Intelligent Control and Automation, **2013, 4, 250-262**
http://dx.doi.org/10.4236/ica.2013.43030 Published Online August 2013 (http://www.scirp.org/journal/ica)

A Video Game Based on Elementary Differential Equations

Marco Giacinti[1], Francesca Mariani[2*], Maria Cristina Recchioni[1], Francesco Zirilli[2]
[1]Dipartimento di Scienze Sociali "D. Serrani", Università Politecnica delle Marche,
Ancona, Italy
[2]CERI-Centro di Ricerca Previsione Prevenzione e Controllo dei Rischi Geologici,
Universitµa di Roma "La Sapienza", Valmontone, Italy
Email: m.giacinti@univpm.it, *fra_mariani@libero.it, m.c.recchioni@univpm.it, f.zirilli@caspur.it

Received April 10, 2013; revised May 10, 2013; accepted May 17, 2013

ABSTRACT

In this paper a prey-predator video game is presented. In the video game two predators chase a prey that tries to avoid the capture by the predators and to reach a location in space (*i.e.* its "home"). The prey is animated by a human player (using a joypad), the predators are automated players whose behaviour is decided by the video game engine. The purpose of the video game is to show how to use mathematical models to build a simple prey-predator dynamics representing a physical system where the movements of the game actors satisfy Newton's dynamical principle and the behaviour of the automated players simulates a simple form of intelligence. The game is based on a simple set of ordinary differential equations. These differential equations are used in classical mechanics to describe the dynamics of a set of point masses subject to a force chosen by the human player, elastic forces and friction forces (*i.e.* viscous damping). The software that implements the video game is written in C++ and Delphi. The video game can be downloaded from: http://www.ceri.uniroma1.it/ceri/zirilli/w9.

Keywords: Video Game; Differential Games; Mechanical Dynamical System

1. Introduction

This paper illustrates an example of how a physical system modeled by elementary mathematics can be used as the basis of a video game where the (human) players must move considering some restrictions to their movements (and of the automatic characters as well) that are implied by Newton's dynamical principle. Moreover they must consider the physical characteristics of the automatic characters (*i.e.* their masses, velocities, ...) together with their elementary form of artificial intelligence (AI). The target age of the players is made of the children between five and thirteen years. A user experiment we conducted suggests that children in that age group playing the game develop an intuitive understanding of Newton's dynamical principle. The children aged between five and eight years enjoy the game. The fun of the game decreases when the age of the children involved increases. Teenagers could be interested in the game more as a learning experience than as a recreational experience. In fact the video game can be seen as a compelling context where virtual experiments about Newton's dynamical principle can be done. The video game presented in Sections 3 and 4 considers the situation where two predators (the automatic characters) chase a prey (the player). The predators want to capture the prey. The prey has to avoid the capture and reach a given location in space (*i.e.* its "home"). This game is formulated using appropriate mathematical models.

The video game has two purposes. The first one is to show how mathematical models, such as optimal control problems and differential games, can be used to model a physical system, that is the basis of the video game, where the actors movements satisfy Newton's dynamical principle and the automatic characters have a simple form of artificial intelligence (AI). The second purpose is to provide an example of video game where players in the choice of their strategy must consider the restrictions to their movements and to the movements of their opponents coming from Newton's dynamical principle. They also must understand how the AI of the automatic characters works to anticipate their behaviour.

The game is an example of how a simple mathematical

[*]Corresponding author.

model inspired by physics can be exploited to build an interesting learning experience [1].

A second form of AI that goes beyond the AI of the automatic characters is the one that adapts the game to the ability of the (human) players. The behaviour of a (human) player in a sequence of game matches can be monitored by the game logic, for example, making an elementary statistical analysis of the "points" scored by the player in the matches. This statistics contains information about the learning experience and the acquired ability of the player and can be used to modify the behaviour of the automatic characters. The behaviour of the automatic characters is modified changing the values of the parameters contained in the equations that define their movements. In this way it is possible to calibrate the game difficulty (*i.e.* the effectiveness of the predators in chasing the prey) on the basis of the ability of the player. Serious games are a way to improve knowledge and to acquire skills (see, for example, [2] and the references therein). This can be done using the so called experience engines [2]. These engines select task sequences based on the current player profile. The purpose of using this profile to improve the learning process of the player building ad hoc tasks designed for the specific player goes beyond our purposes here. The present version of the game adapts the game to the ability of the (human) player in a very simple way, in fact the game has three levels of increasing difficulty and it is necessary to score a certain number of points in a sequence of matches to go from one level of the game to the next one.

