Department of Computer Science & Engineering
Indian Institute of Information Technology Senapati

Kinect-Based Exercise Game

(Course Code: CS 200)

Mohd Shadab
18

Supervised by
Dr. Navanath Saharia

24 June 2020

Abstract

This project is designed for heart patients,people who want to do physical exercises as well as for gaming enthusiasts. It is based on Kinect Skeleton Tracking.

Chapter 1

Introduction

My project is based on body skeleton tracking. It is a game where one can control every joint of the main character without any controllers attached to themselves. It is mainly designed for heart patients who are prescribed for certain routine exercises. The idea is to perform the necessary exercises through a gaming environment and thus making it really interesting and enjoyable. I used Kinect for tracking the body motion and Unity to design the virtual environment.

1.1 Problem statement

The problem is to design a system for people/heart patients who are prescired for daily cardiovascular exercises. Using this system, they can do that with fun . They don't have to wear any gear,like VR, which becomes uncomfortable for these people.

1.2 Motivation behind selection of the project

This is a digital age where people invest most of their time on internet and don't get involved much in physical activities or exercises. People dont have time to go out to gym. Few people start exercises at home but find it really boring and most of them quit it and go back to their unhealthy lifestyle.One solution is to have a gym setup at home, but its not feasible for most of people because its really expensive to install a gym setup at home. Lack of physical activites leads to diseases. So I thought to take a initiative about this and came up with this idea.

1.3 Roadmap

Designed the world scene in unity where all the gaming actions take place. Made different types of 3D obstacles that keep coming floating at the user,he need to dodge it and thus it will make him perform exercises without him even noticing the purpose of the game! Modelled the character in Blender with meshes and did a little scultping on it too. Painted the textures in Photoshop and added rigs/skeleton to the character.Then animated it in Blender so that it can move its limbs naturally. Then mapped the characters joints with joints of the bodyframe object used by the Kinect. This gives the player full control of every limbs / joints of the character. Which means if we raise our hand in front of the Kinect, the character will also raise its hand.This way the model will be in the gaming world and we have to help him dodge obstacles otherwise will lose! Designed all the game logics in Unity and Visual Basics using C# coding.

1.4 Your contribution

The obstacles are designed in a such a way that the user automatically performs cardiovascular exercises without even noticing or worrying about the exercise. Also there are no strings/controllers/headset attached to the player hence it is really comfortable to use. All they have to do is to perform free actions according to the game

1.5 System/Software used

Models designed in Blender verion 2.80
All the levels and UI designed in Unity verion 2019.
All the scipting done in Visual Studio 2019.
Textures paint done in Photoshop.

1.6 Implementation plan

I designed the world scene in unity where all the gaming actions take place.

Then I modelled my character in Blender and added rigs/skeleton to it.

Then I mapped the characters joints with joints of the bodyframe object used by the Kinect. This gives the player full control of every limbs / joints of our character.

Then I designed all the game objects in Blender and Unity.

Then I designed game concepts and behaviours of objects/interactions in Unity C# using class data structure since it is most popular and reliable language for game scripting and game is all about objects and behaviours,hence class is excellent for this job.

I have used unity debugger in the Visual Studio for the C# codes. Unity editor is great for putting our game world together, but we can't write our code in it. With Visual Studio Tools for Unity, we can use the familiar code editing, debugging and productivity features of Visual Studio to create editor and game scripts for our Unity project using C#, and we can debug them using Visual Studio's powerful debugging capabilities.

But Visual Studio Tools for Unity is more than that, it also has deep integration with Unity editor so that we'll spend less time switching back and forth to do simple tasks, provides Unity-specific productivity enhancements, and puts the Unity documentation at our fingertips.

