DEPARTMENT OF INFORMATICS ENGINNERING

SCHOOL OF ENGINEERING

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

M.Sc. IN INFORMATICS & MULTIMEDIA

MASTER THESIS

Exergames for Parkinson's disease patients using Microsoft Kinect

NIKOLAOS S. PAPADOPOULOS

SUPERVISOR

Dr. IOANNIS PACHOULAKIS

HERAKLION
2015

Abstract

Physical rehabilitation can be beneficiary for the physical condition and the mental health of patients with Parkinson’s disease (PD), especially if it is conducted on a daily base. Although due to the lack of interest, PD patients usually avoid to participate in long-term repetitive exercise programs. Exergames can address the specific problem by combining physical training with a playful and immersive game environment. Recent studies have proved that exergames are feasible for PD patients and revealed the necessity for custom-based game solutions using exercises that target PD.

The current thesis presents a game platform and two 3D exergames designed and developed in the Unity game development platform, using the Kinect sensor as a motion-capture device.

Table of Contents

1. Introduction 6
  1.1 Parkinson's Disease 6
  1.2 Rehabilitation practices in Parkinson's disease 7
2. State of the art 9
  2.1 Tools and methodology 9
  2.2 Related Work / Solutions 9
3. Solution Specification 13
  3.1 Requirements 13
  3.2 Game scenario design 15
  3.3 Technology Specification 18
    3.3.1 Microsoft Kinect 18
    3.3.2 Unity3D 21
    3.3.3 Middleware: Kinect with MS-SD 22
    3.3.4 Autodesk 3ds Max 22
    3.3.5 Autodesk MotionBuilder 22
  3.4 Software architecture 23
4. Posture and Gesture detection 24
  4.1 Personalization and T-Pose detection 26
  4.2 Pairing Gestures to Animations 28
    4.2.1 Balloon Goon Game Gestures and Animations 29
    4.2.2 Skiing Game Gestures and Animations 31
    4.2.3 Menu Gestures 34
5. Game design 35
  5.1 Game Logic 35
  5.2 Game Scene design 41
6. Discussion and Future Work 43
7. Bibliography 44

List of Figures

Figure 1: Balloon Goon Game: a frontal arm extension is required to pop a balloon falling along the right hand post 17
Figure 2*:* Cross country ski: lean left (insert) to switch to the left lane. 17
Figure 3*:* Exergame using Kinect sensor proposed development workflow. 18
Figure 4*:* Hardware components, embedded on the Kinect device array 19
Figure 5: Angle view range of Kinect sensor, vertically and horizontally, by selecting the default settings 19
Figure 6*:* Left: Joints supported by the Kinect SDK v1.8. Right: Joints detected in seated vs standing mode. 20
Figure 7: Software architecture proposed for both exergames developed 23
Figure 8: Gesture detection algorithm. 25
Figure 9: Joints filtering proposed by Microsoft for gesture detection. 26
Figure 10: The T-pose gesture identifies the 20 joints tracked by Kinect and their derivative body measurements. 27
Figure 11: Various snapshots on the way to a T-pose gesture. The gesture is identified in insert icon #4. 28
Figure 12: T-pose detection inside the game environment. 28
Figure 13: Single Hand Raise & Extend gesture is shown executed using the right hand. 29
Figure 14: Animation key frames during the development process of Single Hand Raise & Extend gesture. 29
Figure 15: Single hand raise and extent gesture avatar animation during gameplay, by performing the gesture using the left hand. 30
Figure 16: Single Leg Raise & Extend gesture is shown executed using the right foot. 30
Figure 17: Animation key frames during the development process of Single Leg Raise & Extend gesture. 30
Figure 18: Single Leg Raise & Extent gesture avatar animation during gameplay, by performing the gesture using the right foot. 31
Figure 19: Raise & Lower Both Hands gesture is shown. 31
Figure 20: Animation key frames during the development process of Raise & Lower Both Hands gesture. 31
Figure 21: Start skiing animation by detecting Raise & Lower Both Hands gesture. 32
Figure 22: Side Lean Core gesture is presented by strerching to the left side. 32
Figure 23: Animation key frames during the development process of lean left gesture, that results or making the 3D character turn left 33
Figure 24: Turn left during Lean To One Side gesture detection inside game environment, leaning the core to the left. 33
Figure 25: Squat gesture detection 33
Figure 26: Animation key frames during the development process of squat gesture. 34
Figure 27: Animation during gameplay while squat gesture is detected. 3D character is lowering gaining more speed. 34
Figure 28: Rotate Shoulder gesture by steering right. 35
Figure 29: Raise Both Hands gesture 35

Figure 30: Balloon Goon general game logic diagram. 36
Figure 31: Balloon Goon gameplay. Upper left image shows the countdown counter; Upper right image is displaying the pause menu screen; Middle images show snapshots of the gameplay, with the middle right displaying popping a bonus balloon; Lower left image shows the end game screen, by displaying the progress of the user; Lower right image displays the menu screen that appears in order to proceed to the next level. 38
Figure 32: Skiing general game logic diagram....... 39
Figure 33: Skiing gameplay. Upper left image is displaying user performing a squat in the speed lane; Upper right image is displaying that virtual character performing a high jump collecting the star; Middle left image shows user avoiding an obstacle by leaning to the left, while the middle right one displaying animation triggered when falling on obstacle; Lower left image shows virtual character collecting rings; Lower right image displays the screen indicating the score of user upon finishing the game....... 41
Figure 34: Hand and foot pillar 3D objects 42
Figure 35: Balloons as shown in order in the figure are the balloon popped with hand gestures, the balloon popped with foot gesture, the bonus balloon and the bomb balloon. 42
Figure 36: Skiing game static scenery game objects are composed by trees, fences and the game terrain. 43
Figure 37: Skiing game premade grouped game objects. In the left it is presented the virtual character along with the ski equipment, in the middle the speed lane and the ramp group object is shown and at the right image the obstacle premade object is displayed....... 43

1. Introduction

1.1 Parkinson's Disease

Parkinson's disease (PD) is a neurodegenerative condition which affects parts of the brain that control body movement. More precisely, PD results from the loss of dopamine-producing neurons in a mid-brain location known as "substantia nigra", responsible among others for the smooth and purposeful coordination of body muscles. It is still unclear why these cells are lost, making PD so far untreatable [1],[2]. The ailment was named after James Parkinson, who first recorded and reported the symptoms. According to the European Parkinson's Disease Association (EDPA) [1], around 6.3 million people suffer from the disease worldwide. Symptoms progress slowly but irreversibly, so that late stages are more abundant in the elderly (>60 year old) population, while only approximately 10% of PD patients are under 50 years old.

In early stages, PD commonly affects motor function, while cognitive, behavioral and mental-related symptoms are usually met at more advanced stages [3]. Non motor-related symptoms include sleep disturbances, depression, anxiety, psychosis and visual hallucinations, cognitive impairment, pain and fatigue. In addition, four distinct, fundamental non-motor symptoms can be grouped under the acronym TRAP: Tremor, Rigidity, Akinesia (or bradykinesia) and Postural instability [4]. Any of these symptoms hinders common daily activities and troubles patients' social relationships, thus reducing the quality of life, especially as the disease progresses [5],[6].

Tremor can be distinguished in tremor at rest and postural tremor. Tremor at rest is the most ordinary symptom, appearing at some point during the disease among the 75% of the patients [7], making it the most distinctive and easily recognized sign of the disease. It is described [1] as an unintentional and rhythmic movement of body parts (limps, head, and face sections) while relaxed. Meanwhile postural tremor, associated with body movement (e.g. walking) or a controlled still posture (e.g., standing) can easily be misdiagnosed as essential tremor in the absence of other symptoms [8]. Rigidity, on the other hand, describes the inability of limb muscles to relax, providing high resistance during passive movement of the limb, known as cogwheel phenomenon. In more advanced stages, the muscles of those limbs can be described as stiff and highly inflexible. PD patients may also suffer from pain due to the rigidity, such as 'painful shoulder', one of the trait characteristics for PD [4]. According to the EPDA [1], akinesia (or bradykinesia) is one of the three main signs of the PD condition and refers to the slowness of PD patients in carrying out a voluntary movement, rather than initiating one. Akinesia can be present to the 78-98% of PD patients [8] troubling them during the entire course of the disease. In order to detect akinesia, PD patients are usually asked to repeatedly perform rapid movements. Finally, Postural Instability expresses the lack of postural reflexes during standing, walking or interacting with objects in space [9] and is usually absent in early stages of PD. The symptom depends on the severity and course of the disease [10] and is highly correlated to the frequency of patients' falls. Postural instability combined with akinesia can be an especially dangerous mix which can lead to severe injuries.

In order to assess PD progress as well as a patient's level of disability, several rating scales have been proposed. In 1967 Hoehn and Yahr [11] proposed a rating scale that is characterized by simplicity and ease of application. It recognizes three practical classification types for Parkinsonism (primary, secondary and indeterminate) and contains a 5-stage system for grading the severity of the condition ranging from insignificant motor impairments in the first stage, to severe impairments in late stages. It focuses on observations such as unilateral or bilateral expression of the disease as well as the degree of postural reflex impairment. In order for the rating scale to become more detailed, several additional stages have been proposed which include non-motor aspects of the disease and describe motor aspects more precisely [12]. In 1980 the Unified Parkinson's disease Rating Scale (UPDRS) [13] combined several elements from previously introduced rating scales. The scale has been further updated by the Movement Disorder Society (MDS) [14] in order to include new aspects of non-motor symptoms. UPDRS consists of three major sections, which evaluate significant areas of disability (Part I: Mentation, Behavior and Mood; Part II: Activities of daily living; Part III: Motor Function). The scale is accompanied by a fourth section that evaluates complications in treatment. UPDRS is the most widely used clinical rating scale and according to EPDA is commonly used in tandem with the Hoehn and Yahr as well as the Schwab and England Activities of Daily Living (ADL) scales. It must be noted that the ADL scale provides a useful measure of a person's capability in performing daily activities, and as a result of independence.

1.2 Rehabilitation practices in Parkinson's disease

Although a cure has not been found thus far for PD, medication usually helps contain the symptoms and maintain body functionality at reasonable levels through the lifetime of the patient. According to information provided by the American Parkinson Disease Association (APDA) [15] six categories of drugs are proposed for PD condition therapy: levodopa, dopamine agonists, MAO-B inhibitors, COMT inhibitor and levodopa, anticholinergic agents and amantadine. Standard medication paths cannot be easily achieved, as the progress and the symptoms of the disease varies significantly among patients [16]. Based on clinical practice, Levodopa is considered as the most efficient drug for improving PD-related motor symptoms, especially at the onstage of the disease, although with large doses over an extended period of time have been connected to symptoms like dyskinesia. It is a common practice during the last decade to start a patient's treatment by providing agnostics and start providing levodopa at a later stage when the motor symptoms cannot be controlled anymore [17]. In advance, several combinations of levodopa, dopamine agnostics, COMT inhibitors and MAO-B may be used during the course of the disease in order to eliminate the effects of the PD condition symptoms and achieve optimal results [16].

Adding to the value of medical treatment, physiotherapy appears highly effective in reducing or containing PD-related symptoms. A number of Parkinson clinical facilities and associations provide physical activity guidelines, suggesting daily activities and tasks, even diet schedules. For example the Parkinson Society of Canada [18] provides online detailed instructions on how to correctly perform stretching and other physical exercises. Exercise interventions in randomized controlled trials [19] prove that physical exercise such as stretching, aerobics, unweighted or weighed treadmill and strength training improves motor functionality (leg stretching, muscle strength, balance and walking) as well as patients' health-related quality of life. It is worth mentioning that one training program was conducted in the patients' homes instead of at a clinical facility using exercises tailored to the condition of each patient. In this case, a physiotherapist was visiting them on a weekly base [20]. In addition, participants kept a record of fall events for the duration of the program. Analysis of these records reveals lower fall rates for patients that follow home-based exercise programs with detailed preparation and documentation compared to those who do not. These results are corroborated by a different study [21] which employed experimental balance training including self-destabilization exercises, externally-forced destabilization exercises and coordination of leg and hands during walking. These exercises improved postural stability and boosted patients' confidence as a result of the reduced frequency of falls. In fact, the benefits in postural stability as a result of that exercise program were maintained for at least one month after the end of sessions.

Fringe benefits resulting from home based rehabilitation include significant cuts in treatment cots. Indeed, PD patients must frequently attend physiotherapy sessions to either notice an improvement or maintain the gains from clinical rehabilitation programs [22]. In fact, Calgar, et al. [23] point out the effectiveness of home-based, structured physical therapy exercise programs tailored to an individual patient, which is witnessed by measurable improvements in motor capability.

In addition, the "training BIG" protocol for PD rehabilitation have also shown promising results. For example, exercises that focus on amplitude training [24] can lead to faster upper and lower limb movements – in this case, participants repeatedly performed various exercises using maximum range of motion (maximum amplitude). The exercises set employs the entire body of the patient, both in seated and standing posture and include BIG stretches (i.e. reach and twist to side) and reparative BIG multidirectional movements (i.e. step and reach forward).

Several other training programs have been applied to the PD condition. Prominent among those is Tai Chi, a form of martial art based on gaining balance through the continuous movement of the body's center of mass. As several studies have proved that Tai Chi can improve strength, balance, and physical function in healthy older adults, Fuzhong, et al. [25] conducted a study to evaluate Tai Chi's potential in improving postural stability in PD patients. The randomized trials included three independent patient groups: a Tai Chi, a resistance training, and a body stretching group. Tai Chi participants showed more significant improvements both in balance and in maximum excursion compared to patients of the remaining two groups. In addition, Zhou, et al. in their systematic literature review [26] conclude that Tai Chi seems to significantly improve motor and balance impairments for PD patients, although larger-scale samples and high-quality randomized control trials are necessary to make this statement conclusive. Another promising rehabilitation exercise trial was conducted by Combs et al. [27] using box training. While the number of patient participating in this trial was relatively small (six patients), the patients showed both short and long term improvements in balance, gait, daily activities and quality of life, although more advanced stage participants seem to require more persistence and time to reap these benefits.

2. State of the art

2.1 Tools and methodology

PD patients attending long-term reparative exercise programs tend to get bored of the same daily physical rehabilitation [28]. In fact, Mendez, et al. " [29] suggest that "it is important to investigate the learning potential of patients with Parkinson's disease by applying new therapeutic strategies and validating their utility. Indeed, it is worth examining the potential benefits that can be reaped by PD patients through their participation to and practicing exergames - computer games indented to be used as an exercise tool, using motion capturing systems like Sony Playstation Eye, Nintendo Wii, Microsoft Kinect or even cameras. Such games use audio and visual cueing in loose virtual reality (VR) environments and offer an enormous motivating potential because they do away with repeatability (the seed of boredom) and engage the user in immersive goal-oriented scenarios. Motion capture devices provide an interface to the virtual world and can be programmed and tuned to provide real-time information about the specific interactions.

VR interaction seems impressively promising compared to physical reality in terms of rehabilitation not only for people with PD condition, but also for people with motor degenerative conditions. A resent review [30] conducted by Vieira et al. evaluated various studies in the literature with an eye on the possible benefits of VR-based systems for PD patients. It concluded that VR can not only be used as a therapeutic tool, but can play a significant role in controlling and regaining motor function, mobility and cognitive capacities as well as balance. In addition, participants in all home-based trials showed improvements in all post-training tests, supporting the idea of using VR as a therapeutic home-based tool. The authors continue to discuss the benefits of VR-based games on physical neuro-rehabilitation supporting that visual and auditory cues may stimulate a player's reaction at a cognitive level. Indeed, some sort of brain re-training of the brain may be possible in the sense of rechanneling brain activity through so far unused neuron paths. Other recent studies seem to corroborate not only the feasibility but also the benefits of exergames for PD patients, as they seem a highly effective rehabilitation practice [31]. At the same time, extreme care and forethought must be put forth on designing exergames specifically for PD patients. For example, exergames must provide motivation and positive feedback, be progressively more challenging but also adaptable to a specific patient's condition and all that with a "maximum safety" net from the design stage and not as an afterthought so as to minimize and even eliminate accidents such as falls.

2.2 Related Work / Solutions

Yu et al [32] developed a real-time multimedia environment aiming at PD patient's rehabilitation. The system was developed using a ten near-IR camera Motion Analysis System to capture a patient's body in 3D. In order to successfully capture movement, retro-reflected markers are attached to the patient's body. Guided by visual and auditory cueing, patients are called to execute several exercises based on the BIG protocol [24]. At the same time, incoming data streams are analyzed in real-time via motion analysis software and are mapped onto an on-screen avatar. However, the proposed solution is not cost efficient due to the need of installing multiple infrared cameras and calibrating the system. In addition, it is hard for people with PD condition to use the system on their own, without external help, due to the fact that retro-reflected markers must be attached to their body at specific points every time they need to exercise.

In the context of the EU-funded FP7 CuPiD project, Tous et al. extended the tele-rehabilitation platform Play For Health (P4H) in order to create three exergames for PD patients [33]. Three games was created, Touch 'n' Explode (pop objects in the virtual environment), Stepping Tiles (step on the tiles of the kitchen) and Up 'n' Down (sit to stand exercise in the correct way). Patient motions are translated as 3D Avatar moves in a virtual environment. Motion capture employed a Body Area Network of wearable sensors that were developed specifically for the project along with the necessary algorithms for posture recognition. These sensors additionally used to detect freezing motion incidents.

Another line of efforts, e.g. [31], evaluated the Nintendo Wii gaming system as a possible rehabilitation tool for PD. The Wii system uses a handheld controller to communicate with a console, usually installed by the display monitor / TV in-front of the user. Data such as the rotation and acceleration of the controller is sent to the console wirelessly. In addition, the "balance board", a rather common extender component used by several Wii games contains several sensors which calculate the mass of the player and his/her center of gravity. Case studies evaluated Wii as an off-the-self, cheap solution that already provided major releases in the field of rehabilitation games. Data related to the improvement of gait, balance, cognitive reaction and impaired functional mobility were recorded and presented. In most cases the Wii Balance board extender was used as it is required by a wide variety of Wii games.

Zettergren et al [34] evaluate three exergames (Penguin Slide, Table Tilt, Balance Bubble) and four activity games (Free Step, Island Cycling, Obstacle Course and Rhythm Parade). The results show significant improvements on gait speed, timed up-and-go, and on Berg's balance scale, while no signs of (psychological) depression were observed during the trial. An additional study that exhibited improvements among PD patients in terms of gait and balance was conducted by Mhatre et al [35]. The game exercise trial used the Balance board extender of Wii gaming system for the Marble game, Skiing and bubble game. Esculier [36] et al used similar Wii games to evaluate benefits in balance for PD patients and compared the results against a set of healthy elderly individuals. The trial showed improvements in most static and dynamic balance aspects for both user groups. It is also noticeable that the 83% of the participants liked the games, while the rest were neutral on the matter but no subject disliked them.