Let us now concentrate on the mathematical model of the physical system considered and on its solution. The video game proposed is modeled as a rendez vous problem, that is as an evader pursuit game. Rendez vous problems involving two actors (or two sets of actors referred in the sequel as Sets 1 and 2) whose dynamics is defined by systems of differential equations have been widely studied in the scientific literature and have many applications in science and engineering. These problems are formulated as optimal control problems (see, for example, Athans [3]), or as differential games (see, for example, Athans [3], Isaacs [4], Melikyan [5]). These references are only indicative since the scientific literature on this subject is huge.

Roughly speaking, the use of an optimal control problem as a mathematical model of the "rendez vous" problem assumes that the actors belonging, say, to Set 1, do not pursue a goal conflicting with the goal pursued by the actors of Set 2. These actors do not react to the action of Set 2 actors and move on trajectories implicitly defined by differential equations pursuing their own goal, for example, the goal of reaching their home. Moreover the model assumes that the trajectories of set 1 actors are at least in part known to set 2 actors and that these last actors rely on this knowledge to choose a "strategy" (possibly optimal) to achieve a goal that involves the set 1 actors (*i.e.*, for example, to have a "rendez vous") minimizing a cost functional.

However, in many applications it is considered the situation where not only set 2 actors pursue a goal (*i.e.* the rendez vous) through a (possibly optimal) strategy but also set 1 actors pursue a goal at least in part conflicting with the goal of set 2 actors (*i.e.*, for example, to avoid the rendez vous and/or to reach a certain position in space called "home") through a (possibly optimal) strategy. In this situation, the mathematical model provided by control theory is insufficient and models coming from game theory must be used. In our case, since the actors of the game move on trajectories implicitly defined by differential equations (describing the constraints imposed by physical laws) and the two sets of actors present in the game have conflicting goals, differential games are the natural mathematical model to consider. These models must be solved in a closed loop form in order to be useful in the video game context where the players must react in real time to the developments on the game scene. In fact, the closed loop solution of the mathematical model (*i.e.* the control problem eventually coming from the differential game formulation) that defines the behaviour of the automatic characters (Set 2 actors) does not require the knowledge of the state variables of the set 1 actors (human players) over the entire time period of the game action in order to compute the control variables (*i.e.* the variables that determine the movements of the automatic characters (set 2 actors)). In the closed loop solution the control variables at the current time are obtained using only the knowledge of the state variables of set 1 and set 2 actors until the current time. A similar statement holds for the human players that animate the set 1 actors and for all the players when closed loop solutions of differential games are considered. In fact the human players choose their control variables (*i.e.* the variables that determine the movements of set 1 actors) using a joypad on the basis of the knowledge of the game scene shown on the computer screen up to the current time.

The use of differential games and optimal control problems to develop prey-predator video games is an interesting idea that is worth investigating. In fact, prey-predator video games can be realized using simpler mathematical approaches (see for example [6-9]). For example in [6] and [7] mathematical models based on system of ordinary differential equations to describe the movements of automatic characters are presented. These models range from simple heuristic rules to sophisticated ones that describe an articulated body consisting of rigid links connected by joints. In absence of joints, each link has six degrees of freedom, three translational and three rotational. In [6] and [7] the problem of modeling players that are able to make decisions is not considered. In [9] the behaviour of the automatic characters is based on pathfinding algorithms, such as Dijkstra pathfinding algorithm (Section 4.2 of [9]), and decision making capability of these players is based on the analysis of random trees. Moreover, in [9] Chapter 7, a detailed discussion of several approaches for the management of learning experiences based on AI and on the use of statistical analysis of the players performances in the game after having played a given number of matches is proposed (see Section 7.3.2 of [9]). In [8] a prey-predator system is studied to develop game-like interaction between prey and predators. The mathematical model proposed in [8] characterizes each actor of the game by its position vector and by a unit vector called "heading direction". The heading direction is not necessarily the velocity of the actor. The dynamics of the actors is described by a system of ordinary differential equations that depends on the interactions among actors. The function that defines the interactions between the actors is inspired by potential theory (see [8] for further details). The model used in [8] does not involve differential games or optimal control problems.