Chapter 2

Related work

i) Exercise recognition for Kinect-based telerehabilitation

by D. Antón; A. Goñi; A. Illarramendi Department of Languages and Information Systems, University of the Basque Country UPV/EHU Donostia-San Sebastián, Spain

Introduction: An aging population and people's higher survival to diseases and traumas that leave physical consequences are challenging aspects in the context of an efficient health management. This is why telerehabilitation systems are being developed, to allow monitoring and support of physiotherapy sessions at home, which could reduce healthcare costs while also improving the quality of life of the users.It is a Kinect-based algorithm that provides real-time monitoring of physical rehabilitation exercises

Results: Two relevant aspects of the algorithm were evaluated in our tests, classification accuracy and real-time data processing,. We achieved 91.9 percent accuracy in posture classification and 93.75 percent accuracy in trajectory recognition. We also checked whether the algorithm was able to process the data in real-time. We found that our algorithm could process more than 20000 postures per second and all the required trajectory data-series in real-time, which in practice guarantees no perceptible delays. Later on, we carried out two clinical trials with real patients that suffered shoulder disorders. We obtained an exercise monitoring accuracy of 95.16

Conclusions: We present an exercise recognition algorithm that handles the data provided by Kinect efficiently. The algorithm has been validated in a real scenario where we have verified its suitability. Moreover, we have received a positive feedback from both users and the physiotherapists who took part in the tests.

ii) A Kinect-Based Rehabilitation Exercise Monitoring and Guidance System

Introduction: Our system is the first rule-based system for rehabili- tation exercise monitoring and guidance with realtime visual feedback. A set of basic rule elements are developed such that they can be used to define correctness rules for common rehabilitation exercises. This enhances the adaptability of our system to accommodate new rehabilitation exercises and the adjustment of existing correctness rules when they are needed. Facilitated by the Unity framework, the motions are rendered via 3D avatars with frame-by-frame replay control and full 360-degree view. Hence, our systemallows the patient to study the prescribed exercises, and the clinician to examine the patient sessions in great detail. Furthermore, our system preserves the privacy of its users while enabling the exercise session to be captured for full examination.

Conclusion: In this paper, we described a rule-based system that facili- tates in-home rehabilitation exercise monitoring and guidance. The core foundation enabling realtime assessment in our system is a rule-based approach. For each exercise, we define a set of correctness rules and they serve as the invariance of the exercise which is independent from the size and form of the users. We designed a set of basic rule elements that can be used to express the correctness rules for various common rehabilitation exercises. Our system may be operated in a number of modes. In addition to providing guidance and realtime assessment, our system can be used to study the demonstrated exercise and the recorded patient session frame- by-frame with 360-degree view. Hence, we believe that our system can be of practical use for rehabilitation exercises.

Comparison:
"Exercise recognition for Kinect-based telerehabilitation" is a system closely related to mine. Similar to my system, this system provides real exercise experience to the user . However, It uses a different approach called Programming-byDemonstration. In this approach, a model is derived from the recorded motion data of a demonstration by an expert. When the user performs the exercise, it is compared to the previous datasets recorded by the experts.But my project doens't require any prequisite dataset to work.The user is free to make their moves without thinking about coreectness of their movementst.Theyonly to need to play and enjoy the game. "A Kinect-Based Rehabilitation Exercise Monitoring and Guidance System" also proposed a system to facilitate in-home exercise assessment. The basic procedure is rather similar to that of "Exercise recognition for Kinect-based telerehabilitation" . The main difference is that statistical algorithms are employed to compare the motions of the demonstrator and those of the user. While potentially robust, this approach is significantly more computationally intensive and hence the assessment is done off-line instead of in realtime. Furthermore, the feedback contains only three categories: excellent, good, and bad. On the contrary, the other system can provide specific feedback regarding which rule is violated.Whereas my system doesn't require any such complicated computations. Thus making the source easier to managae and improve.