Pompeu et al [37] investigated whether PD patients can improve their performance on the Wii gaming system and compared the effects of Wii-based motor and cognitive training with balance exercise therapy in the aspect of independent performance of activities of daily living. Tested Wii games included balance games (Table Tilt, Tilt City, Penguin slide and Soccer heading) as well as static balance games (Torso Twist and Single Leg extension), while stationary gait is practiced in games like Rhythm Parade, Obstacle Course, Basic Step and Basic Run. Results showed that the PD patients improved their performance in each category of Wii-based games. Although Wii-based motor and cognition training had good impact in the independent performance of activities of daily life, the same results were observed with the balance exercise therapy executed in the real environment.

Hertz et al [38] on the other hand, used the Wii games Tennis, Boxing and Bowling to determine the effectiveness of the specific gaming system on both motor and non-motor symptoms for PD. The games were chosen as out-of-the-box solutions bundled with the Wii system (lower cost than buying as add-ons) with familiar movements to the subjects of the study. Participants witnessed improved motor and non-motor aspects as well as improvements in quality of life.

Most aforementioned studies using commercial Wii games demonstrated improvements in gait, motion and balance among PD patients. In addition, some studies state that benefits gained during the intervention period were maintained for a time period after program completion. It must be mentioned that most games developed Wii use the Balance Board raised platform as an extension. For example, in Free Step patients step on and off the Balance board in order to follow the game's sequential activities. It should be noted, however, that the Balance board is a risky piece of hardware for PD patients, as it may lead to falls [31]. In addition all Wii games require a handheld controller in order to capture the player's movement, which may be troublesome for at least some PD patients. Therefore, whereas commercial Wii exergames seem to offer low cost rehabilitation opportunities, they may not be suitable for all PD patients. In fact, Mendes et al [29] record that PD patients have trouble in learning and retention using some Wii commercial games when compared with healthy individuals, resulting in poor performances for these games. In addition, study participants failed to improve their performance as these games required fast reactions which the case subjects lacked as a result of cognitive and/or motor problems. It follows that game performance improvements for PD patients depends on the demands of the specific Wii game played.

Wii-based rehabilitation games tailored to PD patients using the Wii remote handheld controller were developed by Paraskeyopoulos et al [39]. Two games were developed based on PD-specific movements extracted from bibliography. In the first game the patient controls a two paddle row boat from a seated position to reach a specific point in a certain amount of time. In the second game (water valve mini golf game) the player rotates a valve several times releases and rolls the hands in an up/down direction to release and guide a ball through a pipe into a hole. These game exercises are intended to improve speed, rigidity, range of movement and bilateral mobility. Data retrieved from the sensor's accelerometer and gyroscope were algorithmically combined to provide improved orientation and linear motion results so as to map the user's movements onto a 3D avatar located.

Evaluating the advantages/or disadvantages of similar (to the Wii) technologies, Assad et al [28] examined the Sony Playstation Eye system as a potential rehabilitation tool for PD condition. They developed WuppDi!, a collection of five PD-oriented motion-based games various playful environments. Most of these games required one or two (one for each hand) makers in form of a glove or wooden stick to interact with virtual game objects. Participants welcomed this approach as a physical activity albeit having trouble handling the input devices. In their preliminary study they also evaluated patients in Wii gaming system and recorded that PD patients weren't able to coordinate effectively by moving the handheld controller and pushing buttons on it at the same time.

The latter two studies revealed a necessity for developing exergames that are focused on PD rehabilitation. Emphasis must be given to gaming systems that accurately capture full body motion without the need of external handheld or wearable devices. In their review, Gillian et al [31] propose that platforms such as the XBOX Kinect which do not require raised platforms, handheld controllers and/or body markers need to be evaluated with respect to the rehabilitation opportunities with maximal safety for PD patients. In addition, such systems are natural candidates for home based solutions that can be easily used by PD patients, without guided instructions about how the system and its peripherals connect.

Microsoft Kinect requires no external input controllers and can capture the motion of the entire human body in 3D, using an RGB camera and a depth sensor. Players can manipulate game interactions by just moving their body in space in front of the sensor. Kinect-based exergames tailored to PD are however sparse due to the fact that the technology is relatively new. The first generation of Kinect sensors was released in November of 2010 and was paired to the Xbox 360 console, while in June 2011 Microsoft released a Software Development Kit (SDK) to allow the creation of Kinect-based applications.

Galna et al [40] developed a Kinect game for PD patients which included specific upper and lower limb movements to improve dynamic postural control. The upper torso of the player is mapped onto a farmer avatar driving a tractor who collects fruits and avoids obstacles in a 3D environment. Fruit collecting is achieved by specific hand movements of the patient, choosing to move the right or the left hand depending on the color of the fruit. Large steps (front, back and sideways) with one food centered guide the tractor to avoid obstacles in the form of sheep, high wires, birds, etc. To maintain patient motivation, the game has several levels of increasing difficulty: game complexity increases from simple hand movements for low levels through more complex activities combining cognitive decisions and physical movements all the way to dual tasking (simultaneous hand and foot work). The design principles resulted from a workshop where participants played and evaluated a wide range of commercial games developed for Microsoft Xbox Kinect and Nintento Wii. The workshop once more revealed the difficulty for some patients to interact with Wii's handheld controller and Balance Board. A pilot test of the game led to several useful conclusions, extracted through interviews and questionnaires, in terms of gameplay design, feasibility and safety. The authors conclude that Kinect seems both safe and feasible for PD patients, although the use of Kinect as a home-based rehabilitation solution needs further investigation.

Another multiplayer game also based on the Kinect sensor was developed by Hermann et al [41] in order to investigate if cooperation gamer cooperation can lead to improvements in terms of communication and coordination. Using hand movements, participants had to collect buckets of water in a flooded area to reveal an object underneath. The game was played in two different modes. In the first mode (loose cooperation) both players drained the area, while in the second mode (strong cooperation) only one would drain, while the second player's duty was to reveal the object. Gameplay was recorded to reveal information regarding the discussion and level of cooperation between the participating patients. Subsequent analysis showed that multiplayer games are feasible for this kind of target group and that asymmetric roles (strong cooperation) can motivate the communication between the participants and lead to a better game experience.

3. Solution Specification

3.1 Requirements

The present thesis presents loose, goal-oriented game scenarios for PD patients in a playful virtual environment as an alternative to following a predetermined rehabilitation program. To motivate patient participation it is important to create a pleasant and engaging environment which combines atomic exercises extracted from existing PD-specific rehabilitation programs and camouflages them within the context of a virtual environment in form of game actions.

As discussed in the previous section, patients following a long-term repetitive exercise schedule can easily get bored, lose interest and eventually drop out of a rehabilitation program (e.g., [28] and [32]). Exergames on the other hand engage patients into repeatedly executing simple or complex exercise patterns within a goal-oriented enjoyable context, with real-time feedback and rewards along the way using visual and auditory cues. To design for and develop exergames for PD patients, we first scoured the literature looking for proposed solution, as well as their strengths and weaknesses. As a result, we identified the following requirements and design principles that our exergames should adhere to.

Gaming system suitability – Safety first!

In their review Gillian et al [31] support that exergame solutions for PD patients should avoid raised platforms and handheld controllers such as those employed by the Nintendo Wii and Sony PlayStation Eye gaming systems. As lack of postural stability that characterizes the PD condition may lead to falls and severe injuries, Gillian et al suggest avoiding raised platforms (such as Balance board) as this piece of equipment may present additional trip risks. Evaluation trials conducted in two different studies [28], [40] also showed that PD patients wearing gloves or using handheld controllers with buttons encountered difficulties interacting with a virtual game environment. It follows that patient safety is maximally preserved with gaming systems which provide full body motion tracking without additional / external add-ons that users must interact with. Low cost solutions with minimal footprint that can easily be set up either in a clinical setting or a home environment would naturally be preferred. Still, all such systems must be evaluated in the terms of patient safety.

PD-specific game solutions

A number of studies [29] [28] [31] [39] [40] have showed that while off the shelf, low cost gaming systems can easily exist in both clinical and home environments, not all commercial exergames developed for those systems are suitable for the PD condition. For example, Mendes et al deduce that PD patient performance in commercial exergames depends on the motor and cognitive requirements of the game itself. To further elaborate on this observation, in three of the commercial Wii games that were tested PD participants failed to improve game performance compared to a control group of elderly healthy individuals. It is therefore advisable that exergames include specifically designed movements drawn from existing recommended PD training programs. In order to improve motor function (range of motion, balance, postural reflex, strength) the current thesis adheres to the Big training protocol [24] supporting that big and smooth movements increase the speed of upper and lower limbs. The conclusions of the preliminary trials of Assad et al [28] have also showed that PD patients may be frustrated by tremor that hinders performing slow and accurate arm movements.

Personalization and adaptability

Motor skills and more specifically the extent range for limbs vary significantly among PD patients, depending on several factors such as the course of the disease, medication and everyday physical activity. Accordingly, exergame movements must be scaled and calibrated based on the individual user's motor skill set [28]. In addition, exercise adaptations may vary from more challenging strict mappings looser ones, intended to allow for patient fatigue or frustration e.g., by repetitive failure of a task [32]. Therefore, not only captured movements must be adaptable, but also the difficulty of specific game tasks, such as game speed [40]. In the context of the current thesis, captured movements are automatically calibrated based on the measurements of the specific game player and the thresholds for detecting such movements are expressed as percentages of the measurements recorded. Although in the current version of our exergames thresholds cannot be automatically adapted during gameplay, they can be calibrated manually if necessary. In addition, game speed and the complexities of specific tasks can be fully adjustable to a patient.

Variety of game scenarios

In the introductory section of the present thesis it has been mentioned that PD is more prevalent to an aging population section (> 60 years old). Most of these elderly people have not experienced the computer revolution first hand and have little or no experience in gaming systems. To maintain interest for this significant population of PD patients, games must be simple in terms of design and scenario and must contain familiar concepts [28]. Indeed, Galna et al find that patients did not like the idea of complex scenario games like adventures or science fiction but were attracted by real-life events that used cartoon-style graphics. Also some patients show preference in outdoor activities scenarios [40]. Game simplicity was also preferred in the sense that game assets must be easily and immediately recognizable during the gameplay and their orientation inside the game scene more obvious so as not to distract from the goal of improving motor performance.

Exergames must also contain clear instructions and challenging goals to maintain motivation to improve motor performance. During the workshop carried out in [40] patients showed overwhelming preference to games that aren't too complex or too fast in reaction speed, while one mentioned that the game was too easy. A way to allow for a wide gamut of user preferences and abilities is to build multiple game levels. Less demanding low levels introduce simpler / slower tasks to serve familiarization with the game and avoid frustration, whereas more advance levels can pose higher demands like increased complexity, multitasking and faster reaction. Such is the approach of e.g., Assad et al whose games show different degrees of complexity [28], from really simple which focus solely in body movement to more difficult which require coordination and concentration.

One can thus infer that challenging game scenarios will stimulate both motor functionality (balance, strength, postural reflex) and cognitive (e.g., making the right decision in the given time). Following suggestions of physiotherapists that participated in their system design, Yu et al [32] employed the concepts of accuracy and timing to contain and reduce symptoms related to bradykinesia. Patients had to follow and execute instructive movements in a specific amount of time. Time threshold values were chosen carefully to motivate patients to react faster but were also sufficiently tolerant and flexible to accommodate individual player capabilities.

Visual and Auditory Feedback

Positive feedback, in the terms of visual and auditory effects, is very important for all games and even more so for PD-oriented exergames. Visual and auditory effects such as increasing the score, communicating the points gained or rewarding for the completion of a complex task serve at multiple levels: at a cognitive level they help keep the player informed and synchronized with the game status at a psychological level they reward, motivate and challenge. Paraskevopoulos et al [39] suggest that encouraging and motivating feedback increase game appeal and decrease the risk of patients abandoning the game and the benefits reaped from practicing altogether. Corroborating evidence from [28] emphasizes the importance of visual and auditory feedback as an effective mechanism to identify the state of progress during the game and be more effective in their following actions. In addition, patients that couldn't successfully complete tasks felt frustrated and unhappy, suggesting that players should be rewarded also on their effort and be encouraged to keep trying. It should rather follow that exergames developed for PD patients should avoid any kind of negative feedback [31].

Guidelines and Navigation

Especially home-based exergames must provide clear and sufficient game instructions to allow for unguided patient participation. Instructions can be in the form of videos or text. As an example, Assad et al created build-in instructions in the form of video tutorials for each game they developed. Instruction must also be included in any interaction of the patient with the system outside the gameplay itself. Interfaces such as quit or replay require specific visual instructions, guiding players into performing the correct action. Menu navigation must be simple and intuitive, helping players quickly understand what they have to do in order to further navigate into the game. Assad et al report that some participants inadvertently selected unwanted menu items due to the tremor in their upper limbs, suggesting that hand hovering over a specific region of the screen in order to select a menu item must be avoided. Navigation-related movements must be carefully selected in order to avoid accidental selections due to involuntary motion of the upper limbs. The current thesis opted for single arm raise, double arm raise and rotation of the shoulders was selected in order to navigate into the game.

3.2 Game scenario design

Taking into account the user requirements and design principles discussed in the previous section, we developed two exergames, a Balloon Goon game and a Skiing game targeting PD patients with mild symptoms (using Part III of UPDRS modified scale of MDS [13]), i.e., without severe postural instabilities and motor impairments. Patient candidates would have scored > 70% in Schwab and England ADL [1] scale. Our solution employs Microsoft Kinect, an off-the-shelf solution that can be easily installed in both clinical and home environments and provides motion capture at an affordable cost. The Kinect sensor provides datastreams from an RGB and an IR depth camera which can be combined using the MS Kinect SDK to yield the 3D skeleton of a human in front of it in real time. When the player's movements "match" gestures programmed in a particular game, a 3D cartoon avatar moves accordingly in the game environment. The allowed body moving patterns are based on published training programs (e.g., [42]). In addition, gestures were designed to improve postural stability and reflexes as well as increase overall mobility of upper & lower limbs. Game-embedded decision making is intended to improve the player's cognitive reaction. This agrees with [24] which advocates fast decision making combined with (depending on the level of the game) big but slow and fluent movements. Finally, the Unity game engine, a powerful development platform used to develop 2D & 3D games, was used to create proper gaming virtual environments with artefacts users can interact with. This is discussed in detail in Chapter 5.

Several video and audio effects were used in the virtual environment of the exergames, in order to provide the patient, with the necessary feedback about their progress during the gameplay. Such effects include a score board, successful target / obstacle hit sound effects as well as instructive text and applause. To achieve a game goal, patients have to complete a prescribed or dynamically generated sequence of repetitive tasks such as hitting moving objects in a specific window of opportunity, avoiding obstacles and collecting prizes. Progressive difficulty, in the form of different levels or by increasing the frequency of competitive tasks as the game progress, is also introduced to accommodate a wide gamut of patient capabilities and dexterities and also avoid frustration seeded on repetitive failure. Progressive difficulty is achieved by designing increasingly challenging levels or increasing the difficulty of a level with time.

Navigation through a collection of games can be achieved through an interactive main menu that uses images, descriptive text and interactive buttons. No external devices (mouse, keyboard, external controllers) are required to allow player interaction with the menu. Using the Kinect SDK library, we developed navigation gestures that are predominantly friendly to the PD community. We thus avoided altogether on-screen pointers that are manipulated by the movement of a patient's hand in order to eliminate the frustration arising from trying to keep a steady hand to select a menu option. For the same reason we also preferred gestures which cannot be caused by an inadvertent motion due tremor. Examples of PD-friendly gestures that are implemented in our gaming environment are "raise a hand to select" and "torso twist to navigate" to a different screen. A welcome bonus of this approach is that gestures are highly indistinguishable and independent as they require a big, fluid movement. A brief discussion of the exergames follows.

The Balloon Goon game

This score-based game calls for popping balloons which continuously and randomly drop along four vertical posts using upper and lower limb movements. Balloons falling along the inner posts pop using hand gestures (pushes/punches), while leg gestures (leg extensions / kicks) pop balloons falling along the outer posts. To encourage bilateral movement and score maximization, balloons falling along the left/right two pillars can only be popped by left/right hand & foot movements. Admissible hand and foot gestures trigger predefined animations of a virtual cartoon character. Three game levels of increasing difficulty have been implemented. As one progresses to higher levels, the speed and number of the balloons increases so that they drop more frequently. The third (bonus) level drops higher balloons of higher value but puts higher demands on reaction time as well as cognition, because level-specific bomb balloons must be avoided otherwise popping them costs points. During gameplay, but also upon game completion, performance is displayed to the patient.

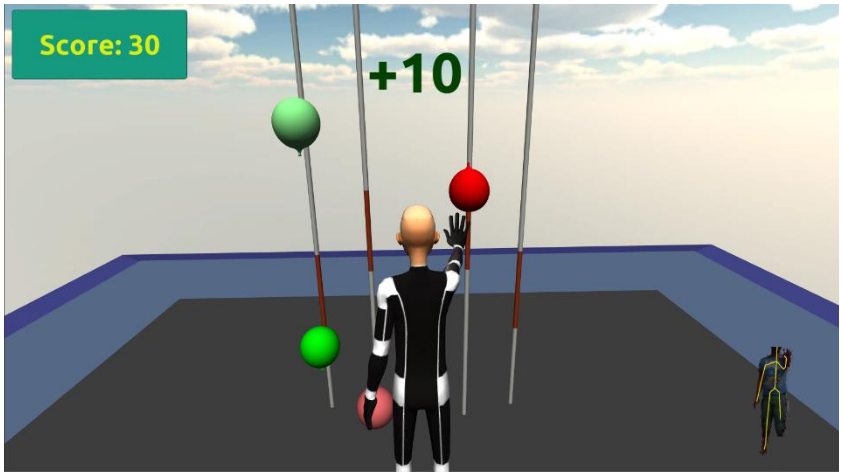


Figure 1: Balloon Goon Game: a frontal arm extension is required to pop a balloon falling along the right hand post

Cross country Ski with a twist

In this gesture-driven game, the player controls a 3D cartoon skier in a virtual snowy environment. The skier moves along either one of two parallel lanes collecting prizes (rings, stars) of different value along the way, while avoiding obstacles. As in real cross country skiing, the player pushes two imaginary ski poles backwards to make the avatar move forward. Leaning left/right causes the avatar to change lanes. Now, regarding prizes, rings are right above the ground and easily accessible, but stars are a lot higher and can only be collected by performing a jump over a platform. A successful jump requires a squat gesture (to pick up additional speed for a higher jump) at some distance before the avatar reaches the ramp. Sparse rocks in the terrain can be avoided by changing lanes, otherwise part of the avatar's life is lost and the game restarts. The game ends when the avatar crosses the finish line. During gameplay and also upon game completion the player is kept inform of his/her performance (rings / stars collected, lives available). Progressive difficulty is introduced even within the same level: rings are gradually succeeded by stars and rocks.