The prey predator video games based on differential games (or on optimal control problems) are a class of sophisticated games where teams of (human and automated) players interact pursuing conflicting goals using natural intelligence and AI. In fact in these video games teams of human players, teams of automatic characters and mixed teams made of human and automatic characters can interact with their opponent teams. The behaviour of the players originating from these interactions is implicitly defined choosing the cost functionals, the constraints and the parameter values of the mathematical models employed. Note that when differential games and optimal control models are used it is easy to fit into the video game the relevant physics laws. In fact the physics laws can be considered as constraints of the models.

The closed loop solution of optimal control problems or of differential games is usually difficult. In fact it requires finding the solution of a system of nonlinear partial differential equations known as Hamilton Jacobi equations (see [4]). The numerical solution of the Hamilton Jacobi equations can be very challenging. Very few examples of models where the Hamilton Jacobi equations can be solved explicitly using elementary formulae are known. Recently some research has been devoted to the objective of finding explicit formulae or numerical algorithms to solve (not necessarily in closed loop form) differential games that model problems in the applied sciences (see for example [10] and the reference therein).

However, for the purposes of the video game presented here it is not necessary to solve these models in a mathematically rigorous or numerically accurate way, it is sufficient to find approximate (suboptimal) solutions of the models that give satisfactory results from the practical point of view. That is, the suboptimal solutions considered must generate a credible form of AI for the automatic characters, which means that the behaviour of the automatic characters on the game scene must be coherent with their goals. Moreover, these (suboptimal) solutions must be easy to compute in comparison with the numerical solution of the Hamilton Jacobi equations associated with the mathematical models used. Note that the suboptimal solutions employed in the video game presented here do not maximize the utility functions of the players, in this sense they are suboptimal. However they satisfy exactly the constraints, that is, they satisfy exactly Newton's dynamical principle. The differential equations that define the dynamics of the prey and of the predators are inspired by classical mechanics. In the case of the prey the equations represent the dynamics of a point mass subject to an external force (chosen by the human player using the joypad) and to a friction force, in the case of the (two) predators the equations model the dynamics of two point masses subject to elastic forces and to friction forces. The elastic forces acting on the predators depend on the prey position. The point masses representing the prey and the predators move in the Euclidean plane obeying Newton's dynamical principle.

The solution of the differential equations that describe the predators can be written with explicit formulae. This solution together with the prey trajectory determined by the human player using the joypad can be thought of as a suboptimal solution of the differential game used as a mathematical model of the rendez vous problem. In fact these trajectories satisfy the constraints of the differential game, that is they are solutions of the differential equations that express Newton's dynamical principle, but do not define a minimax point of the objective function of the differential game (in this sense they are suboptimal). The suboptimal solution considered is a reasonable compromise between the desire of having a simple formula to compute the trajectories that can be easily implemented in the game program and the desire of having an accurate solution of the mathematical problem that models the physical system implemented in the video game. In fact the suboptimal solution considered is easy to compute and the resulting behaviour of the automatic characters is coherent with their goal in the game (*i.e*. the automatic characters try to have a rendez vous with the prey). Alternatively the solution of the differential equations that determine the automatic characters behaviour can be thought of as a suboptimal solution of an optimal control problem that defines the behaviour of the automatic characters in the video game.

Moreover, the suboptimal solution chosen has a physical meaning related to the action of simple elastic and friction forces between the game actors. This makes intuitive the choice of the values of the parameters (elastic constants, masses, friction coefficients, ...) appearing in the suboptimal solution and makes easy to understand what to do to change the effectiveness of the automatic characters in pursuing the rendez vous. This suboptimal solution has an educational value for the player, as it makes him aware of the elementary properties of elastic and friction forces (*i.e.* viscous damping). In fact, the player acting on the joypad must consider that because of the mass and velocity of the prey and of the bounds on the magnitude of the active forces available, the prey will be unable to do some movements such as, for example, changing direction of motion too quickly. On the other hand also the predators have their own masses and friction coefficients and move according to Newton's dynamical principle. This implies that also their rendez vous strategies must be chosen under some physical constraints. That is, the automatic characters neither can change direction of motion too quickly. In fact, they can change direction of motion in a way that depends on their masses, velocities, friction coefficients and on the active forces acting on them. These physical constraints are implicitly written in the equations of motion satisfied by the game actors and are implemented in the game engine through the solution of these equations.

Concluding, the idea of developing games respectful of physical laws is an interesting idea that is widely exploited (see, for example [6,7]), in particular in state of the art commercial video games. However the realization, using elementary mathematics, of a video game based on differential games to model systems where the actors simulate an intelligent behaviour is new. We stress that the video game presented here is a prototype and is not intended to be a commercial video game.