...Method(s) used for system study

2.1 Comperative analysis

"Exercise recognition for Kinect-based telerehabilitation" is a system closely related to mine. Similar to my system, this system provides real exercise experience to the user . However, It uses a different approach called Programming-byDemonstration. In this approach, a model is derived from the recorded motion data of a demonstration by an expert. When the user performs the exercise, it is compared to the previous datasets recorded by the experts.But my project doens't require any prequisite dataset to work.The user is free to make their moves without thinking about coreectness of their movementst.Theyonly to need to play and enjoy the game.

"A Kinect-Based Rehabilitation Exercise Monitoring and Guidance System" also proposed a system to facilitate in-home exercise assessment. The basic procedure is rather similar to that of "Exercise recognition for Kinect-based telerehabilitation" . The main difference is that statistical algorithms are employed to compare the motions of the demonstrator and those of the user. While potentially robust, this approach is significantly more computationally intensive and hence the assessment is done off-line instead of in realtime. Furthermore, the feedback contains only three categories: excellent, good, and bad. On the contrary, the other system can provide specific feedback regarding which rule is violated.Whereas my system doesn't require any such complicated computations. Thus making the source easier to manage and improve.

Limitations of the existing systems:

These systems are too much formal. User/patient may lack interest using them.User is restricted to perform and match the actions already fed in the system.He/She need to match the positions which he/she may find boring and may lack interest after using for few times.

Features of my System:

My project will be more enjoyable where user will perform exercises in a gaming environment with a exciting gaming experience.User don't have to match postures or exercises. He/She only need to focus on playing, dodging the obstacles which will come to them.

- This application/game gives all the advantages of cardiovascluar exercises.
- It is designed keeping in my mind the cardiovascular exercises.
- Apart from heart patients, anyone can play this Game and enjoy.
- There are no strings/controllers/headset attached to the player hence really comfortable to use. All they have to do is to perform free actions according to the game.

- The obstacles are designed in a such a way that the user automatically performs cardiovascular exercises without even noticing.

2.2 Summary

My software is totally free to use.The user only need to buy a a Kinect Hardware and that's it!Kinect is much cheaper than a gym setup at home.
The proposed system can run on any machines supporting Windows OS . The best software and hardware that had been used while designing the system so it would be feasible in all technical terms of feasibility.
The Kinect hardware is really cheaper than the cost of affording a gym setup at home or having a personal physician thus making it useful for people who can't afford costly gym setups.

My application is more joyable to use and you are free to use your tactics as well . Also my application is more comfortable to use because of lack of any compulsory gear other than your body to play.

Chapter 3

System Design

3.1 System design

- Open the application.
- Connect Kinect.
- Stand in front of Kinect so it can detect your body structure.
- Move your hands, legs and body according to the game requirements and score!
- The game levels are designed keeping in mind the cardiovascular exercises, so playing it will make you healthier, and will decrease heart disease chances and will minimize it if you already have .

3.1.1 Architecture

Obstacles: are the cubes floating in the games which needs to be dodged by the player.
Components: Mesh Renderer gives the obstacle a 3D form. Rigidbody makes it behave like a real obstacle in 3D space. Obsmovement provides motion and force to it. Material gives a color to our object.

Third Person Controller: It is our main character in the game.
Components: Avatar Controller Script responsible for mapping of character to a real person.Player Collision script keep tracks if an obstacle hits the player. Rigidbody makes it behave like a real obstacle in 3D space. Animator is responsible for the riggs of the character.

Plane: It is ground on which everything takes place.
Components: Mesh Renderer gives the obstacle a 3D form. Mesh Collider used to detect collisions. Ground 1 material gives color to the plane.

End Wall: It is the wall that detects the completion of level ,by detecing the collision of the last obstacle to it.
Components: Rigidbody makes it behave like a real obstacle in 3D space. Endtrigger script makes it detect the end condition for the completion of level.Boxcollider gives property of a real -like box in 3d space, which can collide with other objects.