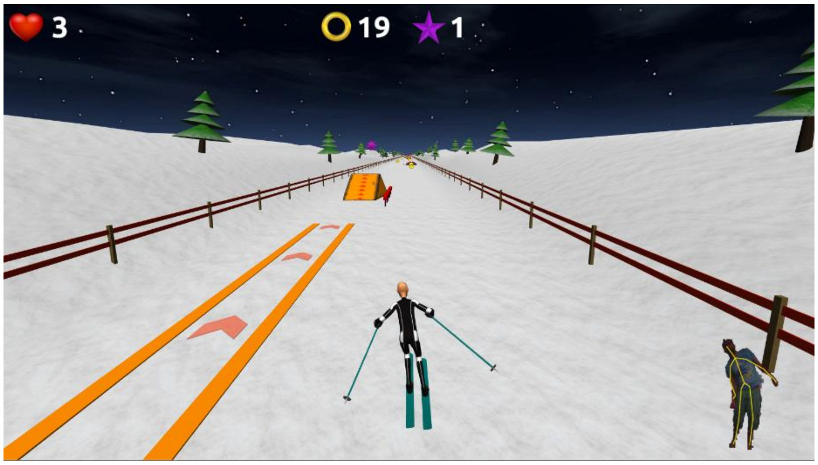


Figure 2*:* Cross country ski: lean left (insert) to switch to the left lane.

3.3 Technology Specification

Figure xx shows how key hardware & software technologies have been fused in our game solution. Real-time 2D (RGB camera) and depth (IR depth camera) streams captured by a Microsoft Kinect v1.0 sensor are processed by MS Kinect SDK v1.8 functions to produce a skeleton that is updated in every frame. Game logic has been coded in C# using the MonoDevelop editor of the Unity3D v4.6 game engine (which combines well with Kinect v1.0 and the MS Kinect SDK v1.8) – Unity can produce executables (the final game product) in a variety of platforms: Windows, MacOS, Linux, etc.

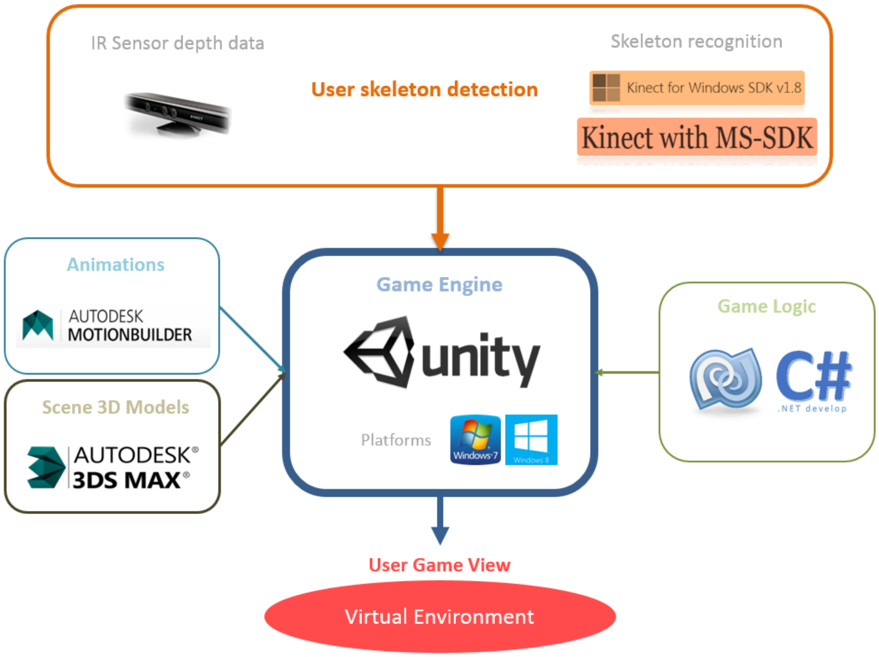


Figure 3*:* Exergame using Kinect sensor proposed development workflow

The virtual game environment has been created from scratch in Unity3D. Custom artefacts (ballons, pillars, rocks, stars, etc) have been designed in Autodesk 3dsMax 2016 (student license) and imported to Unity3D as game assets. Additional animation clips for the avatar have been designed in Autodesk MotionBuilder 2014 (student license) and were also imported into Unity – a successful game-specific player gesture will activate the corresponding clip (among other things e.g., move, destroy, etc game assets). Finally, the missing connection between the game engine and the captured skeleton information exposed by the Kinect SDK was implemented using MS-SDK middleware.

3.3.1 Microsoft Kinect

Kinect, initially designed as an extension for the Microsoft Xbox gaming platform, is a motion sensing input device released by Microsoft in November of 2010. Its development was based on another Microsoft project, code-named Project Natal that had begun in 2008 and was announced in 2009. Project Natal introduced a new 3D camera that was able to recognize body movements, plus face and voice recognition, based on video recognition technology developed by PrimeSense. What was different and perhaps revolutionary compared to other game consoles of the time, is that body movement recognition did not require an external controller. Kinect combines two cameras, a microphone array and an accelerometer and is accompanied by a software pipeline to process captured data.

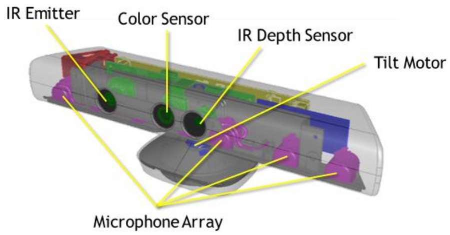


Figure 4*:* Hardware components, embedded on the Kinect device array

The sensor combines an RGB camera (1980x960 @ 12fps), which provides 2D image data, to an infrared (IR) emitter and an IR depth sensor (11-bit 640x480 pixelst @ 30fps) to capture depth. The IR emitter shoots thin infrared light beams (which show as white spots on objects when viewed with IR glasses, while the IR depth sensor captures the deformations of the impinging IR beams on various objects in space. Kinect SDK functions are used to calculate the distance between those objects and the sensor providing the missing depth coordinate. The depth data for each frame yields a real time 3D matrix representing coordinates from the surface of material objects which face the sensor. Direct sunlight and highly reflective surfaces may yield disturbed / biased 3D data in and should be avoided, if possible. The sensor also houses a microphone array used to capture sound in stereo (e.g., for voice recognition) and an accelerometer to determine the current orientation of the device (very useful in 3D reconstruction). The default view angle is 57^o horizontally and 43^o vertically with the latter able to shift vertically by +/- 27^o using a tilt motor.

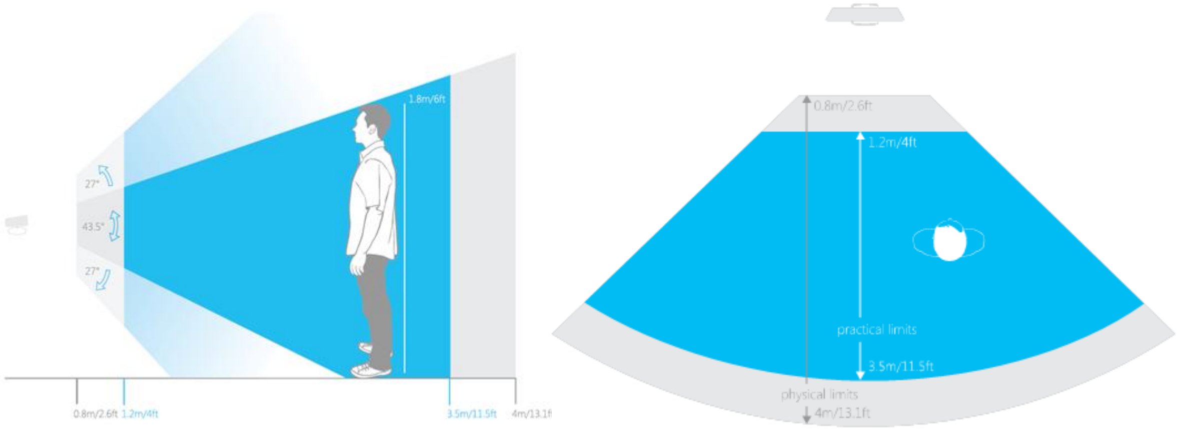


Figure 5: Angle view range of Kinect sensor, vertically and horizontally, by selecting the default settings

The API that exposes captured Kinect data to developers was not initially released to developers. In 2010 PrimeSense released open source drivers and APIs that developers could use in order to create software application for Windows platforms, until Microsoft released the first version of the Kinect SDK in June of 2011. A skeleton is defined by a set of joints, each of which represents an actual human joint. The benefit of the Kinect skeleton API, compared to previously released (e.g., PrimeSense) open-source APIs, is that the former does not require calibration to recognize a user. The final version of the Kinect

SDK for the Kinect v1.0 sensor is 1.8[1] and has been used in the current thesis. The device through the SDK software provided, using Natural User Interface (NUI) can manipulate the stream data coming from all the sensors. The Kinect SDK API enables recognition of up to six persons (associated skeletons are tagged as "position-only"), but provides skeleton data for only two of them (these skeletons are tagged as "tracked"). Tracked skeletons provide 3D coordinates for as many as twenty joints in standing (full body) mode and up to ten upper torso joints in seated mode. The Kinect sensor, or more accurately the associated software (Kinect SDK), automatically detects the user skeleton in standing mode by calculating the distance of the person from the background, while in seated mode the user detected, must move or lean in order to be detected.

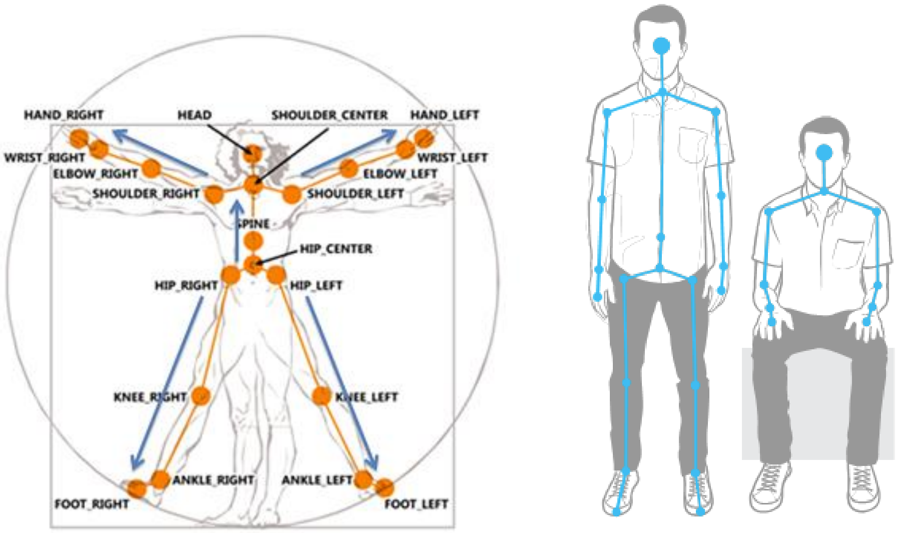


Figure 6*:* Left: Joints supported by the Kinect SDK v1.8. Right: Joints detected in seated vs standing mode.

The exergames developed in the present thesis utilize skeleton detection in standing mode, yielding real time skeleton joints for both upper (shoulders, elbows, wrists, arms and head) and lower body (spine, hips, knees, ankles and feet). For every frame where a player is visible and tracked, the joints of the captured skeleton position are stored in an array. For each joint stored information includes its 3D position (x, y, z coordinates) as well as a tag with two possible values: "tracked" for clearly visible joints, or "inferred" for joints that are not clearly visible but their position can be calculated from other tracked joints. Valid coordinate ranges are as follows: x from -2.2 to +2.2 meters, y from -1.6 to 1.6 and z from 0 to 4 meters. Positional changes of joints for a tracked skeleton contain information of macroscopic moves of a player (hand wave, swipe hand etc.). The process of recognizing such skeleton moving patterns is called gesture detection.

However, gesture detection is strongly affected by the fact that raw positional joint data can be jittery on an inter-frame basis. This is naturally intrinsic to the game platform (Kinect sensor + Kinect SDK) and is as much present for a healthy player as for a PD patient with severe tremor symptoms. In fact, the associated timescales are completely different: intrinsic jitter is locked to the stream capture frequency of 30fps, whereas PD episodic tremor is of the order of a few (3-7) Hz. Aware of the jitter problem, the

[1] In October of 2014 Microsoft released Kinect SDK v2.0 for the new Kinect for Windows v2

Kinect SDK provides a Holt Double Exponential Smoothing mechanism to stabilize a joint's position, as it does so with a smaller latency penalty compared to other algorithms. The following smoothing parameters with normalized values between 0.0 and 1.0 are provided by the Kinect SDK API.

- **Smoothing:** defines the amount of smoothing applied on raw skeleton data. Lower values yield less smoothed data (0 returns the raw data in their original state) and smaller latency than higher values. Latency affects the delay between a player move (as recorded by the cameras) and the corresponding tracked skeleton move.
- **Correction:** refines the way smoothing acts on raw data by modifying the depth of analysis performed on raw data. The correction applied affects the percentage of raw data to apply smoothing on (lower values correspond to higher correction).
- **Prediction:** affects the number of frames that will be predicted in the future.
- **JitterRadius:** defines a radius in meters within which jitter joints must be positioned - joints outside this radius are automatically relocated inside the radius.
- **MaxDeviationRadius:** defines the maximum radius in meters that a jitter joint may deviate from a raw joint during filtering. This parameter can be usefully combined to the Prediction parameter: Prediction values > 0.5 lead to overshooting when skeleton moves quickly. Using MaxDeviationRadius can restrain this effect.

The Kinect SDK allows dynamically changing the values of these parameters during gameplay to achieve desirable behavior. In fact, Microsoft suggests parameter sets for various setups. For example, gesture detection in games benefits from filtering only small jitters combined with medium smoothing with the intent to minimize latency. Accordingly, so values of {Smoothing: 0.5f, Correction: 0.5f, Prediction: 0.5f, JitterRadius: 0.05f, MaxDeviationRadius: 0.04f} are highly recommended.

3.3.2 Unity3D

Unity3D is a cross-platform game engine developed by Unity Technologies. It is a powerful platform that can be used to develop 2D&3D games and interactive application for PC, mobiles, consoles and websites. Most major components required to develop games can be obtained in the free edition. Since the initial release of the platform in 2005, five more major version have been released. Unity provides all necessary tools to create a complete game: an animation system, graphic solutions to create interactive user interfaces, audio controllers, lighting system, cameras, 2D / 3D physics engines, an API library solution to write scripts and many advance components to create a fully customized game. AssetStore, an online marketplace for Unity (also embedded inside the development environment), provides complete free and paid for packages such as terrains, artwork, complete games that can be imported in Unity projects. To maintain compatibility with multiple platforms, Unity uses Mono in the background, an open source development platform based on Microsoft's .NET framework of Microsoft, and includes among others a C# compiler, a runtime and various libraries.

The Unity interface combines many layout windows that are important into developing a fully custom experience. One of the major layout windows is the Scene view, which is used by the developers to set up the game environment. Scene View gives the capability to drag and drop game objects (characters, planes, buildings, etc.) in the environment, as well as keys to position, rotate and scale them as needed. In addition, Unity contains several components that can be attached to game objects, such as animator controllers that choose the right animation to played for the specific game object during gameplay, components that can give the object physical behavior in order to be affected by gravity or collisions with other objects in the environment, even components that transforms the game object's material or texture. Scripting components can be also attached to the game objects, written by the developers and affect the game object's functionality during gameplay.

Game objects can be initial positioned in the game environment and be visible when the game starts or can be stored and loaded dynamically during gameplay (prefabs). Prefabs enable developers to design dynamically created environments that can dynamically adapt during gameplay. While Unity provides tools for creating primitive objects and setting them inside the game environment, creating sophisticated game scenes with complex graphics requires use of external graphic programs. Accordingly, Unity supports several formats of graphic objects that can be imported, such as .fbx or .obj. The platform also exposes capabilities to create simple animations used to animate game objects or user interface interactive components, but it inherently lacks tools to easily create more complex animations (such as animating a character's avatar).

3.3.3 Middleware: Kinect with MS-SD

The Kinect SDK libraries are based on Microsoft's .NET 4 framework. However, Unity's Mono framework is based on an older version of the .Net framework and is not compatible the .NET 4 framework. It is for this reason that the Kinect SDK library cannot be directly accessed inside the Unity development environment. To alleviate this problem, developers began to develop custom middleware solutions that expose Kinect SDK functionality for development inside Unity. A well respected middleware solution that can be imported in a Unity project using the asset store is Kinect with MS-SDK developed by Rumen Filkov. The package uses a C# script called KinectWrapper to expose Kinect SDK library (Kinect10.dll) functionality inside the development environment of Unity3D. A second C# script called KinectManager includes functions to read data from the Kinect sensor to build a skeleton. The package is accompanied by several examples of use to create a complete game scene, control an avatar and detect user gestures.

3.3.4 Autodesk 3ds Max

3ds Max is a powerful modeling, animation and rendering program developed and produced by the Autodesk Media and Entertainment group. Offering a plethora of capabilities, only those pertinent to the current thesis will be currently presented. The program comes in two versions: 3ds Max which caters to entertainment and gaming professionals and 3ds Max Design, aimed at visualization specialists, architects and engineers. Game designers are provided with several tools to create 3D objects that can be modified or combined into 3D models through polygon modelling, as well as a toolbox to create and apply materials and textures to those models. In addition, 3ds Max provides tools to create terrains and natural scenery to set up a 3D game space. It can also be used to create skinned and rigged character avatars, design a character skeleton with customized joints and create a skin mesh to be applied to its body. It is worth mentioning that 3D model creation of a game object inside 3ds Max can be achieved in several different ways, and it is up to the designer to select the best approach in each case. All models created in 3ds Max can be exported in a variety of formats to be imported by different software, such as Unity3D.

3.3.5 Autodesk MotionBuilder

Although 3ds Max can be used to develop animations for a rigged skeleton character, in 2013 Autodesk produced a professional 3D character animation software, named MotionBuilder. Developers can use the MotionBuilder to create keyframe animations for characters of their game. Character models can be imported in MotionBuilder and can be animated using a wide variety of tools embedded in the software.

In order to animate a character model inside the software, it is necessary to complete a process called, mapping and characterization. Mapping defines the structure of the character model by associating the bones found on the character skeleton to the mapping list recognized by the software. MotionBuilder also provides a visual display indication, showing when this procedure is completed successfully. Characterization calls for choosing a biped or quadruped character for the bone mapping list previously created. Upon completion, the current pose of the model is stored as the default pose for all animations to be created. As soon as the character model is mapped and characterized, a control rig can be created, which is an animation tool that helps developers control the position and rotation of any joint found on the character. Another major advantage of the software is the capability to create animation poses in order to reuse them in several frames of the animation timeline or easily create mirror movements.

In recent versions of the software, Autodesk introduced interoperability workflow to streamline the development process. More specifically, while developing a character model in 3dsMax, it can be quickly send to MotionBuilder to be animated. Upon completing the animating task, the character can be sent back to 3ds Max. Both modeling suites are commercial and available on a paid license, although Autodesk offered free student licenses for a 3-year period.

