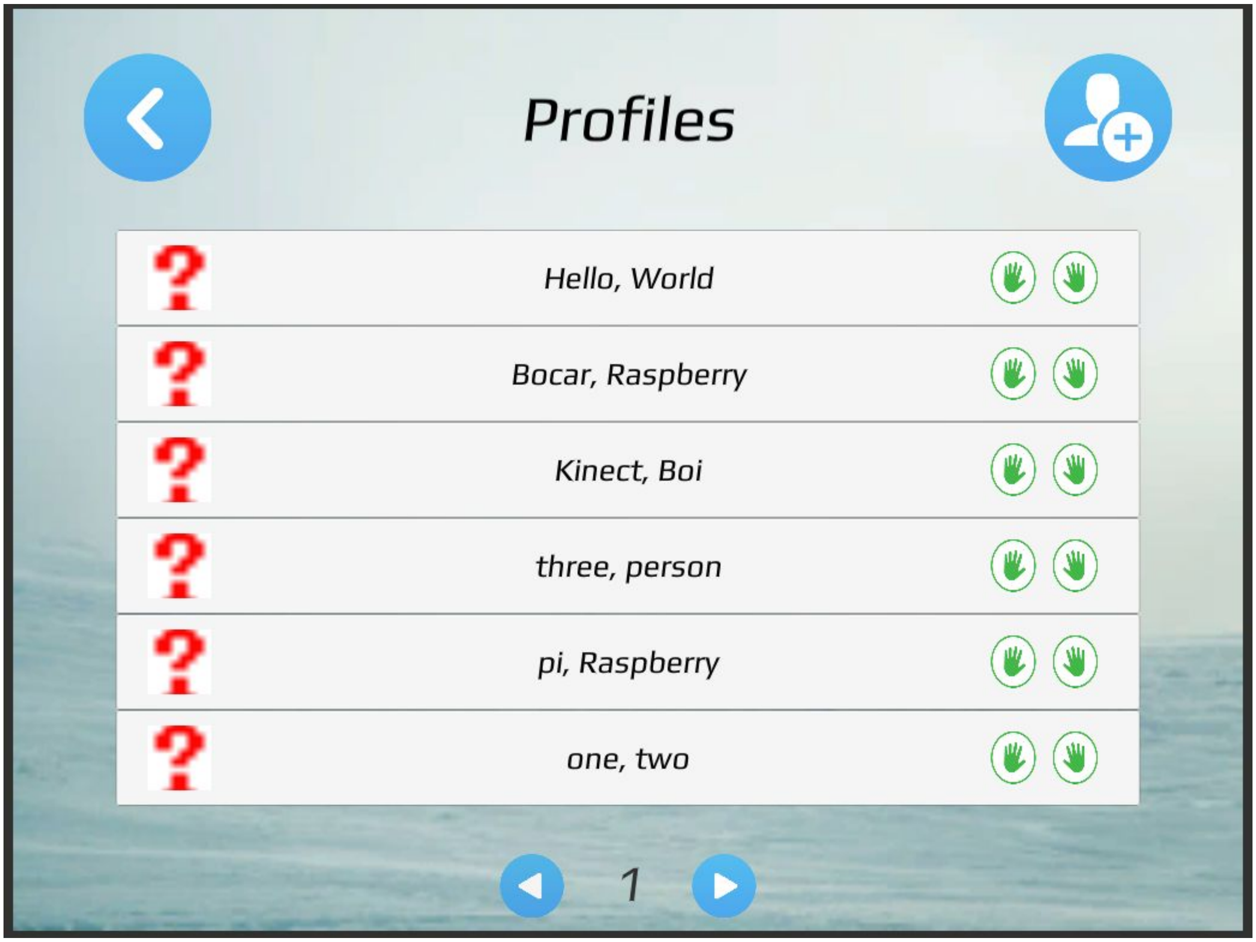


Figure 5: Profile Selection

This is the profile list screen. The screen will list all current existing profiles with the profile picture on the left and their configured exercises on the right. Since only 6 profiles can be viewed per screen there are convenient navigation buttons on the bottom which will allow you to cycle through all available profiles. There is a back button located on the top left which will take you back to the main menu and a add profile button on the top right which will allow the therapist to create profiles for new patients.