Canvas: It is a 2d UI panel that displays the states and score of the game.
Components: PausemenuUI displays Pause menu. It consists of 3 buttons -NEW GAME, OPTIONS, QUIT. Score Panel displays the score on top of screen. LevelCompleteUI get activates when level is completed. GameOverUI panel gets activated when game is lost.

Gamemanager: Manages the level complete and gameover UIs.
LoadScene: Responsible for management of all scenes.

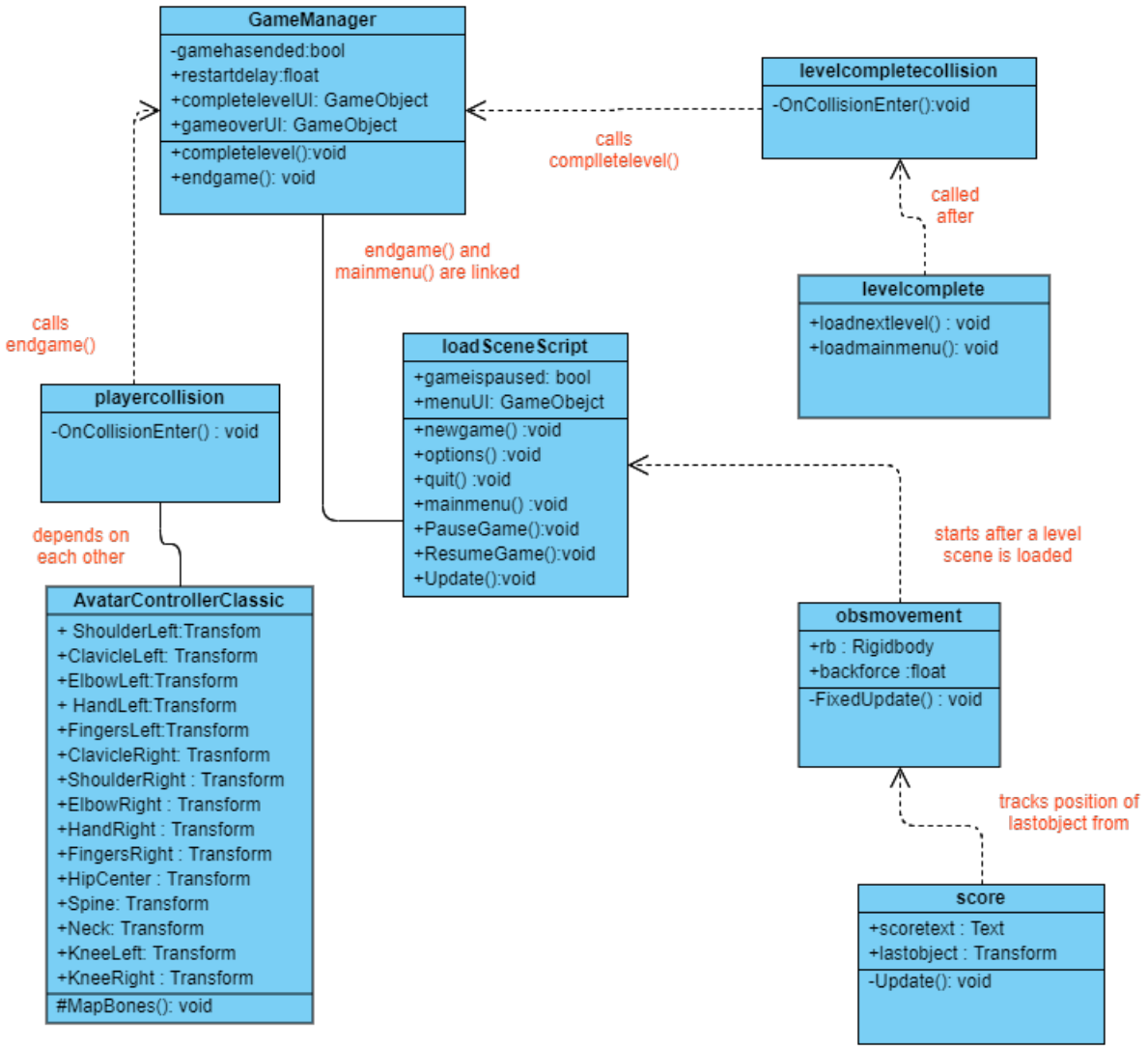
Object- Component Relationship

Chapter 4

Implementation

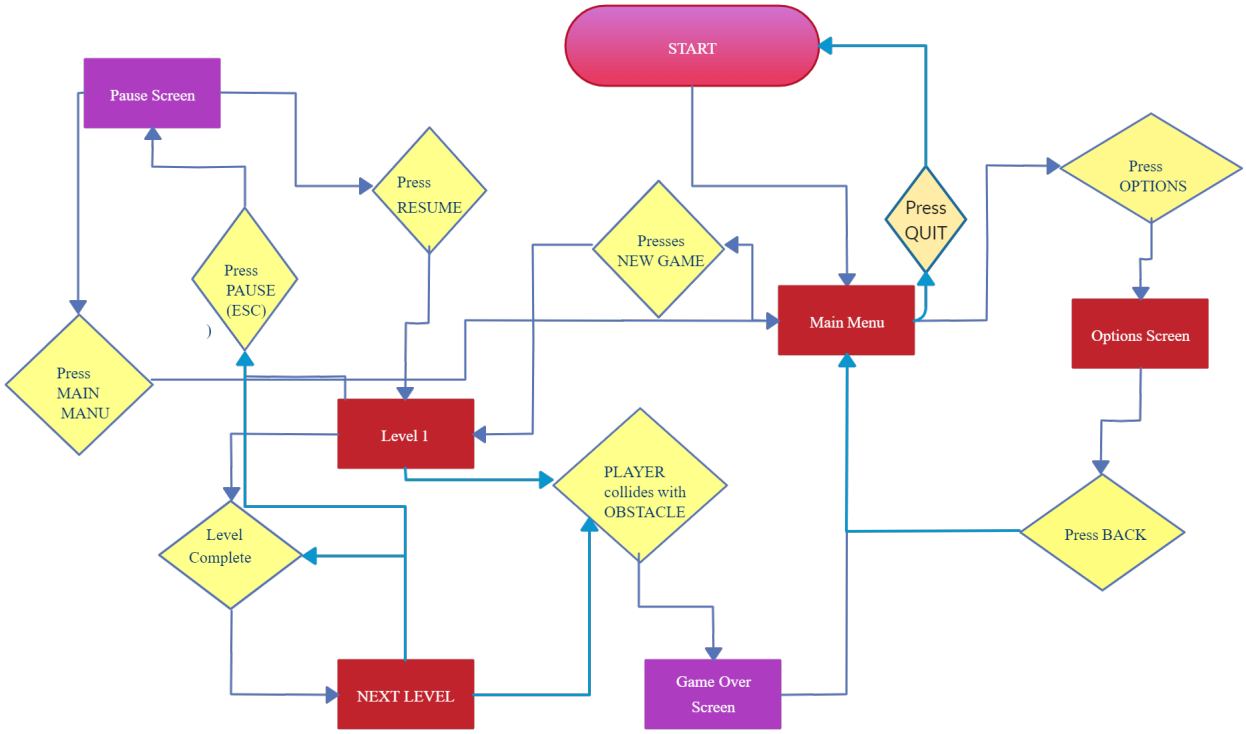
4.1 Experimental set-up description

- Class diagram with Description:



The class diagram includes the entities and behaviours of the classes used in the game.These are well described in section 4.1.1

- Control flow diagram



The flow diagram indicates the states and levels of the games at different times accordingly. When we start the game, we have a Main Menu with 3 button options- NEW GAME ,OPTIONS, QUIT. Pressing New game, Level 1 begins. Pressing options, An Option menu pops up with volume settings.Pressing quit will take you the start screen. While in Level1, if we hit an obstacle, game is lost and game over Screen is displayed and then Main menu appears. on completing the level, level complete screen pops up and we are taken to level 2 and it goes on again. While in game, pressing Pause(ESC) , Pause menu appears with options of resuming game, going to main menu and quitting the game.