3.4 Software architecture

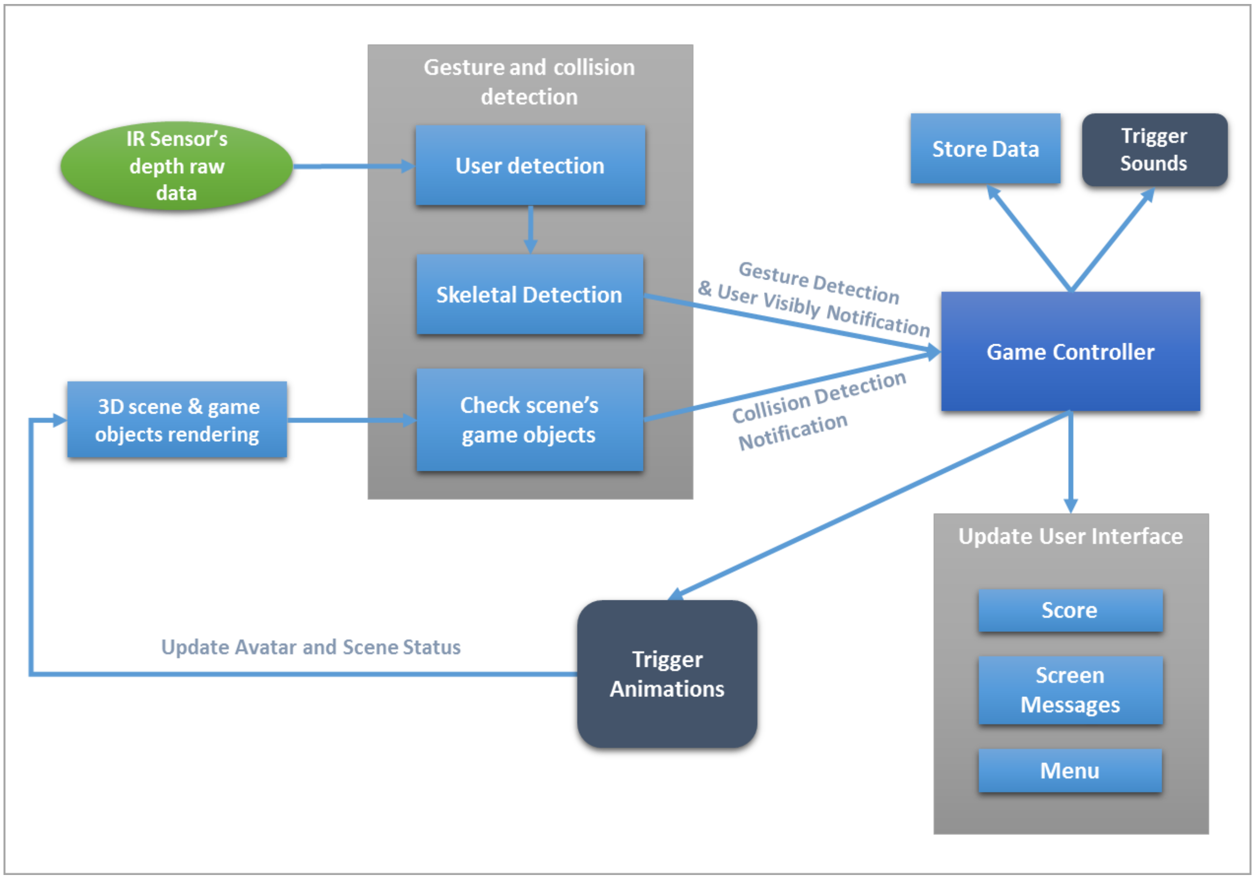

Gesture and collision detection
IR Sensor's depth raw data
User detection
Skeletal Detection
Check scene's game objects
3D scene & game objects rendering
Gesture Detection & User Visibly Notification
Collision Detection Notification
Store Data
Trigger Sounds
Game Controller
Update User Interface
Score
Screen Messages
Menu
Trigger Animations
Update Avatar and Scene Status

Figure 7: Software architecture proposed for both exergames developed

A high-level software architecture for both exergames developed in the present thesis appears in **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**. Both games begin by instantiating (a) the 3D scene environment, (b) the animated 3D avatar controlled by the patient's gestures and (c) pertinent game objects (balloons, ramps, obstacles etc.) the avatar can interact with. Using Kinect SDK library functions that are exposed by the Kinect for MS-SDK wrapper, the stream of Kinect provided data is continuously analyzed in real time to identify user gestures and collisions with game artefacts. Specific functions provide skeletal data for a detected (tracked) patient, followed by the gesture detection stage to identify macroscopic skeletal motion patterns that match predefined gestures such as kick, push and lean, in order to detect a gesture. At the same stage, game logic checks for interactions of game objects with the avatar (collision detection stage) such touching a balloon or hitting a rock.

Whether a patient gesture or a collision event has been detected, a notification event is sent to the game controller, a central game component responsible for communicating with all others. When the game controller receives a gesture detection notification, it selects the appropriate animation clip that corresponds to that gesture (reposition the avatar, extend a leg or a foot, etc), triggers the animation and records the gesture performed. These avatar animations may lead to collision with existing game objects. If the game controller receives a collision event with a game object, it responds accordingly. Collision events may update user interface elements (e.g., change the score, display an informative), trigger sound effects (e.g., sound of a popping balloon), even cascade to new avatar animations (fall when after hitting a rock).

However, it is not necessary that all gestures move the avatar. Indeed, some gestures simply trigger menu actions. For example, raising both hands pauses the game and leads to the appearance of the pause menu. Menu scenes also allow interaction with the patient through gestures. In addition, the game controller is continuously notified about the gamer's visibility, so that if the player is lost (e.g., steps out of the Kinect sensor's field of view), the game automatically pauses and the appropriate menu screen is displayed.

4. Posture and Gesture detection

Natural User Interface (NUI) interactions at the core of Kinect's SDK API enable user identification and motion capture for users within the sensor' field of view (FOV) without the need to interact with handheld controllers. This is achieved by providing access to color and depth data transmitted by its sensors as a data stream. The Kinect SDK can identify up to six users moving inside the FOV of its infrared (IR) camera and extract and track twenty-joint skeletons for up to two of these users. For each tracked user, skeletal tracking provides for each frame the 3D position of each joint along with a tag describing whether that joint is directly visible by the sensor or not. Our game logic processes this real time data stream to identify skeletal motion patterns that fit predetermined gestures, a process called gesture detection. Gesture detection is as much an integral process of a game as it is necessary to navigate a menu system.

The Kinect SDK does not provide an out-of-the-box software solution for gesture detection, so one has to be developed. In his middleware software solution, used to expose Kinect SDK API functionality inside Unity3D, Filkov also proposes a C# software implementation for gesture detection. That implementation was adapted and modified to create the gestures required by our game solution. Filkov's approach is a multi-state, time-dependent gesture detection solution and works as follows. A single gesture is broken into a set of states which are designed into a Finite State Machine (FSM). Each such state can be thought of as a snapshot of the actual moving relevant skeleton part and is accompanied by specific joint-level deviation limits that must be respected (in the sense of modeled-detected) to proceed to the next state until the full gesture is completed. This means that if a gesture passes the N first checkpoints (states) but deviates significantly on state N+1, it will be cancelled. In addition, not completing a gesture within specific time limits will cause its cancellation. That makes sense for PD patients as purposeful and large, but not lazy, movements are preferred. It becomes obvious, then that successfully executing a gesture requires both adherence to form and a timely completion.

Specific gestures were designed and developed to be associated to the range of movements need to be performed by PD patients during gameplay. For each gesture, joints returned by skeletal tracking were carefully selected to participate in the gesture detection process to ensure that movements performed by players are correctly identified and implemented. In accord with Filkov's implementation, our gesture detection algorithm (Figure 8) comprises two states, gesture-begin and gesture-end, where the latter state also checks whether for gesture completion in the prescribed (for that gesture) time.

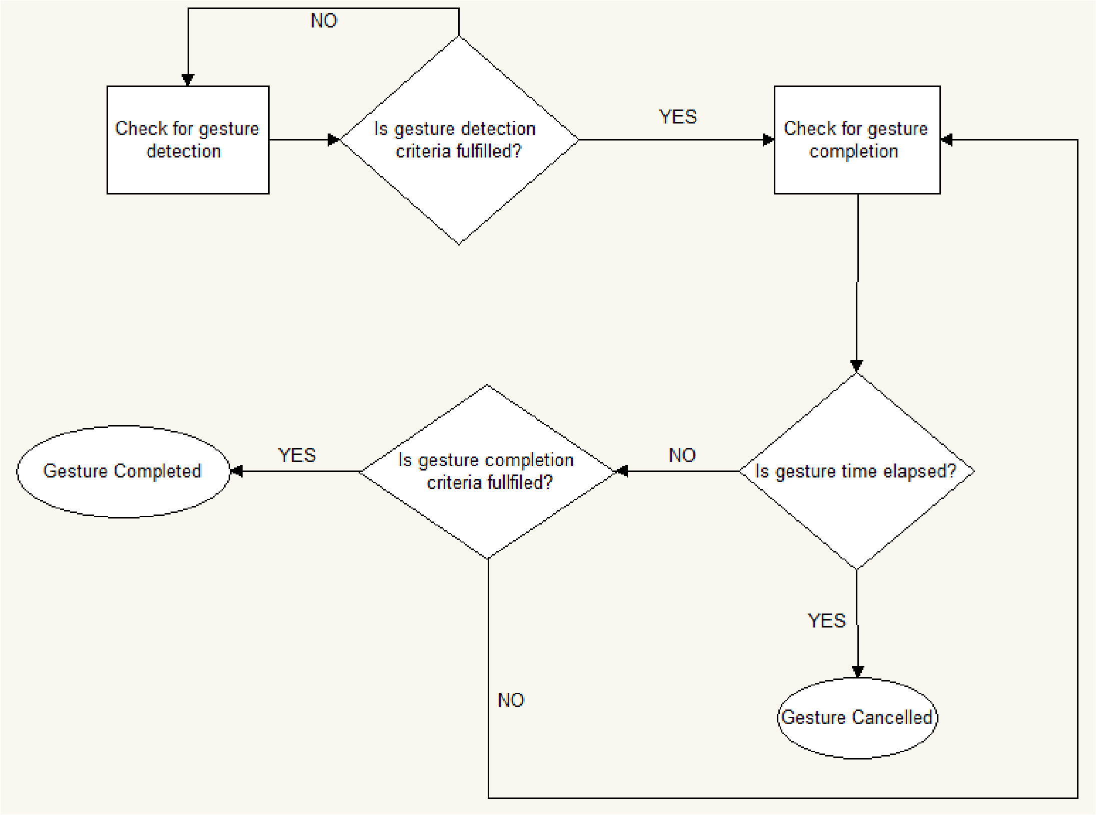
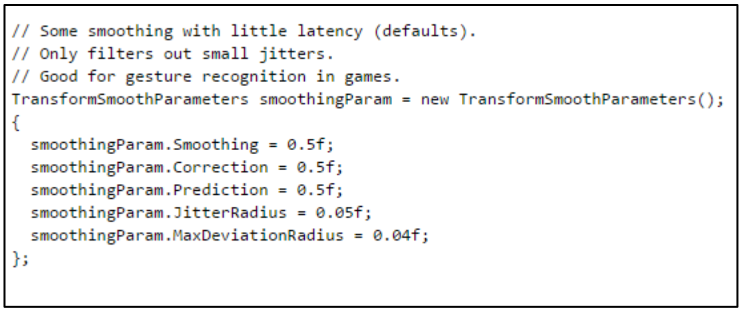

NO
Check for gesture detection
Is gesture detection criteria fulfilled?
YES
Check for gesture completion
Gesture Completed
YES
Is gesture completion criteria fullfiled?
NO
Is gesture time elapsed?
YES
NO
Gesture Cancelled

Figure 8: Gesture detection algorithm

In every game loop, our gesture detection implementation tests for possible detection or completion of every gesture. If positional deviations of participating joints are sufficiently small in one game loop, gesture completion in a timely manner (i.e. detection) is tested in the next game loop. If the criteria are met both at gesture begin and gesture end, then the gesture is considered detected, otherwise it is cancelled and the test is run anew in the next game loop.

As gesture detection appears highly dependent on the 3D positions of the participating tracked joints, jittery positional data can affect the gesture detection (section …). Accordingly, time-base smoothing is performed on raw positional data, manipulated and fine-tuned by the following Kinect SDK parameters: smoothing, correction, prediction, JitterRadius and MaxDeviationRadius parameters. The parameter values that appear in Figure 9 only slightly deviate from those suggested by Microsoft and are selected so as to alleviate problems due to patient tremor. These will be further fine-tuned pending our planned test runs with patients.



```
// Some smoothing with little latency (defaults).
// Only filters out small jitters.
// Good for gesture recognition in games.
TransformSmoothParameters smoothingParam = new TransformSmoothParameters();
{
  smoothingParam.Smoothing = 0.5f;
  smoothingParam.Correction = 0.5f;
  smoothingParam.Prediction = 0.5f;
  smoothingParam.JitterRadius = 0.05f;
  smoothingParam.MaxDeviationRadius = 0.04f;
};
```

Figure 9: Joints filtering proposed by Microsoft for gesture detection.

4.1 Personalization and T-Pose detection

As mentioned in the previous section, gesture detection relies on the accurate tracking of participating joints. Several distances between joints are combined and compared to threshold values to determine whether a gesture is in a begin- or an end- state. However, as each tracked joint's x, y, z coordinates place that joint in 3D space, the distance between two successive joints reflects the length of the bone between those joints which varies between humans. For example, the distance between the foot and hip joints is generally bigger for tall users. Accordingly, threshold values for allowed deviations must be customized to the player.

To enable absolute measurements we use the T-pose (Figure 10), which identifies the position of the 20 joints that are tracked by Kinect. T-pose is actually implemented as a gesture where a user stands in front of the Kinect sensor with both hands extended to the sides and palms at the height of the shoulders. Although one may get by without T-pose by using nominal values, the customization afforded by exploiting the T-pose allows for more realistic and accurate gesture detection deviation limits.

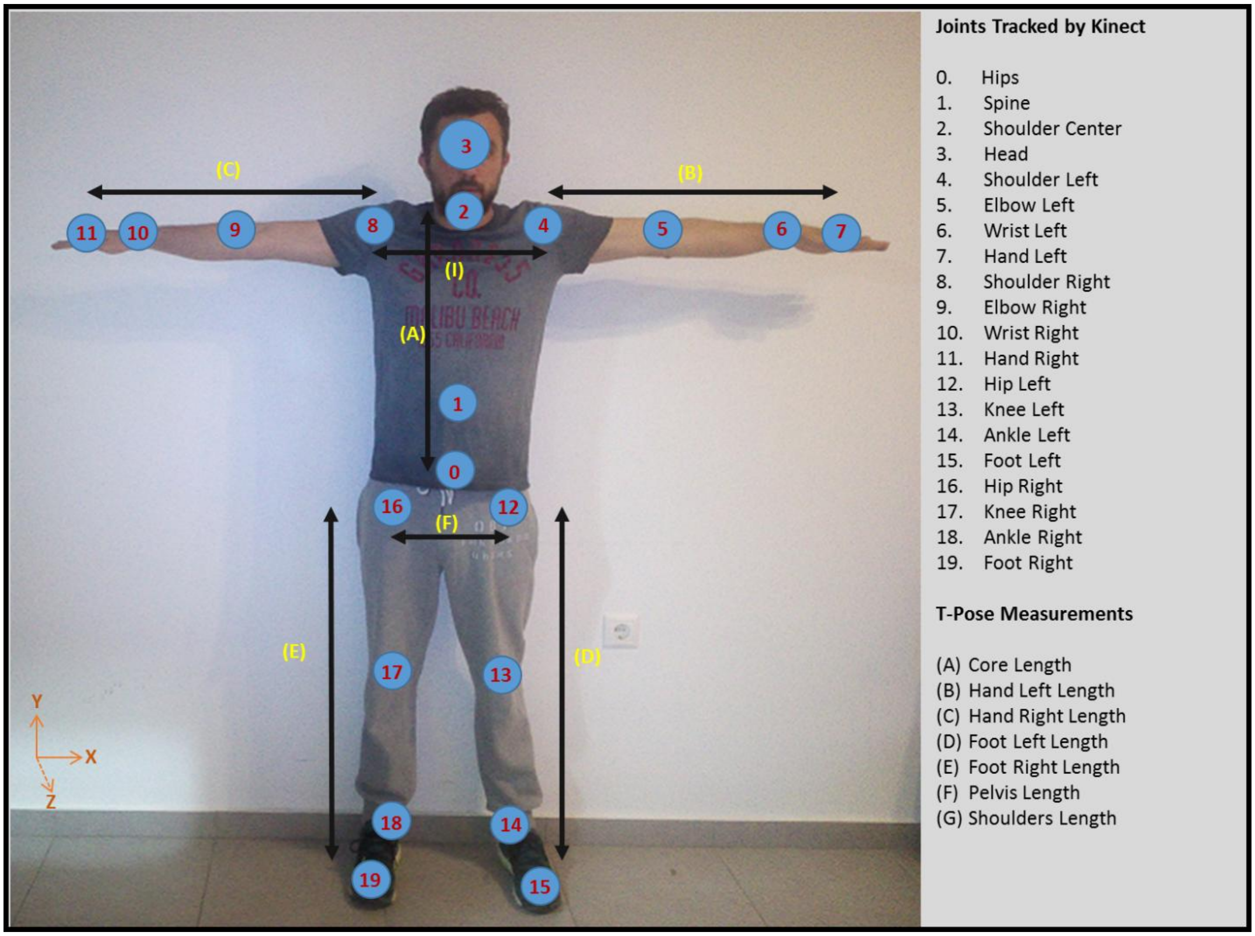


Figure 10: The T-pose gesture identifies the 20 joints tracked by Kinect and their derivative body measurements.

(A) Core Length: the distance between the Hip (0) joint and the Shoulder Center (2) joint along the y axis.

(B) Hand Left Length: the left arm length expressed as the distance between the Hand Left (7) joint and the Shoulder Left (4) joint along the x axis.

(C) Hand Right Length: Represent the right hand length and it is defined as the positional deviation between the Hand Right (11) joint position and the Shoulder Right joint (8) position in x axis.

(D) Foot Left Length: Represent the left foot length and it is defined as the positional deviation between the Foot Left (15) joint position and the Hip Left joint (12) position in y axis.

(E) Foot Right Length: Represent the right foot length and it is defined as the positional deviation between the Foot Right (19) joint position and the Hip Right joint (16) position in y axis.

(F) Pelvis Length: It is defined as the positional deviation between the Left Hip joint (12) position and the Right Hip joint (16) position in x axis.

(G) Shoulders Length: It is defined as the positional deviation between the Left Shoulder joint (4) position and the Right Shoulder joint (8) position in x axis.