New profile

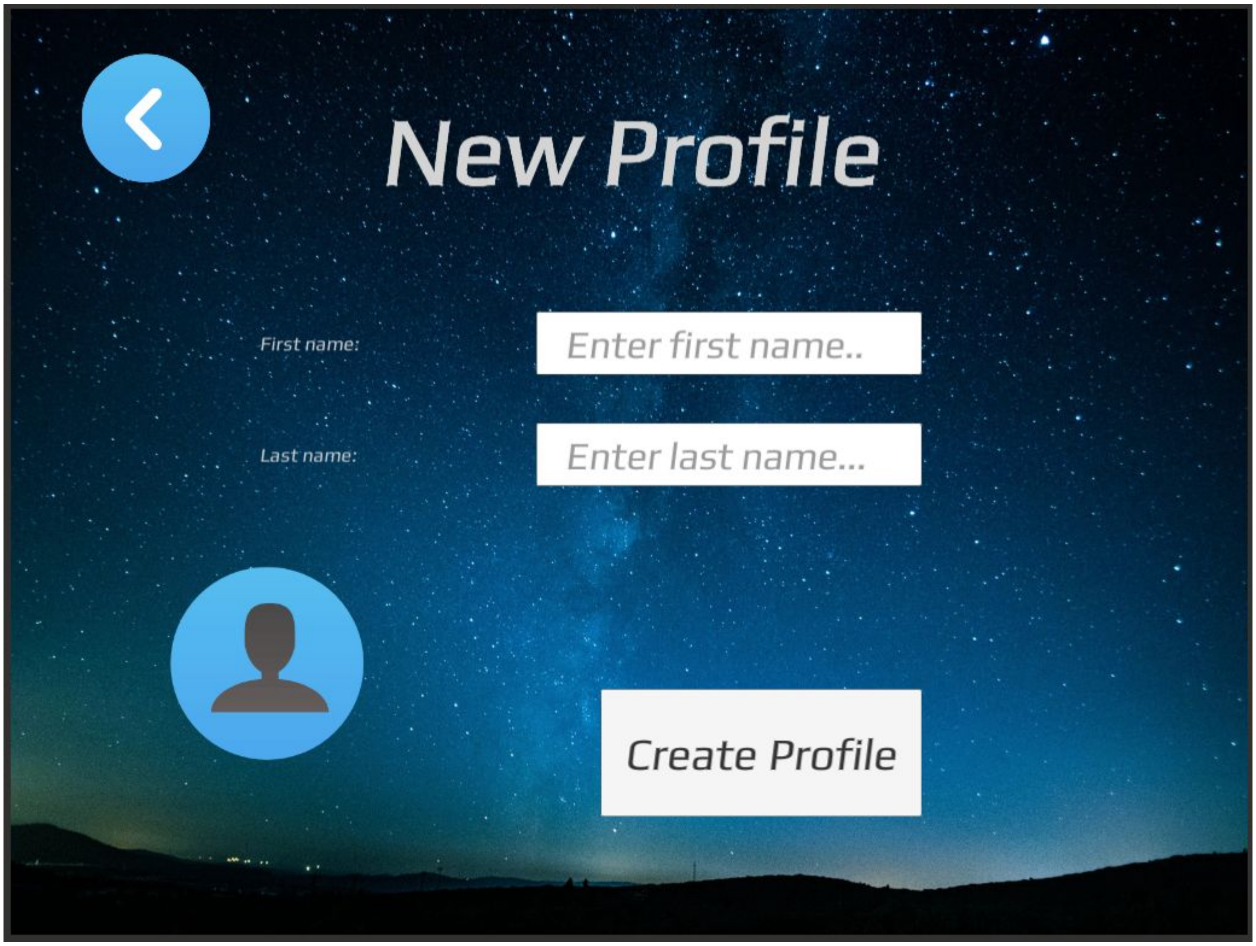


Figure 6: New Profile

The new profile screen will allow the therapist to create a profile for a new patient. There is a convenient back button on the top left which will bring the user back to the profile selection screen. In addition, there are fields to set the patients first and last name and to set a profile picture for the patient.