The video game is implemented in C++ and Delphi. It can be downloaded from the website:
http://www.ceri.uniroma1.it/ceri/zirilli/w9/.

In Section 2, we describe the "rendez vous" problem and its formulation as an optimal control problem or as a differential game. In Section 3 we give a suboptimal solution of the mathematical models introduced in Section 2. The software implementation of this suboptimal solution is the video game engine. In Section 4 we describe some technical and practical details of the implementation of the video game. Finally in Section 5 we draw some conclusions.

2. The Mathematical Model of the "Rendez Vous" Problem

Let us formulate an optimal control problem and a differential game problem that will be used as mathematical models in the development of the video game. These problems are sufficiently general to describe several situations where rendez vous between Set 1 actors and Set 2 actors take place. In these problems the dynamics of each actor is described by a system of linear ordinary differential equations. Let t be a real variable that denotes time. We assume that the state of the Set 1 actors and the state of the Set 2 actors at time t are described by vectors belonging to suitable real Euclidean spaces and that a similar statement holds for the control variables used to govern these state vectors. The dynamics of the state vectors is defined through differential equations whose independent variable is time and whose right hand side is given by the action of appropriate time dependent real matrices on the state vectors and/or on the control vectors.

Let p, m, n be positive integers, R^p be the p-dimensional real Euclidean space, t_0 be a real number and let set 1 be made of n actors and Set 2 be made of m actors.

Let us first formulate the rendez vous of the Set 2 actors with the Set 1 actors as an optimal control problem. The dynamics of the Set 1 actors is given by the following system of ordinary differential equations:

$$\frac{\mathrm{d}\underline{z}_i}{\mathrm{d}t} = G_i(t)\underline{z}_i(t),\ t > t_0,\quad i = 1, 2, \cdots, n, \tag{1}$$

with the initial condition:

$$\underline{z}_i(t_0) = \underline{z}_{0,i},\ i = 1, 2, \cdots, n, \tag{2}$$

where $\underline{z}_i(t) \in R^4$ is the state vector of actor i of Set 1 at time t, $G_i(t)$ is a 4×4 matrix that acts on the state vector $\underline{z}_i(t)$ and defines the dynamics of actor i of Set 1 at time t, $t > t_0$, finally $\underline{z}_{0,i}$ is the initial state of actor i of Set 1, that is the state of actor i of Set 1 at time $t = t_0$, $i = 1, 2, \cdots, n$.

The choice $\underline{z}_i(t) \in R^4$, $t > t_0$, $i = 1, 2, \cdots, n$, is motivated by the fact that in Section 3 a Set 1 actor will be a point mass moving in R^2 following Newton's dynamical principle. In this case we have $\underline{z}_i(t) \in R^4$, $t > t_0$, in fact the state vector of actor i at time t consists of two bi-dimensional vectors that represent respectively position and velocity at time t of the point mass moving in R^2, $t > t_0$, $i = 1, 2, \cdots, n$. That is R^4 is the state space of actor i of set 1, $i = 1, 2, \cdots, n$. The choice of R^p with $p \neq 4$ as state space is necessary to describe different situations, such as a point mass moving in R^3 or an actor represented by a dipole. The choice of R^p with $p \neq 4$ as state space can be done without changing substantially the analysis that follows.

Note that the trajectories of the Set 1 actors are implicitly defined by the system of differential Equations (1), with the initial condition (2) and that this initial value problem does not involve control variables. That is with the previous choices the Set 1 actors are not pursuing a goal conflicting with the goals of Set 2 actors, they are moving according to a preassigned dynamics expressed by Equations (1). The choice of Equations (1), (2) to express the dynamics of the Set 1 actors is only an example and can be changed in many ways leaving unchanged the substance of the mathematical model proposed.