T-pose measurements are used in the gesture detection process. All conditions used to identify each gesture's detection or completion are expressed using distances between the tracked joints that are compared to parts of the body measured during the T-pose detection. In order to track the T-pose of a user, distances are calculated between the Shoulder, Elbow and Hand joints in both y, z axis. The T-Pose gesture is shown in Figure 11. The algorithm to detect a T-pose, demands that the y-coordinates of the left and right Hand, Elbow and Shoulder joints are sufficiently close (e.g. a deviation of 10cm from the mean is allowed). An analogous statement must be true for the z coordinate for these joints. In the specific gesture the detection and the completion stage uses the same calculations. By design, the specific gesture is independent of the actual player measurements because the condition to be met does not require absolute comparison between body segments.

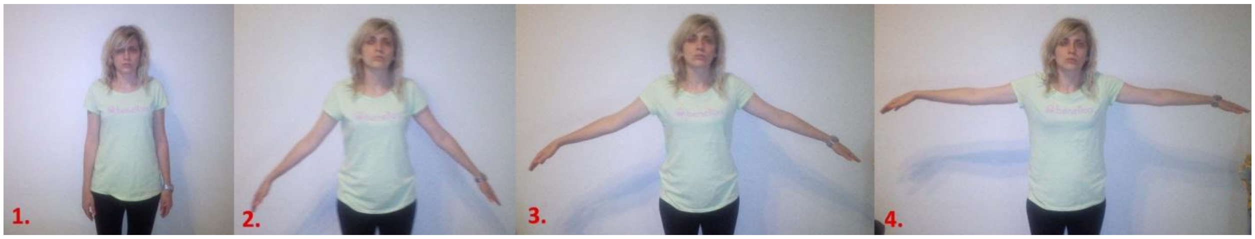
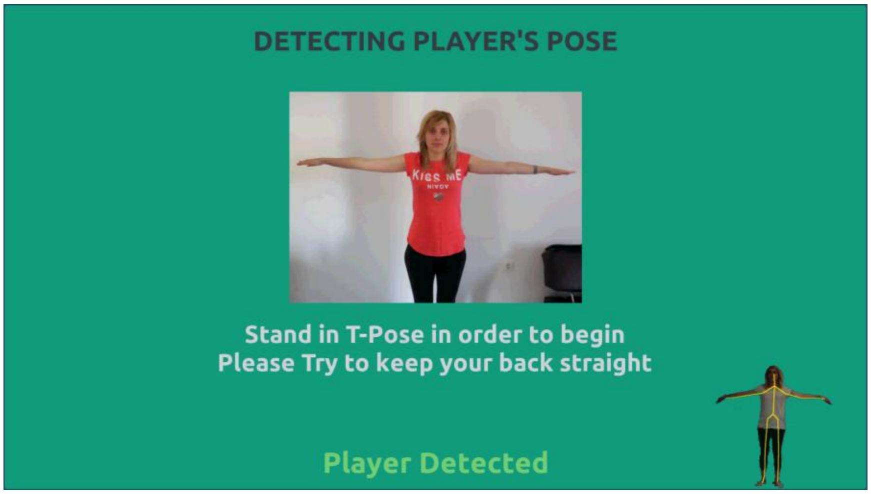


Figure 11: Various snapshots on the way to a T-pose gesture. The gesture is identified in insert icon #4.

T-Pose detection is executed before the start of any game, and is performed even before the main navigation menu appears. Figure xx presents the T-pose gesture detection inside our game environment.



DETECTING PLAYER'S POSE

Stand in T-Pose in order to begin
Please Try to keep your back straight

Player Detected

Figure 12: T-pose detection inside the game environment.

4.2 Pairing Gestures to Animations

Gestures used in games to detect specific player movements are based on variations of exercises targeting PD patients with mild symptoms. Several Parkinson's disease associations, clinic facilities, physiotherapy centers and institutes around the world provide guidelines and training programs for people suffering from PD condition [18], [43], [44], [45], [46], [47]. Exercises found in those training programs are proposed in order to improve balance, mobility, strength and flexibility (stretching exercises). Gestures that designed and developed are in some cases variations (or combinations) of the proposed exercises found in training programs in order to be meaningful in the context of the gameplay. In addition the gestures are designed taking into account the "Big protocol" [24], which advocates big, smooth and purposeful movements in order to increase speed.

4.2.1 Balloon Goon Game Gestures and Animations

Single Hand Raise & Extend gesture

The specific gesture is detected when the user raises one hand at a time and then stretches it forward. In order to detect the gesture we use deviations of the Hand and Shoulder joint position compared with Hand Length found during T-Pose gesture. The gesture is used in order to trigger the animation, of the 3D character inside the game environment, responsible for breaking balloons using either left or right hand. It's simulating the movement, similar like the user is trying to reach a balloon in front of him and grasp it in order to break it. The specific gesture helps the PD patients mobilizing the joints in order to decrease symptom effects of rigidity and in addition it helps stretching both hands. Gesture is based on reaching and grasping therapy exercises [47]. Single raise hand extend gesture is also used in order to select options found on menu screens in both games. For example by rising and extend left or right hand user can push virtual buttons, used for example in order to quit game or proceed to next level.

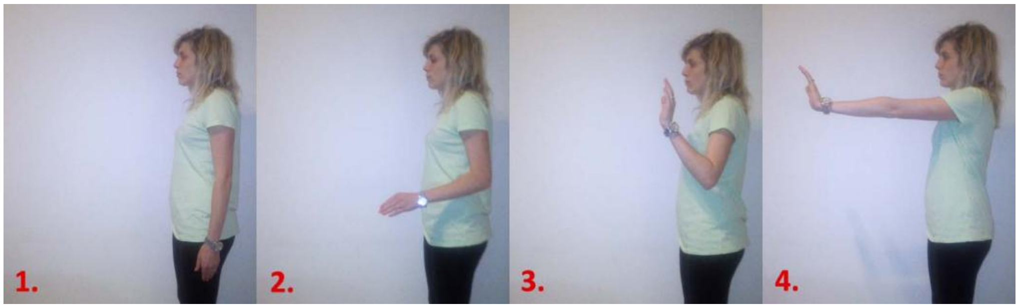


Figure 13: Single Hand Raise & Extend gesture is shown executed using the right hand.

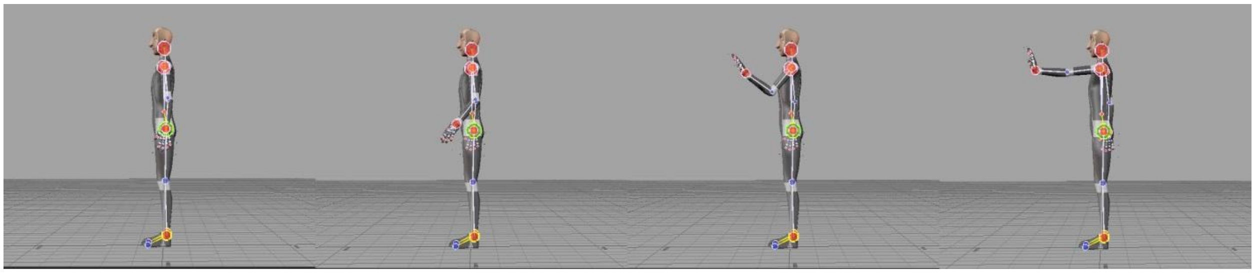


Figure 14: Animation key frames during the development process of Single Hand Raise & Extend gesture.

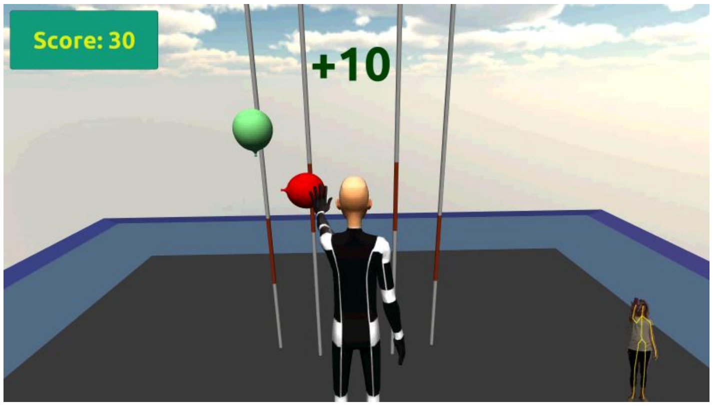


Figure 15: Single hand raise and extent gesture avatar animation during gameplay, by performing the gesture using the left hand.

Single Leg Raise & Extend gesture

The presented gesture is detected if user raises one foot up straight by bending his knee, followed by stretching the leg while the weight is on the other leg. Positional deviations between the Foot and Hip joint are calculated and compared with Foot Length found during T-Pose, in order to detect the specific gesture. Foot kick animations are triggered by detecting this gesture while playing the Balloon Goon game, and are responsible for breaking balloons by either the left or right leg. The gesture is a combination of exercises use lift foot in marching position while holding it for a few seconds, in order to improve patients balance [43][46] and straighten leg exercises [45] used for strengthen foot muscles. It is proposed, for the patients with balance problems to use side supporters when executing this gesture.

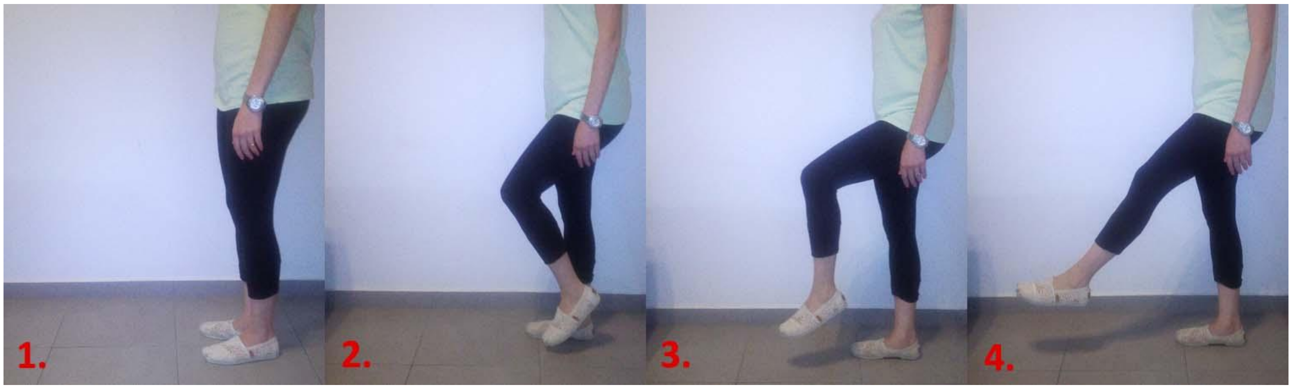


Figure 16: Single Leg Raise & Extend gesture is shown executed using the right foot.

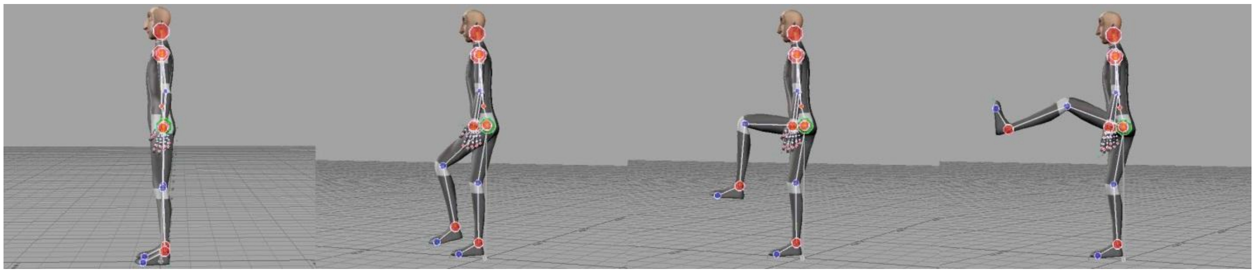


Figure 17: Animation key frames during the development process of Single Leg Raise & Extend gesture.

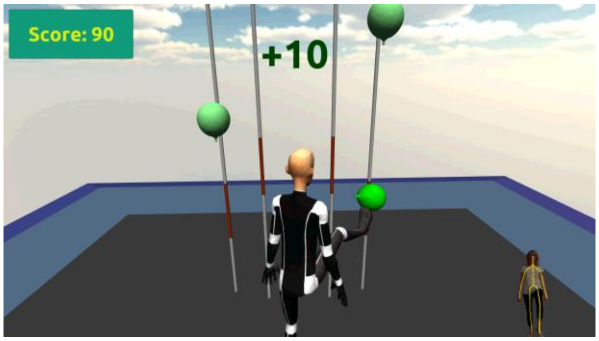


Figure 18: Single Leg Raise & Extent gesture avatar animation during gameplay, by performing the gesture using the right foot.

4.2.2 Skiing Game Gestures and Animations

Raise & Lower Both Hands gesture

The presented gesture is detected if user raises both hands in front of him, followed by lowering them in the initial position. Positional deviations between both the left/right hands and left/right shoulder joints are calculated and compared with Left Hand Length and Right Hand Length measurements. Positional threshold for both hands must be respected in the same time in order the gesture to be detected. A start skiing animation is triggered that is responsible for making the 3D character start moving inside the game environment. Gesture is based on exercises whose purpose is to strengthen the muscle of the hands of PD patients. Usually this exercise is performed with patients grabbing a stick.

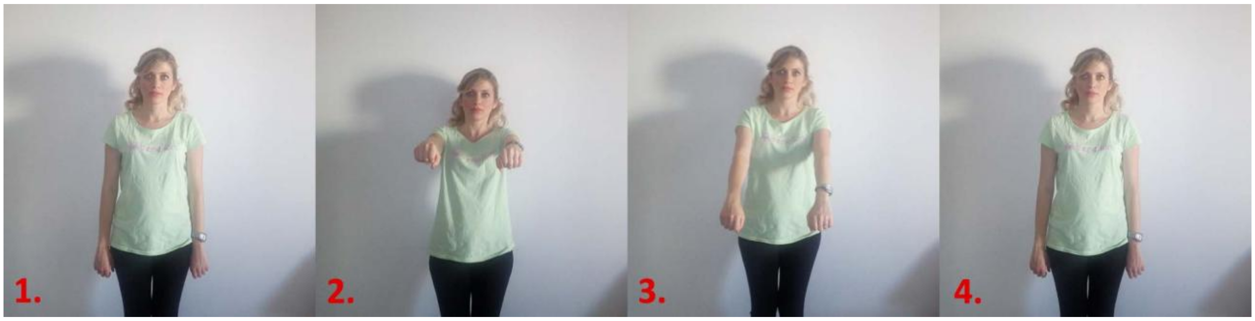


Figure 19: Raise & Lower Both Hands gesture is shown.

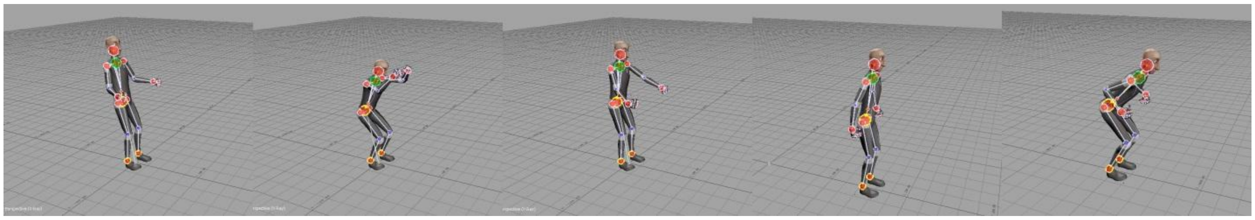


Figure 20: Animation key frames during the development process of Raise & Lower Both Hands gesture.

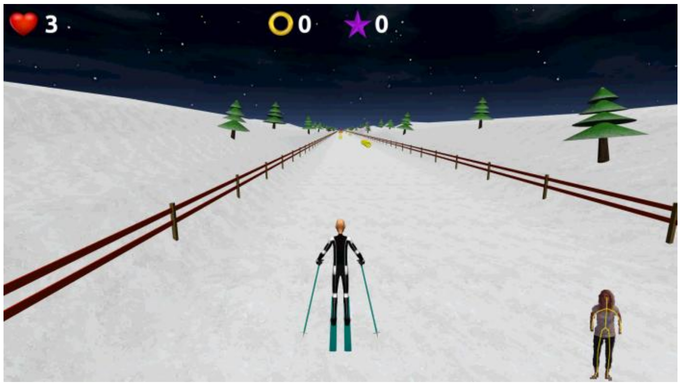


Figure 21: Start skiing animation by detecting Raise & Lower Both Hands gesture.

Side Lean Core gesture

The specific gesture is detected when the lean the core left or right. In order to detect the gesture we use deviations between the positions of multiple joints, such as the Hip Center, Shoulder Center and both Shoulder joints, the Core Length and the Shoulder length measured in T-pose gesture are used in order to compose the conditional barriers. The gesture is used in order to turn the 3D character inside the game environment, left or right by applying the correct animation clip. The gesture is similar like stretch exercise used in various training programs for improving core flexibility. The exercise is usually used by raising, the opposite hand from the side that patients lean, above the head [45]. The gesture detection process also secures that the specific gesture is performed correctly, with the patient lean to one side but keeping the core straight. Except from the stretching benefit of this exercise, if it's like in this case, executed in standing position it may be beneficiary for the balance of the patients, as they transfer their weight from one side to another.

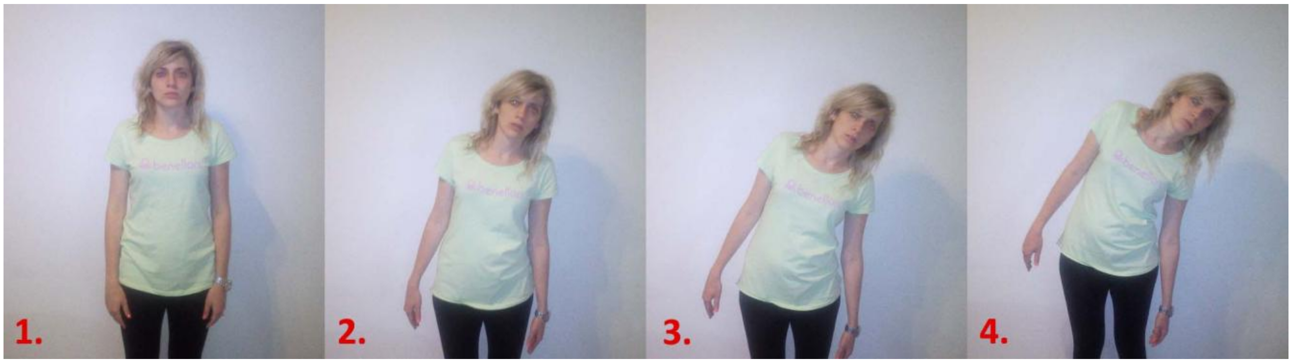


Figure 22: Side Lean Core gesture is presented by strerching to the left side.