4.1.1 Function/Module/Class description

Description of Class and their behaviours:

1. loadscenescript : Responsible for loading all scenes depending on the calls.
    - newgame() : Loads level 1 of the game when called.
    - options(): Loads the options screen when called.
    - quit() : Quits the game when called.
    - mainmenu() : Loads the mainmenu.
    - PauseGame() : Pauses the game by setting timescale = 0 and activating Pause Panel.
    - ResumeGame():Resumes the game by setting timescale = 1 and deactivating Pause Panel.
    - Update() : function to keep track whether ESC (pause) is pressed or not.
2. gamemanager :Manages the events like game over and level complete.
    - completelevel1() :Activates LEVEL COMPLETE UI.
    - endgame(): Activates GAME OVER UI.
3. obsmovement : Controls the force and its direction for the obstacles.
    - FixedUpdate(): Responsible for motion of obstacles.
4. levelcompletecollision : Detects the collision of last obstacle which marks the winning of level.
    - OnCollisionEnter(): Detects if last object has been dodged by the player hence game won and calls completelevel().
5. levelcomplete : Contains the functions which are called upon the completion of levels.
    - loadnextlevel(): Loads next level after current level is completed.
    - loadmainmenu() :Loads main menu after gameover UI.
6. score : Keeps track of the score .
    - Update() : Keeps updating the score of player by tracking the position of last obstacle.
7. AvatarControllerClassic : Responsible for mapping of real human with virtual character
    - Mapbones(): Responsible for mapping of player body parts to map with our character.
8. PlayerCollision : Responsible to detect the losing conddtion of game.
    - OnCollisionEnter(): Called automatically when an obstacle hits the player

4.2 Obtained result

The system (game) works as expected. On starting the game, Main menu loads.There are 3 option buttons namely Main Menu, Options and Quit.
Clicking on Main menu we go straight to level 1. The Kinect initializes the limbs and bones of the real person with the game character. Person needs to dodge the obstacles coming at him by doing appropriate actions. In level 1, he has to walk past the obstacles. Thus level 1 is a basic warm up but with a whole new Experience of walking /running ! It is quite enjoyable and beneficial ! Level 2 is a bit more difficult and is designed for exercise of upper body and neck.

Chapter 5

Result analysis and Testing

The game application runs smooth though some tuning is required with the obstacles' positions and force to make it align perfectly with the goals presumed .The levels needs to be more engaging and eye catching.

Chapter 6

Conclusion and Future work

My software is totally free to use. The user only need to buy a Kinect Hardware and that's it!
The proposed system can run on any machine supporting Windows OS. The Kinect hardware is really cheaper than the cost of affording a gym setup at home or having a personal physician, thus more beneficial for people who can't afford gym setups at home.

In future, more levels will be added to the game keeping in mind the workouts of different body parts.
I will be including VR features. To make it more efficient and powerful. To improve its user-interface with 3D environment where one can find himself in another world. There is a future scope of this facility that many more features such as multiplayer,more exciting and practical levels added to this project thus making it more interactive , more user friendly , more enjoyable and thus a project which fulfills each users need in the best way possible.There will also be a feature to record and analyse the movements,positions and steps performed by the user which will be very helpful to guide him/her throughout his exercise period where they can learn what previous mistakes they did in their last session /game play.

Chapter 7

Bibliobgraphy

- https://www.instructables.com/id/Kinect-SDK-Hello-World/
- http://www.learn.unity.com
- https://shiffman.net/p5/kinect/
- https://www.tutorialspoint.com/csharp/