Main Game

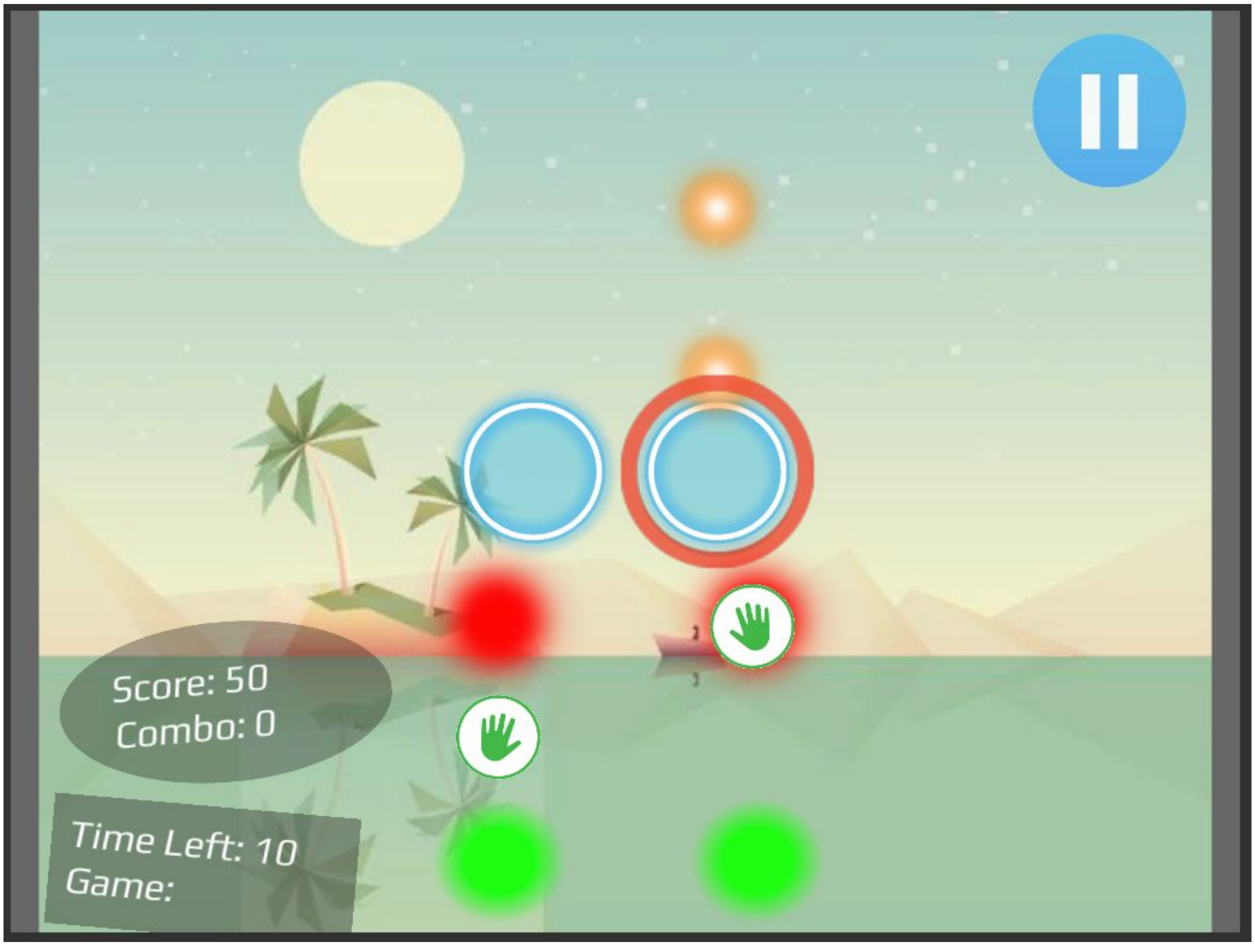
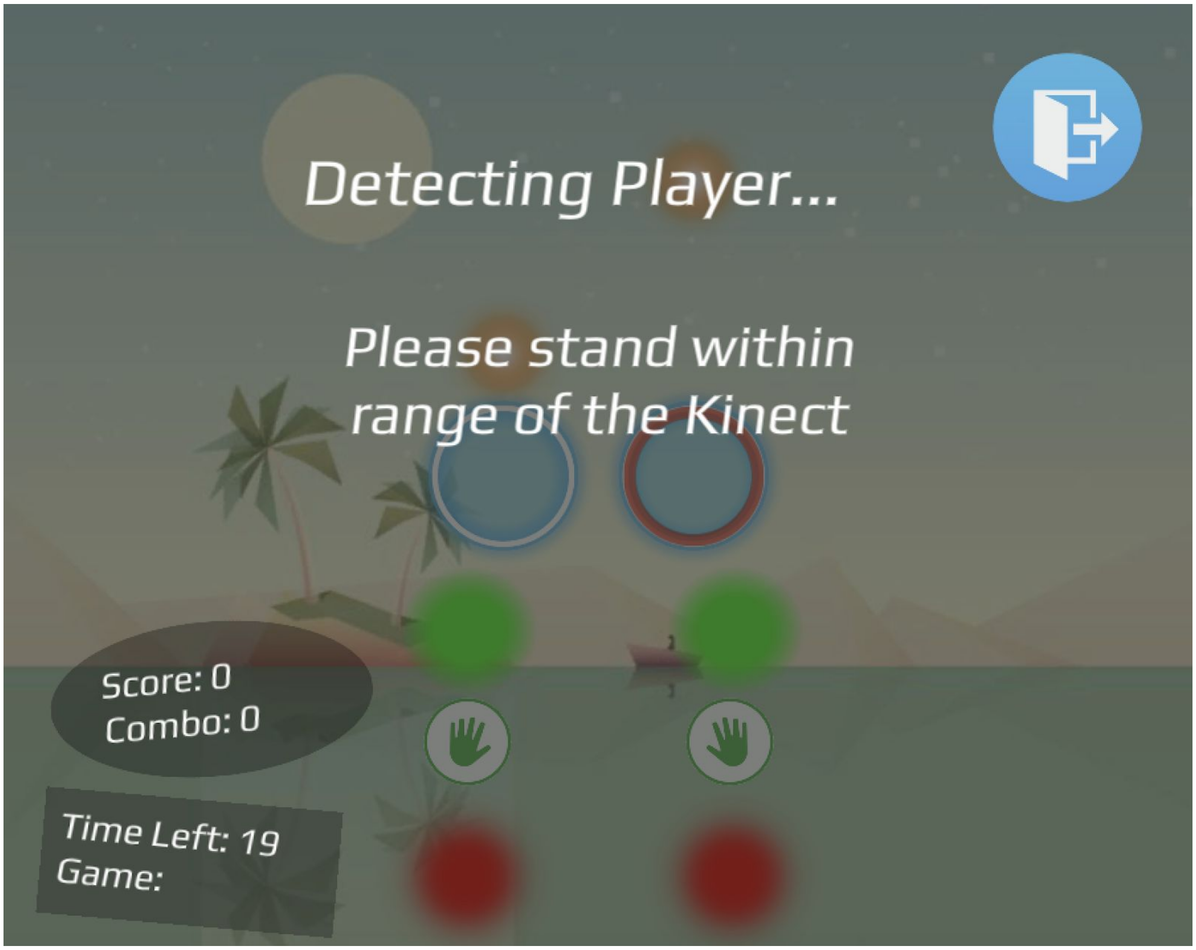