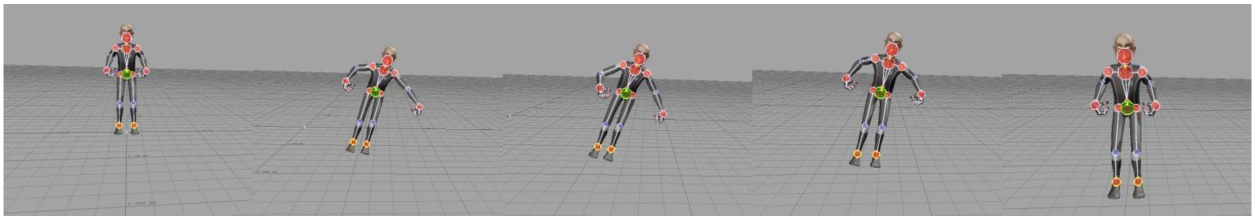


Figure 23: Animation key frames during the development process of lean left gesture, that results or making the 3D character turn left.

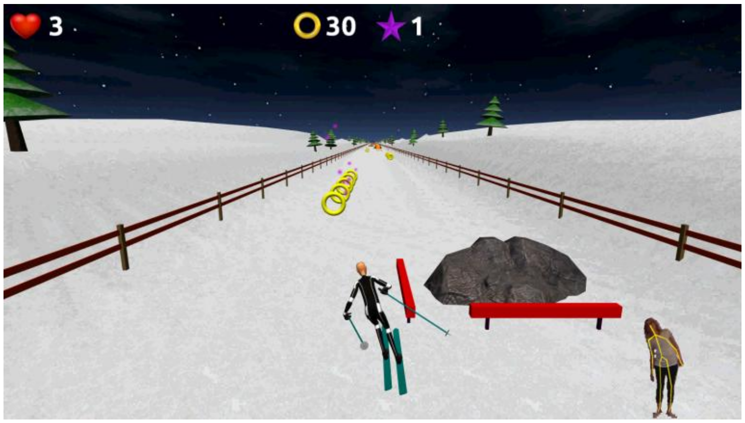


Figure 24: Turn left during Lean To One Side gesture detection inside game environment, leaning the core to the left.

Squat gesture

The presented gesture is detected if perform a squat exercise. The length of both legs measured in T-pose are used in ordered to be compared with position deviations between the left/right foot and left/right hip joint respectively, in order to identify the squat gesture. Squat exercise is used in order to improve the patient's balance and to strengthen the leg muscles [43]. The gesture is used in game in order the 3D character perform a squat and gain speed during jumping from a raised platform. This exercise may appear risk of falling back, so it is recommended for patient with higher instability to use a chair behind them.

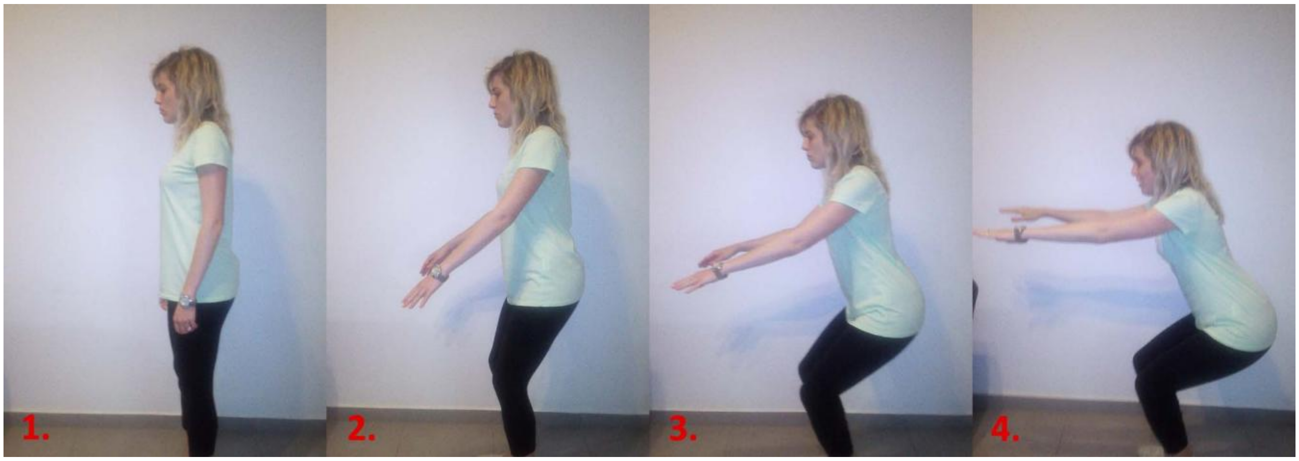


Figure 25: Squat gesture detection

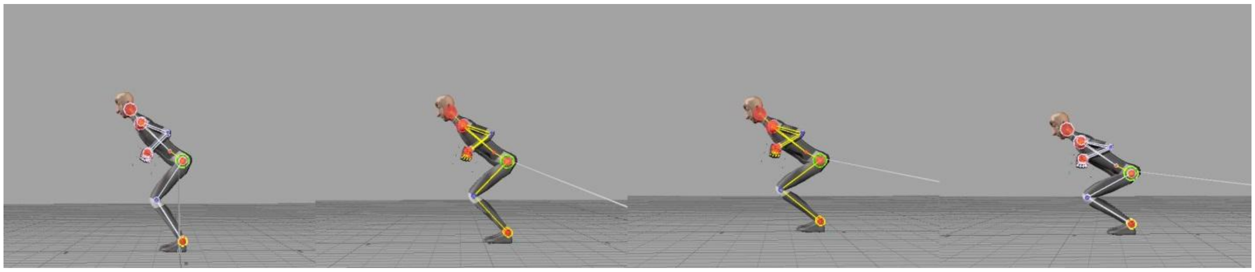


Figure 26: Animation key frames during the development process of squat gesture.

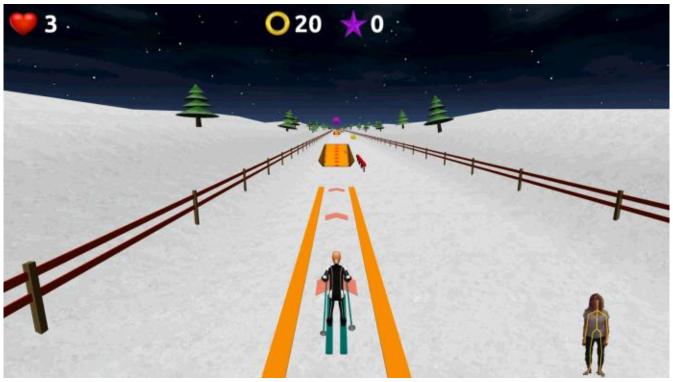


Figure 27: Animation during gameplay while squat gesture is detected. 3D character is lowering gaining more speed.

4.2.3 Menu Gestures

As it is mentioned before not all gestures are driving animation clips during the gameplay, in order to change the posture status of the 3D character. Several gestures are used in order to navigate through the game menu, or select options found on menu screens. We used gesture detection in such cases in order to avoid alternative input methods that may trouble the PD patients controlling the games. Using gestures for such events PD patients are feasible to play the games without any external help.

Rotate Shoulder gesture

Rotate shoulder gesture is used in order the users can select between the different games during the menu navigation. Gesture is accomplished when a user is steering the shoulder, moving their upper body in a left or right direction. In order to detect the gesture positional deviations between the two shoulders are performed in order to identify the correct movement. Rotating the shoulders is used in order to enable trunk rotations and is used as a stretching exercise in order to achieve higher flexibility and better postural control of the core. Gaining a better postural control helps the patients to perform also other exercises [44] [46].

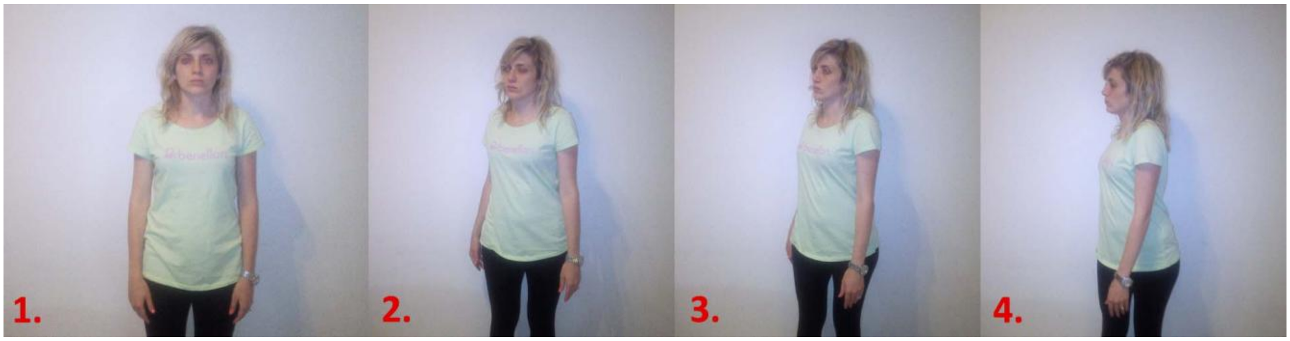


Figure 28: Rotate Shoulder gesture by steering right.

Raise Both Hands gesture

The specific gesture is used in order to start a specific game or pause the game during gameplay. The users must raise both hands in the same time in a certain position above the shoulders. It is usually used as a stretching exercise [18], providing improvement in flexibility of both hands and core. In order to be detected the translation of hands is monitored.

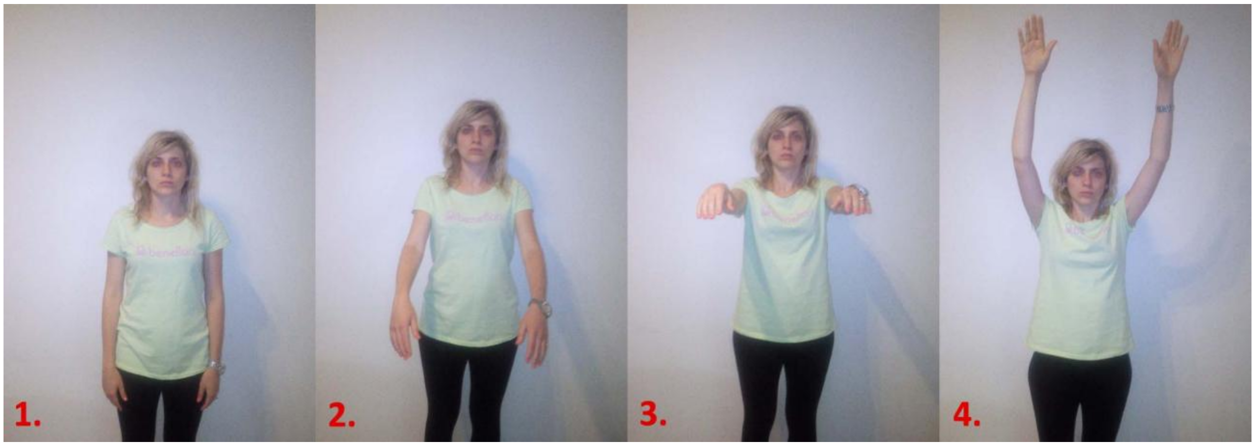


Figure 29: Raise Both Hands gesture

5. Game design

5.1 Game Logic

Balloon Goon Game

Users in Balloon Goon game are motivated into popping a predefined number of balloons, falling along four posts placed in the 3D environment, using hand and foot gestures as described in the previous section. The specific game is composed by three levels with progressive difficulty, which is defined by the speed of the falling balloon and the distance between them. As the user proceeds to the next level, the speed of the falling balloon increases and the distance between then is smaller, forcing the user to decide and react faster. Balloon Goon game is trying to improve motor, balance and flexibility status of the users but also the cognitive one, as they are called to make the right decision in the right time in order to achieve the goals of the game. Users have to break the balloon using the correct hand or foot gesture in order to pop the balloons in a specific time frame. Figure xx, presents a general diagram of the Balloon Goon game logic.

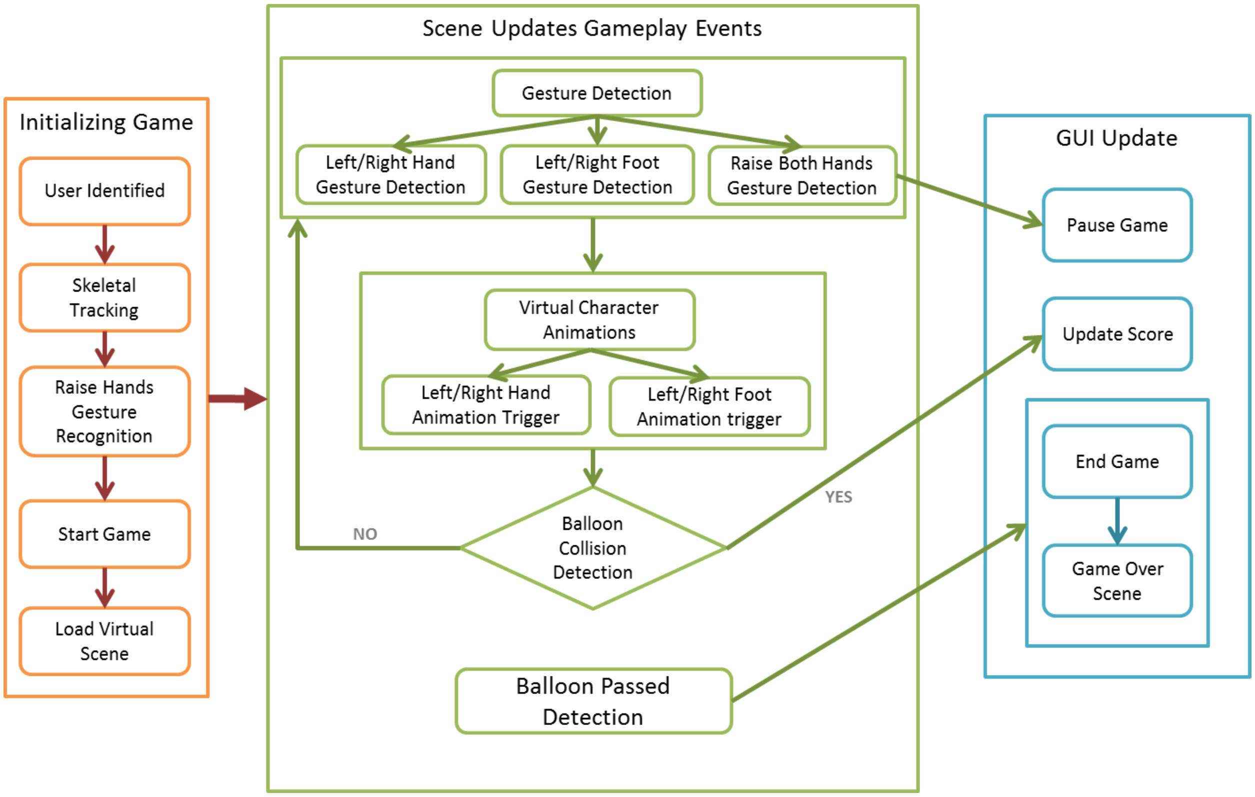

Initializing Game
User Identified
Skeletal Tracking
Raise Hands Gesture Recognition
Start Game
Load Virtual Scene
Scene Updates Gameplay Events
Gesture Detection
Left/Right Hand Gesture Detection
Left/Right Foot Gesture Detection
Raise Both Hands Gesture Detection
Virtual Character Animations
Left/Right Hand Animation Trigger
Left/Right Foot Animation trigger
NO
Balloon Collision Detection
YES
Balloon Passed Detection
GUI Update
Pause Game
Update Score
End Game
Game Over Scene

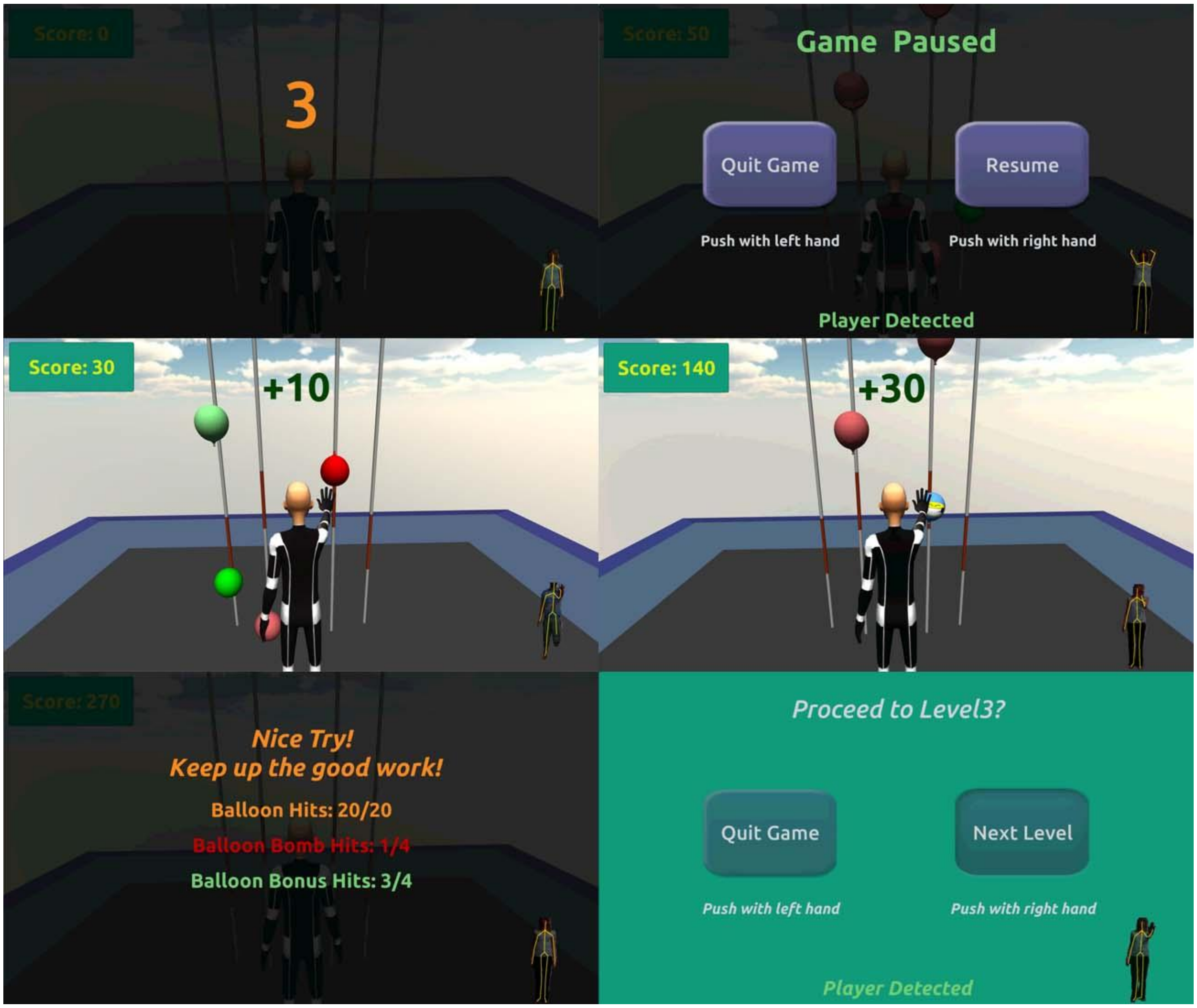
Figure 30: Balloon Goon general game logic diagram.

