Project Specification

Jordan Goldin
Diana Ly
Simon Kwong

5/21/2015

Project Overview

I. Overview

Our project involves the gamification of physical therapy. Our game utilizes the Microsoft® Kinect to allow patients to interact with the game world. Our goal is to lessen the obstacles associated with physical therapy by making the experience more enjoyable. This has several benefits over normal physical therapy. For instance, patients may focus less on their pain if they are immersed within the game. This approach may also encourage patients to progress with their physical therapy sessions rather than give up by instilling the motivation to proceed and the inclination to play and continue the game in them.

One of our biggest design goals is to create a game that is both enjoyable and clinically useful to as many patients as possible. This presented a significant challenge as the abilities and limitations of patients undergoing rehabilitation can vary immensely. Our solution to this challenge is to incorporate as many movements as we can into our game to cater to this diverse spectrum of patients. We also hope to incorporate specific obstacles to attend to precise movements desired from the physician if time permits. By using this design, we hope our game will effectively assist patients with their physical therapy.

The game we chose to create is a 3D puzzle platformer about a treasure hunter named Shamoov whose life-long ambition is to find the long lost treasure chest of the first pirate king. The basic gameplay involves Shamoov navigating through Dolphin Island, a tropical island known for its bananas, to acquire this treasure. His only hint in attaining his dream is a sign that tells him to collect the bananas.which are scattered throughout the island. The objective of this game is to follow one of the numerous trails of bananas and hope the chosen trail leads to this treasure.

As such, body movements of the patient functions as the input for our game. These motions are detected by the Microsoft® Kinect sensors and thus operate as the controller to move the character sprite around. Our game focuses on enabling the following motions:

- reaching in all directions including across the midline in all planes such as to the side, overhead, and downward
- marching in place

- jumping
- lifting of the arms

as the movements would be most beneficial for patients who will be playing our game. However, other simple motions can also act as input such as the movement of the arms. Our primary movement and basic requirement for the game is simple trunk rotation. All of these movements will be recorded and sent to a server to store the data which will be later used by the therapist.

II. Application

Once our game is complete, it will be immediately put to use at St. Mary's Hospital for healthcare providers to use. If it is shown that our game is an effective treatment for physical disabilities at St. Mary's Hospital, the use of this game can potentially expand to other healthcare settings.

III. Intended Users

This game is intended for stroke patients and other patients needing physical therapy. Ideally, this will help them by reducing the level of pain they experience when they move a certain part of their body. The idea is that if they focus their concentration at the game, they will take less notice of their physical pain and move the target area of their body more freely as opposed to when they feel conscious about it. It would also encourage them to continue their treatment by turning what was a repetitive and grueling experience into a rewarding one. The outcomes we expect is improved mobility and potentially improved cognitive function for the patient.

IV. Related Projects

There are similar projects that use the Kinect sensor. Some projects have been developed by teams who have worked with David Galles in the past. For example, they developed a game that is similar to the temple run game found in our mobile devices. However, the difference between that and the temple run game is that the movements are controlled by leaning. Another project they have implemented is a Simon Says game where they follow the on-screen prompt and maneuver objects as closely as possible using reaching motions.

Although we have stumbled across startups that explore the idea of virtual reality rehabilitation, we have not seen any products made readily available on the market as studies are still being conducted on their effectiveness. The closest product that relating to this is GestureTek's IREK® (Interactive Rehabilitation and Exercise System). The similarities between this and our game is its purpose and functionality. The difference is the implementation. IREK® is composed of multiple mini games and simulations that can range from playing the drums virtually to reenacting snowboarding whereas our game will be one cohesive game that contain the same overall theme and plot.

Project Team

Jordan Goldin
Email: jorgoldin@gmail.com
Cell: 805-795-3144

Jordan is studying computer science and fine arts at the University of San Francisco. He was born in Tarzana, California in 1993. He is well versed in Java, Python and web design. Jordan is very interested in game development and content creation. He is thrilled to have an opportunity to work on a project that has the potential to have such a positive impact on so many people.