Figure 7: Main Game

Displayed above is the main game. The time left, difficulty, score and combo are all displayed on the bottom left. There is a pause button located on the top right in case the patient requires a break or the game needs to be paused for whatever reason. The 2 green and 2 red circles indicate the zones in which the player must move their hand to gain points. The played must time these movements such that the orange spheres are in the middle of the blue circle. Feedback is provided to the patient through the use of either green, yellow or red rings which indicate the accuracy of the timing.

Kinect Sensor Adjustment


Detecting Player...

Please stand within range of the Kinect

Figure 8: Kinect Sensor Adjustment
If the player is not detected by the kinect sensor, this screen is displayed until the player moves into the range of view of the sensor.

Conclusions and Recommendations for Future Work

Working on this project was a thrilling experience and we managed to complete the prototype for our product in time for presentation on design day. The difficulty of working in a group for a larger scale, larger commitment project were present during the course of the semester. Since every member of the group had different course schedules, it was fairly difficult to plan times where we could all get together to have project discussions. Although time management and planning started off fairly poor at the start of the semester, towards the end we were able to effectively manage our time and solve the problem presented in front of us. Additionally, we learned a lot about the essence of project management, product design and the technologies required to build our product. Going into this project, no one knew how to use Unity 3D, the game engine software that our application was built upon. By the end of the project, everyone knew the essential mechanisms of Unity and how to design fully functional applications for it. Overall, this experience has been memorable for all the members and we learned more about our limitations as individuals and the steps that we need to take in order to effectively communicate amongst each other so that we could successfully complete our project.