As soon as a user selects to play the Balloon Goon game a game screen will appear trying to identify the player. A success message will appear immediately if the user is in range of motion of the Kinect's cameras, indicating the he is identified by the device. If user is detected, the Kinect libraries will start the skeletal detecting process, and the game will display a small window at the right bottom of the screen, showing the depth/color image of the user and the joints that are tracked on his body. In order to start the game the user must perform a 'Raise Both Hands' gesture and as a result the 3D game environment is loaded upon the successful tracking of the gesture. A GUI countdown counter will appear notifying the user that the game will start in a few seconds. The same counter is shown each time the game is paused and resumed in order to provide the player enough time before continue playing.

Upon the countdown completion the game starts with the balloons start falling along the pillars in a specific speed that is depending on the current level played. Balloons are falling along four vertical posts, and each balloon falling along a different post can be popped by a different user movements. Balloons falling along the inner posts can be popped by 'Single Hand Raise & Extend' gestures while the balloon falling along the outer posts using 'Single Leg Raise & Extend' gestures. The game is checking for possible gestures at every game loop, which two of selected gestures to be detected are related to game control movements and one is related to a menu action. If the gesture detection process of the game logic identify a hand gesture it is trigger the correct hand animation to be executed by the 3D virtual character that is positioned in front of the vertical posts. The same effect happens when a foot gesture is detected, triggering this time the 3D character's leg animations. The previously mention animations change the posture status of the virtual character inside the game, and depended on the animation triggered if the character will perform a left/right hand or left/right leg movement in order to pop the balloons falling in front of him.

In order the character to successfully collide with a specific balloon inside the game environment, the correct animation must be triggered at the right time. In order to notify the user that the balloon falling can be popped, the texture color is changed and the balloon is becoming brighter. If the user perform the gesture at the right time using the appropriate gesture depending on the pillar along which the balloon falls, a collision it is performed and the balloon is popped. Successfully popped balloons trigger a sound and perform an animated movement in order to notify the user that the action was successfully executed and the game score is increased. On higher levels two different types of balloons exist that can be popped in the same way but they give different points and can be identified by the texture image which is different from the regular balloons. The first one can give bonus points while the other can decrease the score so it must be avoided. On the other hand, if a balloon is missed no points are added in the game score and the balloon is destroyed.

Anytime during the gameplay, a user may execute a 'Raise Both Hands' gesture in order to pause the game. A pause menu is displayed on the screen while the game environment is shown faded in the background. The pause menu scene give the user two options, one to resume playing and another to quit the game, followed by on screen text instructions on select each option. In order to select a specific option the user must perform a 'Raise Hand & Extend' gesture with the left or right hand depending on the desirable action, the as one used for popping balloons. The same menu screen appears if during the gameplay the user is lost from the FOV of the Kinect device. If the detected user during the pause menu screen select to resume the game continues upon the completion of the countdown counter, otherwise if the user selects to quit, the game exits and the game selection menu appears. The Balloon Goon game level ends when all the balloons have passed through, either if the player successfully popped them or not. Upon level completion a game screen appears displaying information about the balloon that popped or missed during the game, and a text notifying the user about its progress. After a few seconds the screen changes and the game asks the player to proceed to next level or quit the game, displaying a similar interactive screen like the one appears when the menu is paused.



Nice Try!
Keep up the good work!
Balloon Hits: 20/20
Balloon Bomb Hits: 1/4
Balloon Bonus Hits: 3/4

Figure 31: Balloon Goon gameplay. Upper left image shows the countdown counter; Upper right image is displaying the pause menu screen; Middle images show snapshots of the gameplay, with the middle right displaying popping a bonus balloon; Lower left image shows the end game screen, by displaying the progress of the user; Lower right image displays the menu screen that appears in order to proceed to the next level.

Skiing Game

Users playing the Skiing game are motivated into performing several core and hand gestures in order to control a virtual character and gather the prizes that are located inside the game environment in different spots as also avoid obstacles. The specific game is composed by a single level, during which the difficulty is increased progressively, by adding special goals and obstacles, appearing more frequently, as the game progress. The game is based on gestures targeting the patient's flexibility, muscles strength and balance as also focuses on improving the cognitive reaction of the users, by executing timely depended gestures in order to collect prizes or avoid the obstacles. The general game logic is presented in the Figure xx.

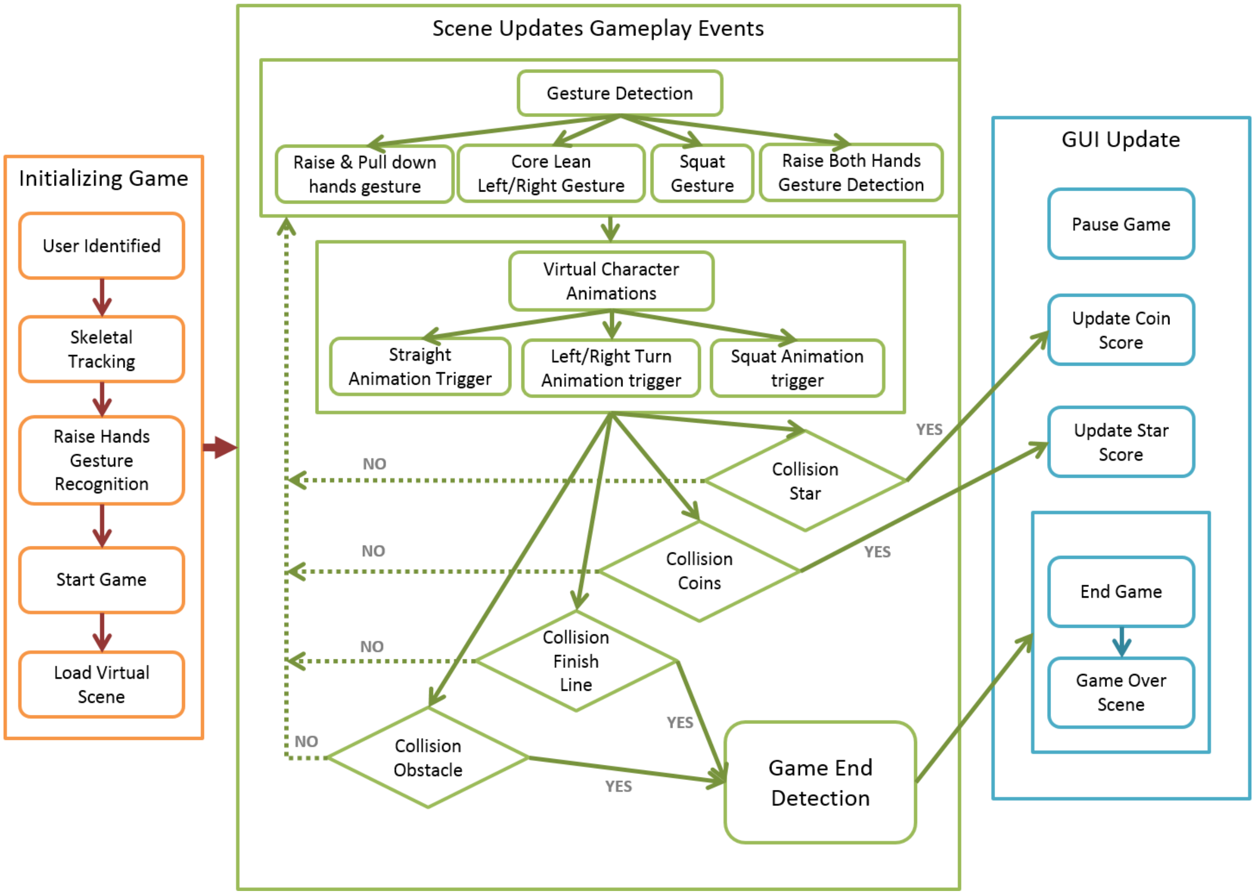


Figure 32: Skiing general game logic diagram.

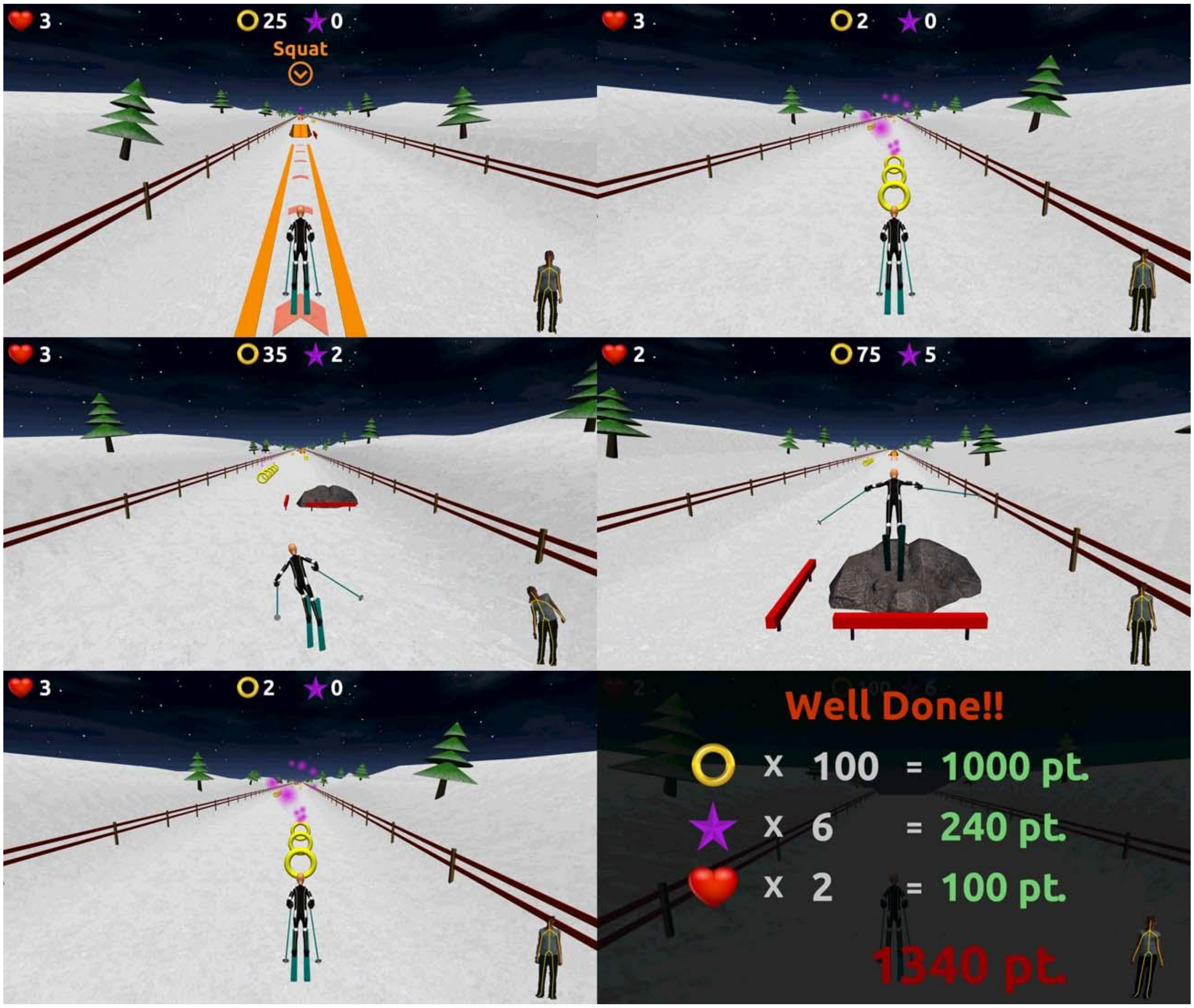
The game starts similar to the Balloon Goon game. It is necessary the user to be identified before the game starts. As soon as the user is identified and the skeletal detection have captured the skeletal information needed in order to monitor the position of the tracked joints, the game informs the user that in order to begin a 'Raise Both Hands' gesture must be performed. In case the Kinect device lost the user from its camera's FOV it displays a message waiting for the user to be visible again. On the other hand if everything is set up and the gesture is performed correctly the game begins, by firstly load the 3D game environment. In contrast to the Balloon Goon game no countdown counter is displayed because in order the 3D game character start moving a specific gesture must be executed by the user. While waiting the user to perform the start gesture, an idle animation is responsible for changing the posture status of the virtual character indicating that the game is in standup mode.

User must perform a 'Raise & Lower Both Hands' gesture in order to force the virtual character start moving in a straight line. The game environment is composed by two non-visible lanes, in which the 3D character can move straight during the gameplay, one on the right and one on the left side of the stage. Character upon start moving for the first time, is placed on one of the two lanes. In order to change lane he must perform a 'Side Lean Core' gesture to the left or right depending on the lane he wants the 3D character to move. Upon the detection of such a gesture an animation is triggered that is responsible for directing the 3D character to the correct side. In some cases the lean gesture cannot result on directing the virtual character to the lane beside, such as when the user is in the left lane and the user is still trying to turn left out of the gameplay scenery.

The user is motivated in directing the virtual character left or right in order to collect set of rings, which are positioned near the ground in various positions across both lanes. Upon a collision of the virtual character with the rings a sound is triggered and the score of rings is increased. Another reason in order the user is forced into performing the 'Side Lean Core' gesture is when he is trying to avoid the 3D character collide with obstacles placed also across the lanes. If the user doesn't successfully avoid an obstacle he loses one life and an animation is triggered that drives the character falling over the obstacle and landing on the ground. In that case the character stop moving and the user need to perform a 'Raise & Lower Both Hands' gesture again in order to start the virtual character again. If the user loses all of his lives the game ends.

In addition, in various places upon the lanes, ramps have been placed, with a star prize positioned after the ramp and a bit higher. If the virtual character is collide with the beginning edge of a ramp, a jump animation is triggered and the virtual character is performing a jump movement over the ramp. Although in this case (slow jump) the virtual character cannot successfully collide with the star in order to collect it. For collecting the star the user must perform a 'Squat' gesture just before the virtual character reaches the ramp. In order to notify the user that it is the right time to perform a squat a speed lane game object is positioned in the right place before the ramp. As soon as the 3D character is passing above the speed lane and the player performs a squat, an animation is triggered moving the character faster inside the game. Hitting the initial edge of a ramp with high speed the 3D character can perform a high jump animation which allows the user to collect the star prize successfully and the score of the star prize increases.

It is described that losing all the lives by stepping upon obstacles can cause the game to end. Another action that leads to the completion of the game is by successfully controlling the 3D character avoiding all the obstacles until the virtual character passes the finish line game object. In either of the two cases, in which the game ends, a progress screen is displayed informing the users about the stars and rings that was collected during the gameplay and the points gained through gathering those prizes. The specific screen is displayed for a few seconds and the next that follows is similar to the one appeared in balloon game, asking the user if he wants to restart the game or quit. The interactive end menu screen is controlled by the user, by performing 'Hand Raise & Extend' gestures. It is significant to add that also in this game the user can pause the game using 'Raise Both Hands' gesture, in which case the menu screen appears and the actions that can be performed are described in the Balloon Goon game.



Well Done!!

○ x 100 = 1000 pt.

★ x 6 = 240 pt.

❤ x 2 = 100 pt.

1340 pt.

Figure 33: Skiing gameplay. Upper left image is displaying user performing a squat in the speed lane; Upper right image is displaying that virtual character performing a high jump collecting the star; Middle left image shows user avoiding an obstacle by leaning to the left, while the middle right one displaying animation triggered when falling on obstacle; Lower left image shows virtual character collecting rings; Lower right image displays the screen indicating the score of user upon finishing the game

5.2 Game Scene design

The scene of both games are composed by 3D objects that were designed using the Autodesk 3ds Max 2016 (student license) and the textures that are attached on them, when this was necessary, were created using Adobe Photoshop CS2. On the other hand, the model of the virtual character was downloaded from the Unity3D asset's store, where it is provided for free and it was already rigged and skinned. Trying to create games that are feasible for playing in a wide variety of windows PC's, including those with older graphic cards or low CPU power and memory, the models was designed with as low polygons as it was possible in order not to affect the performance of the game but still be able to create a visually nice game environment. Some of the game objects created, play significant role to the progress of the game by interacting with the virtual character or other game objects during gameplay, while others are used as static scenery.

Balloon Goon game

The main scene of the game is composed as we mentioned before by four vertical posts and balloons that falling along those posts. Pillars that created have the same texture color covering the whole surface of the object, except from a specific area that it was textured using a different color. That area is used in order to give an indication to the user about when it is the right time to pop the balloons. When a specific balloon is coming along this area of the post, its texture color chances and the balloon is becoming brighter. The balloons are textured with two different colors, indicating which of them can be popped by using a hand gesture in contrast with those that can be popped using a leg gesture. In addition at the third level of the game there are two extra types of balloon that falling along the posts, the bonus balloon that when is popped gives extra points and the bomb balloon that decrease the total points. In order the user be able to distinguish them, a smiley face texture was attached to the bonus balloon and a bomb ready to explode texture to the bomb balloon.

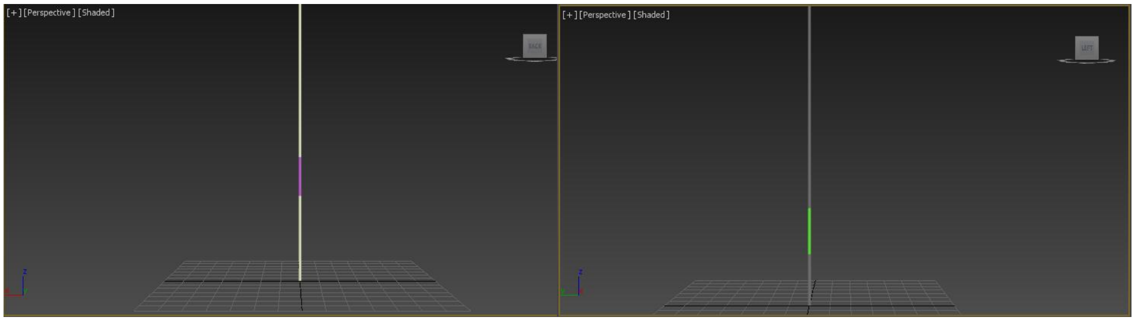


Figure 34: Hand and foot pillar 3D objects

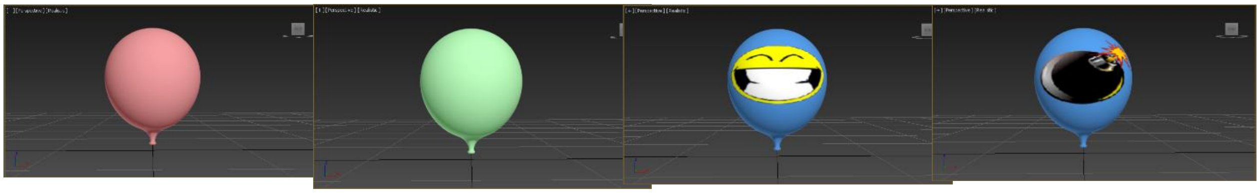


Figure 35: Balloons as shown in order in the figure are the balloon popped with hand gestures, the balloon popped with foot gesture, the bonus balloon and the bomb balloon.