Diana Ly
Email: dmly@usfca.edu
Cell: 408-728-1182

Diana Ly, a senior undergraduate student at the University of San Francisco (USF), is pursuing her studies in computer science and neuroscience. She is set to graduate in May and looks to work at a tech company based in Silicon Valley. Diana is currently a teaching assistant for an introductory level course on processing as well as a tutor for the CS Tutoring Center at USF. She is very familiar with using Java, Python, and C. Her interests include but are not limited to game development and healthcare technologies. Similar to Jordan, she appreciates having been granted the opportunity to work on a project that has the potential to further aid in patients' physical rehabilitation programs through gamification.

Simon Kwong
Email: simonkwong92@gmail.com
Cell: 408-876-8192

Simon Kwong is currently majoring in computer science and pursuing a minor in mathematics at the University of San Francisco. Born and raised in San Jose, he has a keen interest in working with developing games and software development. He is well acquainted with C, Python, and Java. He is privileged to be working with St. Mary's Medical Center on this project on creating a game that will both positively impact the patients' therapy and make the process enjoyable.

Project Sponsor

David Galles
Tel: (415) 422-5951
Email: galles@cs.usfca.edu

David Galles is an associate professor at the University of San Francisco. He received his B.S. from Stanford University and his M.S. and Ph.D. from the University of California at Los Angeles. His areas of interest include causal networks, programming languages, and artificial intelligence.

Project Requirements

I. Core Functionality

By the end of the semester, we will have developed a basic adventure and puzzle game intended to help ease the challenges in physical therapy.

As stated earlier in the *Project Overview* section, our game will integrate the following movements:

- trunk rotation
- marching
- jumping
- elevation of the arms

to navigate the character in the game world. These movements will be detected through the use of the Microsoft® Kinect. In the game menu, there will be a setting to calibrate the movements in relation to the patient's mobility.

As for the basic gameplay, the patient will be able to move their game sprite and interact with objects within the virtual world using collision detection and handling provided by our physics engine. Additionally, we will utilize a point system for the gameplay itself. The score will be determined by the number of items obtained.

II. Interaction

Users will interact with the game via the Kinect sensor. The Kinect will track the player's movements and allows manipulation of the character and objects within the world. Alternative controllers for inputting settings and logging in would require either a mouse and keyboard. Moreover, the therapist will be able to configure and calibrate settings appropriate for the patient and their motor capabilities via a menu.

III. Extra Functionality

If there is spare time after completing the core project, we would like to have the game's configurations saved by utilizing the database system containing the patients' information and generating a unique login associated with each patient using this game. A patient will be able to log in using their own username and password and have their configurations saved based on their own data. Healthcare providers would also be able to create their account, but will have access to all of the patient's data rather than just one and would be able to access their account configurations to make adjustments for the patients' physical rehabilitation program. We would also like to be able to integrate a wider array of movements such as lunging, squatting, and head movements. This would allow the therapist to choose different motions for the patient to focus on.

Additionally, we would like to improve the interactivity and user interface of the web application. This would entail generating a better visualization for displaying data in the web interface. Moreover, if time permits, we would also add more features to the game such as incorporating specific obstacles for certain types of movements, adding additional stages, having a separate theme and music for each level, and creating a setting in the configurations where the player would have the option of shifting the camera from first person perspective to third person and vice versa.

IV. Quality Attributes

The game should be able to run on all modern computers without lag or performance issues.

In regards to security, we will be using the OpenSSL for encrypted communication between the client game and the web server.

Both the game and web interface should be highly usable and intuitive.

Project Design

I. Overview

The game will focus on gently transitioning a patient in physical therapy from a boring and possibly painful recovery process to a more enjoyable and consciously painless experience. Although a runner track-like game exists, we will be developing a new game from the ground up with the runner game concept in mind. The game features the use of the Microsoft® Kinect v1 sensor, which the patient will use to input movement controls into the game and navigate through a world. The game will be implemented using C++ with the use of OGRE as the graphics engine and Bullet as the physics engine.