Bibliography

Ashley, James. "Unity 5 and Kinect 2 Integration." The Imaginative Universal, 27 Mar. 2015, www.imaginativeuniversal.com/blog/2015/03/27/unity-5-and-kinect-2-integration/

"Kinect 2 and Unity 3D: How to..." Sergiy Baydachnyy, 20 Nov. 2014, en.baydachnyy.com/2014/11/20/kinect-2-and-unity-3d-how-to/.

Appendices

Appendix I: User Manual

Features

- Bicep Curl Exercise
    - Accuracy tracking
    - Velocity and range of motion tracking
    - Scoring and combo system
- Profile creation and management
- Data logging and profiling
- Automatic patient progress report generation

Maintenance

The software contains no general maintenance and will work as is. Future updates, feature additions and bug fixes can be found on the github repo found at https://bit.ly/2CrX9KK

Installation Instructions

The project is found as a Unity project which contains the project files and the necessary Kinect files to allow kinect data to be used and processed in the game. The game can be ported to any of the 25 supported platforms through Unity Multiplatform. Afterwards, the game can be started by starting the compiled executable. Tested with Unity 2018+.

Development Instructions

If you wish to continue development of this project the necessary development files can be found on the github repo at https://bit.ly/2CrX9KK. In order to ensure proper kinect functionality, ensure that you are utilizing a Kinect 360 Sensor v1 and that your have the Kinect v1.8 SDK installed found here https://www.microsoft.com/en-ca/download/details.aspx?id=40278. If you wish to debug or test the sensor the Kinect Development Toolkit can be found here https://www.microsoft.com/en-ca/download/details.aspx?id=40276. The development toolkit also provides a variety of examples and documentation in a variety of programming languages which will aid in the development of kinect enabled applications.

FAQ & Troubleshooting

Detecting Player - Please stand within range of the kinect

This screen is shown when the kinect sensor is unable to detect a player. Please ensure that you are in the visible range of the kinect sensor by aligning yourself in the square of the bottom right of the screen and ensure that your entire body is visible.

If multiple people are visible to the kinect sensor this screen will also show to indicate that the sensor is unable to determine which person the game should track. Please ensure that only 1 person is visible to the sensor and the game will continue to work.

The sensor is not tracking my movements properly

This issue can occur for a number of different reasons. Please ensure that all the following criteria are satisfied in order to ensure the kinect sensor can track your movements properly

- Ensure that the user is standing the kinect sensors visible range of view
- Ensure that the user is directly facing the kinect sensor and is not angled
- Ensure that the environment is free from obstruction in the kinect field of view which would prevent the kinect from properly analyzing player movement
- All articles of clothing are fit and are not excessively baggy. Baggy or oversize clothes can confuse the sensor and prevent the game from properly tracking your movements
- Ensure that tops or jackets do not cover the knees. Long down coats which cover the knees or excessively cover the legs will prevent the kinect from properly tracking the body and should be removed when playing the therapy game.

The sensor is not connected to the game

If the game is unable to find a connected kinect sensor attempt to disconnect and reconnect the kinect sensor from the device while also restarting the game.

Appendix II: Design Files

Files regarding the project can be found on MakerRepo and Github

MakerRepo: https://makerepo.com/LaiAnthonyRH/kinect-therapy
Github: https://github.com/NguyenPatrick/Kinect-Therapy-Game

Appendix II: External Resources

Included below are links to any external resources mentioned within this report
Unity 3D - https://unity3d.com
Unity Multiplatform - https://unity3d.com/unity/features/multiplatform
Kinect SDK v1.8 - https://www.microsoft.com/en-ca/download/details.aspx?id=40278
Kinect Development Toolkit v1.8 -
https://www.microsoft.com/en-ca/download/details.aspx?id=40276
Kinect with MS-SDK -
https://assetstore.unity.com/packages/tools/kinect-with-ms-sdk-7747