The balloon game scene is also composed by a terrain that is static and doesn't play any significant role in the gameplay. The camera displaying the game is positioned behind and higher from the 3d virtual character in order the user will have a clear picture of the balloons falling. Also two directional lights were used in order to light the scene properly. The skybox that is shown on the background of the game is chosen carefully in order not to distract the user during gameplay.

Skiing game

The scene of the skiing game is composed by several 3D objects that are designed in order to build the static scenery of the game, the ski equipment that are attached on the virtual character and the objects that can interact with the virtual character. The camera displaying the scene is attached to the virtual character model in order to follow him during the gameplay. Ski equipment objects are also attached to the virtual character model in order to be moved during the animations. The rest of the objects that can interact with the character during gameplay are ramps, speed lanes, sidebars, coins, stars and rocks that are positioned as obstacles. For the specific game a darker skybox was selected to be shown in the background in order to create a contrast with the terrain, which is white due to the snowy effect. While the base of 3D objects was created in 3ds Max, the majority of the game object that interacts with the virtual character during gameplay where recomposed inside Unity3D environment in order to create grouped game objects. As an example the ramp game object used by the virtual character in order to perform a jump, is composed by the ramp 3D model, combined with the speed lane and three sidebars. The combined objects were saved inside Unity as prefabs in order to be easier to be positioned multiple times when setting a larger scene. Those grouped game object along with the scenery objects were combined into game objects sets during the development as ready to be position stage parts in order to create a complete game stage. Using the specific premade sets it's faster to create different stages with varied difficulty by positioning these stage sets in the desirable position inside the game environment.

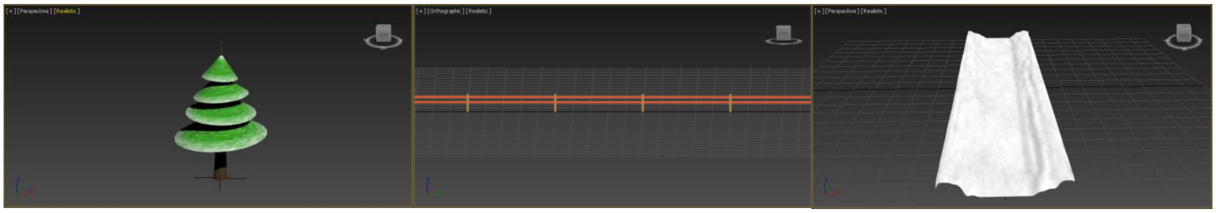


Figure 36: Skiing game static scenery game objects are composed by trees, fences and the game terrain.

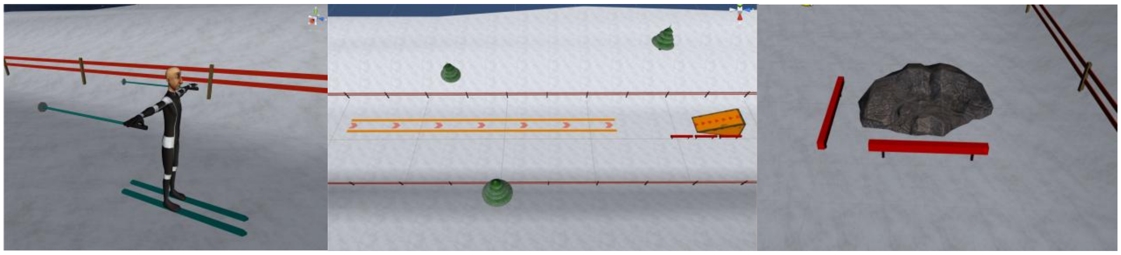


Figure 37: Skiing game premade grouped game objects. In the left it is presented the virtual character along with the ski equipment, in the middle the speed lane and the ramp group object is shown and at the right image the obstacle premade object is displayed.

6. Discussion and Future Work

Patients with Parkinson's disease can benefit from daily physical training, improving their mobility, postural control, balance and strength. In the current thesis a game platform was presented, and two 3D exergames were designed and developed, which can offer physical rehabilitation through a playful and immersive virtual environment. The proposed exergames offer the potential for PD patient to perform exercises selected from training programs dedicated to the disease, in a loose environment. Kinect device was used in order to detect the user's movement, which has the advantage of motion capturing that isn't depended on handheld controllers or external raised platforms that may trouble patients with PD. In addition Kinect is a low cost commercial solution that can be easily set in both clinical and home environments.

The developed exergames was designed to be easily operated by people with PD disease without the need of external help. The platform uses Natural User Interface in order to interact with the users that can navigate or play the proposed games without the use of any addition computer peripheral device. Navigation and interaction with the virtual environment during the gameplay can be achieved using specific body movements which are in most cases accompanied by visual guided instructions. Also the platform use personalized body measurements in order to specifically adjust the detection of those movement for each user.

Although in order to guaranty the feasibility and the safety of the proposed solution several interventions must be conducted in both home and clinical environments. Feedback from patients is really important in order to ensure that user requirements based on which the current solution was designed, are validated. Based also on the literature that was examined during the current thesis, not much have been done in the context of using Kinect to create exergames for PD patients and as a result, it is really important to further examine the suitability of the device as a possible rehabilitation tool for PD.

The development of the presented platform is at an early stage of development and certain improvements must be implemented in order to be sustained as a complete solution for PD rehabilitation, through the use of exergames. Several games must be designed and developed for people with more severe symptoms of the disease, such as games that can be played from seated position. In addition, an extended set of levels is required to be developed for the games that are already developed, in order to motivate patient into participating to game scenarios with additional increased difficulty, in both physical and cognitive level. It is also to significant to examine how movement or game-based historic data that are stored during the gameplay can be proved useful for attending physician, concerning the status of the patient.

Lastly is it important to mention that recently Microsoft have released a new version of Kinect sensor device, Kinect v2. The new device use a time to flight camera that can provide higher depth capturing resolution and wider field of view. In can also provide skeletal detection that can recognize 6 more joints (fingers) than the version used in the current platform. It is necessary to be tested in order to validate if it can be used for providing more accurate results in terms of gesture recognition.

7. Bibliography

[1] "EPDA." [Online]. Available: http://www.epda.eu.com/en/. [Accessed: 12-Jun-2015].

[2] "Parkinsons Disease Information." [Online]. Available: http://www.parkinsons.org/. [Accessed: 12-Jun-2015].

[3] K. R. Chaudhuri, D. G. Healy, and a H. Schapira, "Non-motor symptoms of Parkinson's disease: diagnosis and management," *Lancet Neurol*, vol. 5, no. 3, pp. 235–245, 2006.

[4] J. Jankovic, "Parkinson's disease: clinical features and diagnosis.," *J. Neurol. Neurosurg. Psychiatry*, vol. 79, no. 4, pp. 368–376, 2008.

[5] J. a Opara, W. Brola, M. Leonardi, and B. Błaszczyk, "Quality of life in Parkinson's disease.," *J. Med. Life*, vol. 5, no. 4, pp. 375–81, 2012.

[6] B. L. Den Oudsten, G. L. Van Heck, and J. De Vries, "Quality of life and related concepts in Parkinson's disease: a systematic review.," *Mov. Disord.*, vol. 22, no. 11, pp. 1528–1537, 2007.

[7] D. J. Gelb, E. Oliver, and S. Gilman, "Diagnostic criteria for Parkinson disease.," *Arch. Neurol.*, vol. 56, no. 1, pp. 33–39, 1999.

[8] J. Jankovic, K. S. Schwartz, and W. Ondo, "Re-emergent tremor of Parkinson ' s disease," pp. 646–650, 1999.

[9] G. Wulf, M. Landers, R. Lewthwaite, and T. Töllner, "External focus instructions reduce postural instability in individuals with Parkinson disease.," *Phys. Ther.*, vol. 89, no. 2, pp. 162–168, 2009.

[10] W. C. Koller, S. Glatt, B. Vetere-Overfield, and R. Hassanein, "Falls and Parkinson's disease 31," *Clin.Neuropharmacol.*, vol. 12, no. 2. pp. 98–105, 1989.

[11] M. M. Hoehn and M. D. Yahr, "Parkinsonism : onset , progression , and mortality," vol. 17, no. May, 1967.

[12] R. Bhidayasiri and D. Tarsy, "Movement Disorders: A Video Atlas," pp. 4–6, 2012.

[13] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. T. Stebbins, S. Fahn, P. Martinez-Martin, W. Poewe, C. Sampaio, M. B. Stern, R. Dodel, B. Dubois, R. Holloway, J. Jankovic, J. Kulisevsky, A. E. Lang, A. Lees, S. Leurgans, P. a. LeWitt, D. Nyenhuis, C. W. Olanow, O. Rascol, A. Schrag, J. a. Teresi, J. J. van Hilten, N. LaPelle, P. Agarwal, S. Athar, Y. Bordelan, H. M. Bronte-Stewart, R. Camicioli, K. Chou, W. Cole, A. Dalvi, H. Delgado, A. Diamond, J. P. Dick, J. Duda, R. J. Elble, C. Evans, V. G. Evidente, H. H. Fernandez, S. Fox, J. H. Friedman, R. D. Fross, D. Gallagher, C. G. Goetz, D. Hall, N. Hermanowicz, V. Hinson, S. Horn, H. Hurtig, U. J. Kang, G. Kleiner-Fisman, O. Klepitskaya, K. Kompoliti, E. C. Lai, M. L. Leehey, I. Leroi, K. E. Lyons, T. McClain, S. W. Metzer, J. Miyasaki, J. C. Morgan, M. Nance, J. Nemeth, R. Pahwa, S. a. Parashos, J. S. J. S. Schneider, A. Schrag, K. Sethi, L. M. Shulman, A. Siderowf, M. Silverdale, T. Simuni, M. Stacy, M. B. Stern, R. M. Stewart, K. Sullivan, D. M. Swope, P. M. Wadia, R. W. Walker, R. Walker, W. J. Weiner, J. Wiener, J. Wilkinson, J. M. Wojcieszek, S. Wolfrath, F. Wooten, A. Wu, T. a. Zesiewicz, and R. M. Zweig, "Movement Disorder Society-Sponsored Revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results," *Mov. Disord.*, vol. 23, no. 15, pp. 2129–2170, 2008.

[14] "International Parkinson and Movement Disorder Society." [Online]. Available: http://www.movementdisorders.org/MDS.htm. [Accessed: 12-Jun-2015].

[15] "American Parkinson Disease Association." [Online]. Available: http://www.apdaparkinson.org/. [Accessed: 12-Jun-2015].

[16] M. Horstink, E. Tolosa, U. Bonuccelli, G. Deuschl, a. Friedman, P. Kanovsky, J. P. Larsen, a. Lees, W. Oertel, W. Poewe, O. Rascol, and C. Sampaio, "Review of the therapeutic management of Parkinson's disease. Report of a joint task force of the European Federation of Neurological Societies and the Movement Disorder Society-European Section. Part I: Early (uncomplicated) Parkinson's disease," *Eur. J. Neurol.*, vol. 13, no. 11, pp. 1170–1185, 2006.

[17] M. R. Semenchuk, "Medical Management of Parkinson's Disease, (American Parkinson Disease Association-APDA)."

[18] "Exercises for people with parkinson's." [Online]. Available: http://www.parkinson.ca/atf/cf/%7B9ebd08a9-7886-4b2d-a1c4-a131e7096bf8%7D/EXERCISEMAR2012_EN.PDF.

[19] V. a. Goodwin, S. H. Richards, R. S. Taylor, A. H. Taylor, and J. L. Campbell, "The effectiveness of exercise interventions for people with Parkinson's disease: A systematic review and meta-analysis," *Mov. Disord.*, vol. 23, no. 5, pp. 631–640, 2008.

[20] A. Ashburn, L. Fazakarley, C. Ballinger, R. Pickering, L. D. McLellan, and C. Fitton, "A randomised controlled trial of a home based exercise programme to reduce the risk of falling among people with Parkinson's disease.," *J. Neurol. Neurosurg. Psychiatry*, vol. 78, no. 7, pp. 678–684, 2007.

[21] N. Smania, E. Corato, M. Tinazzi, C. Stanzani, A. Fiaschi, P. Girardi, and M. Gandolfi, "Effect of balance training on postural instability in patients with idiopathic Parkinson's disease.," *Neurorehabil. Neural Repair*, vol. 24, no. 9, pp. 826–834, 2010.

[22] R. Kizony, P. L. (Tamar) Weiss, M. Shahar, and D. Rand, "TheraGame: A home based virtual reality rehabilitation system," *Int. J. Disabil. Hum. Dev.*, vol. 5, no. 3, pp. 209–214, 2006.

[23] A. Caglar, H. Gurses, F. Mutluay, and G. Kiziltan, "Effects of home exercises on motor performance in patients with <B>Parkinson's disease</B>," *Clin. Rehabil.*, vol. 19, no. 8, pp. 870–877, 2005.

[24] B. G. Farley and G. F. Koshland, "Training BIG to move faster: the application of the speed-amplitude relation as a rehabilitation strategy for people with Parkinson's disease.," *Exp. Brain Res.*, vol. 167, no. 3, pp. 462–467, 2005.

[25] E. Eckstrom, R. Stock, J. Galver, G. Maddalozzo, D. Ph, and S. S. Batya, "Tai Chi and Postural Stability in Patients with Parkinson's Disease," pp. 511–519, 2013.

[26] J. Zhou, T. Yin, Q. Gao, and X. C. Yang, "A Meta-Analysis on the Efficacy of Tai Chi in Patients with Parkinson ' s Disease between 2008 and 2014," vol. 2015, 2015.

[27] S. a Combs, M. D. Diehl, W. H. Staples, L. Conn, K. Davis, N. Lewis, and K. Schaneman, "Boxing training for patients with Parkinson disease: a case series.," *Phys. Ther.*, vol. 91, no. 1, pp. 132–142, 2011.

[28] O. Assad, R. Hermann, D. Lilla, B. Mellies, R. Meyer, L. Shevach, S. Siegel, M. Springer, S. Tiemkeo, J. Voges, J. Wieferich, M. Herrlich, M. Krause, and R. Malaka, "Motion-based games for Parkinson's disease patients," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6972 LNCS, pp. 47–58, 2011.

[29] F. A. D. S. Mendes, J. E. Pompeu, A. M. Lobo, K. G. da Silva, T. D. P. Oliveira, A. P. Zomignani, and M. E. P. Piemonte, "Motor learning, retention and transfer after virtual-reality-based training in Parkinson's disease - effect of motor and cognitive demands of games: A longitudinal, controlled clinical study," *Physiother. (United Kingdom)*, vol. 98, no. 3, pp. 217–223, 2012.

[30] G. D. P. Vieira, D. Freitas, and G. Henriques, "Virtual Reality in Physical Rehabilitation of Patients With Parkinson ' S Disease," vol. 24, no. 1, pp. 31–41, 2014.

[31] G. Barry, B. Galna, and L. Rochester, "The role of exergaming in Parkinson's disease rehabilitation: a systematic review of the evidence.," *J. Neuroeng. Rehabil.*, vol. 11, no. 1, p. 33, 2014.

[32] W. Yu, C. Vuong, and T. Ingalls, "An interactive multimedia system for Parkinson's patient rehabilitation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6774 LNCS, no. PART 2, pp. 129–137, 2011.

[33] F. L. Tous, P. Ferriol, M. a. Alcalde, M. Melià, B. Milosevic, M. Hardegger, and D. Roggen, "Incorporating the rehabilitation of parkinson's disease in the play for health platform using a body area network," *IFMBE Proc.*, vol. 41, pp. 1481–1484, 2014.

[34] K. K. Zettergren, M. S. Antunes, J. M. Canhao, and C. Lavallee, "the Effects of Nintendo Wii Fit on Gait Speed, Balance and Functional Mobility on Idiopathic Parkinson'S Disease: a Case Study," *Gerontologist*, vol. 51, no. 2, p. 70, 2011.

[35] P. V. Mhatre, I. Vilares, S. M. Stibb, M. V. Albert, L. Pickering, C. M. Marciniak, K. Kording, and S. Toledo, "Wii Fit Balance Board Playing Improves Balance and Gait in Parkinson Disease," *PM R*, vol. 5, no. 9, pp. 769–770, 2013.

[36] J. F. Esculier, J. Vaudrin, P. Bériault, K. Gagnon, and L. E. Tremblay, "Home-based balance training programme using Wii Fit with balance board for Parkinson's disease: A pilot study," *J. Rehabil. Med.*, vol. 44, no. 2, pp. 144–150, 2012.

[37] J. E. Pompeu, F. A. D. S. Mendes, K. G. Da Silva, A. M. Lobo, T. D. P. Oliveira, A. P. Zomignani, and M. E. P. Piemonte, "Effect of Nintendo Wii^{TM}-based motor and cognitive training on activities of daily living in patients with Parkinson's disease: A randomised clinical trial," *Physiotherapy*, vol. 98, no. 3, pp. 196–204, 2012.

[38] N. B. Herz, S. H. Mehta, K. D. Sethi, P. Jackson, P. Hall, and J. C. Morgan, "Nintendo Wii rehabilitation ('Wii-hab') provides benefits inParkinson's disease," *Park. Relat. Disord.*, vol. 19, no. 11, pp. 1039–1042, 2013.

[39] I. T. Paraskevopoulos, E. Tsekleves, C. Craig, C. Whyatt, and J. Cosmas, "Design guidelines for developing customised serious games for Parkinson's Disease rehabilitation using bespoke game sensors," *Entertain. Comput.*, vol. 5, no. 4, pp. 413–424, 2014.

[40] B. Galna, D. Jackson, G. Schofield, R. McNaney, M. Webster, G. Barry, D. Mhiripiri, M. Balaam, P. Olivier, and L. Rochester, "Retraining function in people with Parkinson's disease using the Microsoft kinect: game design and pilot testing.," *J. Neuroeng. Rehabil.*, vol. 11, no. 1, p. 60, 2014.

[41] R. Hermann and M. Herrlich, "Strong and Loose Cooperation in Exergames for Older Adults with Parkinson s Disease," *Mensch Comput. …*, 2013.

[42] *Exercise program for Parkinson disease*. Novartis Pharmaceuticals Corporation.

[43] "Be Active! An exercise program for people with Parkinson's disease." [Online]. Available: http://www.apdaparkinson.org/uploads/files/BeActive_Feb2008-enw.pdf. [Accessed: 22-Jun-2015].

[44] "Parkinson's UK - Keeping Moving - Exercise and Parkinson's booklet and DVD." .

[45] "Exercise - Parkinson Society Canada." .

[46] "National Parkinson Foundation - Fitness Counts." .

[47] "KNGF Guidelines for physical therapy in patients with Parkinson's disease." .