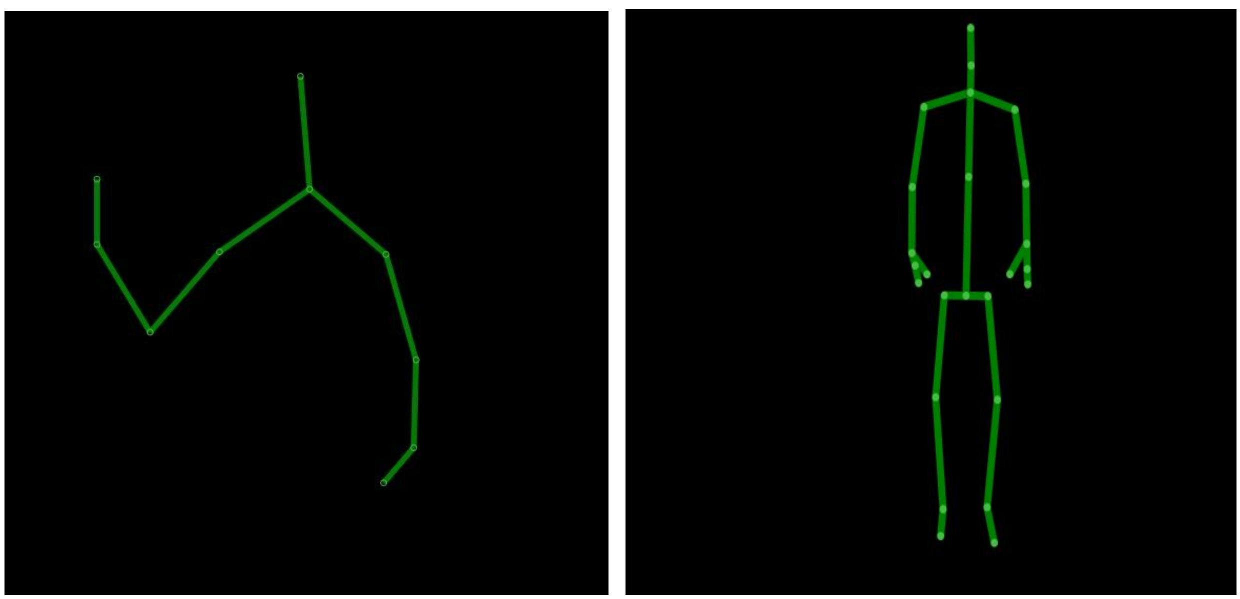
II. Player Component

The patient will be able to control the character as it moves and interacts with the world. The basic movements for the player are walking or running forward, walking backwards, strafing left, strafing right, jumping, and rotation in both clockwise and counterclockwise directions. Diana used the Bullet physics engine in developing the realistic physics for the player which involved using the forces of gravity, friction, and inertia for movement. Ray testing was also implemented to ensure the prevention of the double jump. In addition, a collision filtering system was created to allow the player to interact with all the objects within the world, but have different behaviors when certain objects have been touched. Furthermore, the camera has been created so it will align with the player's field of vision.

Figure 1. Basic collision detection.

III. Kinect Component

The Kinect component of our game is what the patient will be interacting with to control the player in the game. Simon has worked on incorporating the Kinect component. He integrated the Kinect library so that the patient can easily interact with the game with movements of their body. He also worked on a calibration system for the game so that patients with less mobility can still play the game with their own comfortable movements. Simon used the Kinect to track the patients' skeletal movements while sitting or standing, such as leaning, swaying, rotating, leg lifting, and arm lifting, and translated the recorded data into meaningful movements of the player within the game. For example, a lean forward would translate into the player walking forward or a leg lift would translate into a jump.



A B

Figure 2. Joints mapped by the Kinect v1 sensor. A) Sitting-10 nodes. B) Standing-21 nodes.

IV. World Component

The world component of our game includes all of the models and visual scenery that the player is able to interact with. Jordan was in charge of implementing this via the complex GameLibrary class, a robust data driven content management system. The GameLibrary allows developers to easily create new content by defining attributes in JSON files. For example. the attributes could define the model or even the mass of the object. During the creation of the world, content from the GameLibrary is placed into the game at locations specified in a stage JSON file. The world in our game is designed to look like a tropical island. The world features a dolphin shaped island covered in palm trees. We chose a dolphin for the shape of the island because we felt that it best reflected the tropical aesthetic of the game. The player can explore the island and collect bananas. Along the way the player will encounter a broken bridge which they must carefully navigate. At the end of the banana trail the player will discover a treasure chest filled to the brim with gold and precious jewels. a hearty reward for such a grand adventure.

V. Game Component

Once the player, kinect, and world components are completed, we worked together as a group to merge all of our separate parts into a playable game. In addition, we also worked on other components of the game to make the game more enjoyable. For example, Diana worked on a menu system, Simon created the 3D player model using 3DS Max and created animations for the player so that movements in the game are more realistic and natural, and Jordan created the rest of the 3D models viewed in the world.

VI. Languages and Tools

C++ - A general purpose programming language designed for system programming. This will be the primary language used for the game.

OGRE – A 3D graphics engine used to rendering scenes to the monitor. Ogre will be used to render our game's 3D models and the world.

Bullet – A physics engine that simulates collision detection and soft and body rigid body dynamics. Our game will utilize this as the main physics engine for creating the realistic effect of balancing.

Visual Studio – An integrated development environment by Microsoft. We will be using this as the main environment for coding the game.

Kinect – A motion sensing input device developed by Microsoft. This will be used as a peripheral for users to control and interact with the game.

3DS Max – A 3D computer graphics program for creating 3D models, games, animations, and images. We will use 3DS Max as our tool for creating content for our game.

Photoshop – A graphics editor developed by Adobe Systems. Like GIMP, we will use this retouch and edit our game content.

Fuse – A 3D character creator software developed by Mixamo. We will use this to create 3D animations of any character we create.

FL Studio – A digital audio workstation developed by Image-Line. FL Studio will be our main tool for generating the audio features of our game.

GitHub – An online repository used easy version control and collaboration. Our project will be stored on GitHub in order for our group to work on the most updated code.

Javascript – A dynamic programming language used in web browsers. Our web application component will be mostly implemented using javascript.

Project Implementation

I. Results

At the end of the semester, we were able to implement a playable puzzle platformer game that allows the player to explore an island, collect bananas, and search for the lost treasure. To control the player, the user would have to use trunk rotation or marching in place. A menu was implemented that allows the user to calibrate the Kinect prior to playing and/or disable the use of the Kinect so only the keyboard would be used as the input control. The game contains a simple scoring system that increments each time a banana is collected and a message that displays when the user has beat the game by finding the treasure chest.

II. Obstacles

Unfortunately, our expected timeline for the project did not align with the actual time it took to complete it. We faced a few obstacles that hindered the speed of our progress, most of which originates from the lack of familiarity with tools. For instance, none of us had familiarity of C++, OGRE, or Bullet prior to taking on this project which took time to learn. Another obstacle that we had to face was the limited amount of documentation found in the Bullet library. Although there were some tutorials and examples available, it was very limited and overall not very helpful in helping us build our game engine.

III. Expected Project Timeline

Week	Date	Description
1	2/9 - 2/13	First meeting with sponsor. Discuss and understand the overview of the project. Attain the necessary tools and setup the environment needed for project completion.
2	2/16 - 2/20	Get familiar with tools and SDKs needed for the project such as the Microsoft® Kinect. Go through tutorials for C++. Research more efficient ways in generating 3D models.
3	2/23 - 2/27	Continue familiarizing with tools, kits, and languages. Design and build a stage and level builder for our game engine. Create a moveable player.
4	3/2 - 3/6	Continue building stage and level builder. Brainstorm game content. Research OGREbullet.

5	3/9 - 3/13	Build a single stage using a stage builder. Finish level builder. Generate basic prototype of a stage or level. Create basic physics engine for player movement. Create game content.
6	3/16 - 3/20	50% Functionality: Finish basic prototype of a single stage or single level. Use physics engine to handle collisions.
7	3/23 - 3/27	**Midterm Presentations 3/27** Create more stages. Continue generating game content and using the physics engine to handle physics-related movements.
8	3/30 - 4/3	**Midterm Presentations 3/30** Create more stages for each type of movement and build more levels. Continue generating game content and using physics engine.
9	4/6 - 4/10	Have well developed dynamics of movement of the player and interaction with the world and enough stages to create a game prototype.
10	4/13 - 4/17	Test prototype. Fix bugs. Add additional features and stages.
11	4/20 - 4/24	Test prototype. Fix bugs. Add additional features and stages.
12	4/27 - 5/1	Continue testing the prototype and fix bugs. Continue adding additional features and stages.
13	5/4 - 5/8	**Final Presentations 5/8** Code cleanup. Test for bugs and efficiency issues. Work on project documentation.
14	5/11 - 5/14	**Final Presentations 5/11** Prepare final deliverables. Work on project documentation.
15	5/18 - 5/22	**Final Deliverables Due 5/18** Prepare final deliverables.

***the above schedule is tentative and not accurate, events may be subject to change.**

IV. Actual Project Timeline

Week	Date	Description

1	2/9 - 2/13	First meeting with sponsor. Discuss and understand the overview of the project. Attain the necessary tools and setup the environment needed for project completion.
2	2/16 - 2/20	Get familiar with tools and SDKs needed for the project such as the Microsoft® Kinect. Go through tutorials for C++. Research more efficient ways in generating 3D models. Work on Project Specification.
3	2/23 - 2/27	Continue familiarizing with tools, kits, and languages. Work on Project Specification. Add OGRE and Kinect configurations. Research on different 3D Modelling programs.
4	3/2 - 3/6	Brainstorm game content. Research OGREbullet. Analyze different Kinect sensors.
5	3/9 - 3/13	Create basic physics engine for player movement. Create game content. Brainstorm different ways of implementing Stage Builder.
6	3/16 - 3/20	Use Bullet physics engine to handle simple collisions. Create game content. Work on Midterm Presentation. Skeleton recognition from Kinect.
7	3/23 - 3/27	**Midterm Presentations 3/27** Create test stage. Continue generating game content and using the physics engine to handle physics-related movements. Basic movement with Kinect.
8	3/30 - 4/3	**Midterm Presentations 3/30** Continue generating game content and using physics engine. Add more differents motions for Kinect.
9	4/6 - 4/10	Have well developed dynamics of movement of the player and interaction with the world and enough stages to create a game prototype. Animations and integrate Kinect motion with player movement.
10	4/13 - 4/17	Merge components of player movement, Kinect detection, and stage building. Improve collision filtering and interaction.
11	4/20 - 4/24	Merge components of player movement, Kinect detection, and stage building. Improve Kinect motions.
12	4/27 - 5/1	Finish prototype for stage. Add additional features like menu. Prepare for final presentation. Improve Kinect calibration.

13	5/4 - 5/8	Finish prototype. Test for bugs and efficiency issues. Add more game content (3D models). Prepare for final presentation. Improve Kinect calibration.
14	5/11 - 5/14	**Final Presentations 5/11 and 5/13** Prepare final deliverables. Work on project documentation and update project specification.
15	5/18 - 5/22	**Final Deliverables Due 5/18** Prepare final deliverables.