The dynamics of the Set 2 actors is described by the following system of ordinary differential equations:

$$\frac{\mathrm{d}\underline{x}_j}{\mathrm{d}t} = A_j(t)\underline{x}_j(t) + B_j(t)\underline{u}_j(t), \quad t > t_0,\ j = 1,2,\ldots,m, \tag{3}$$

with the initial condition:

$$\underline{x}_j(t_0) = \underline{x}_{0,j},\ j = 1,2,\cdots,m, \tag{4}$$

where $\underline{x}_j(t) \in R^4$ is the state vector of actor j of Set 2 at time t, $\underline{u}_j(t) \in R^2$ is the control vector used to govern the state vector $\underline{x}_j(t)$ at time *t*, $A_j(t)$, $B_j(t)$ are matrices of the appropriate dimensions acting respectively on the state and on the control vectors $\underline{x}_j(t)$ and $\underline{u}_j(t)$ that define the dynamics at time t, $t > t_0$, of actor j of Set 2 and $\underline{x}_{0,j}$ is the initial state of actor j of Set 2, $j = 1,2,\cdots,m$. Note that the control vector $\underline{u}_j(t)$, $t > t_0$, $j = 1,2\cdots,m$, must be chosen in a class of admissible controls, this choice is made in order to achieve a preassigned goal, that in our case will be the "rendez vous" with the preys. The assumptions $\underline{x}_j(t) \in R^4$, $\underline{u}_j(t) \in R^2$, $t > t_0$, $j = 1,2,\cdots,m$, are motivated by the fact that in Section 3 a Set 2 actor will be a point mass moving in R^2 following Newton's dynamical principle with state vector belonging to R^4, and that the control vector acting on it will be interpreted as a force acting on the point mass, that is as a vector belonging to R^2. More general situations can be considered without changing substantially the content of this section.

In this section, for the sake of simplicity, we have assumed that the state vectors $\underline{x}_j$, $j = 1,2,\cdots,m$, and $\underline{z}_i$, $i = 1,2,\cdots,n$, have the same dimension and that for $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$ the rendez vous of actor i of Set 1 with actor j of Set 2 consists in having at a certain time during the game the two actors (approximately) in the "same state".

In Section 3, the actors are point masses subject to Newton's dynamical principle and for $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$, the rendez vous of actor i of Set 1 with actor j of Set 2 consists in having at a certain time during the game the two actors (approximately) in the "same position". Note that in Section 3 there is no condition on the velocities of the actors at the time of the rendez vous. That is the rendez vous of Section 3 can be seen as a clash between a set 1 actor and a Set 2 actor. It is easy to see that the two rendez vous definitions used in this section and in the Section 3 define two different video games. There are many other different ways of defining a rendez vous and as a consequence of defining video games.

Let $T > t_0$ be given, T is the time horizon of the rendez vous problem. That is, the rendez vous must take place (or not take place) in the time interval $[t_0, T]$. This means that the game starts at time $t = t_0$ and ends at time $t = T$.

We consider the following cost functional:

$$J(\underline{u}_1, \underline{u}_2, \cdots, \underline{u}_m) = \sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\gamma_{i,j}}{2}\int_{t_0}^{T}\mathrm{d}t U_2(\underline{x}_j(t), z_i(t)) + \sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\alpha_{i,j}}{2}U_1(\underline{x}_j(T), \underline{z}_i(T)) + \sum_{j=1}^{m}\frac{\beta_j}{2}\int_{t_0}^{T}\mathrm{d}t \left\|\underline{u}_j(t)\right\|^2, \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm of vector $\cdot$, $U_1(\cdot,\circ)$ and $U_2(\cdot,\circ)$ are suitable nonnegative real functions defined in $R^4 \times R^4$ (*i.e.* utility functions) whose minimization guarantees the rendez vous of $\cdot$ with $\circ$, and $\alpha_{i,j}, \beta_j, \gamma_{i,j}, i = 1,2,\cdots,n,\ j = 1,2,\cdots,m$, are nonnegative constants. A simple way of choosing U_1 and U_2 that guarantees the rendez vous is: $U_1(\cdot,\circ) = U_2(\cdot,\circ) = \|\cdot - \circ\|^2$.

The control problem that models the rendez vous of the Set 2 actors with the Set 1 actors when the latter do not pursue a goal conflicting with the goals of Set 2 actors can be stated as follows:

$$\begin{aligned} &\text{minimize} \quad J(\underline{u}_1, \underline{u}_2, \cdots, \underline{u}_m), \\ &\underline{u}_j \in V,\ j = 1,2,\cdots,m \end{aligned} \tag{6}$$

subject to the constraints (1), (2), (3), (4), where V is a suitable set of admissible controls.

Note that for $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$, the choice of the constants $\alpha_{i,j}$, $\gamma_{i,j}$ determines the assignment of the predators to the preys for the capture. In fact when $(\alpha_{i,j}, \gamma_{i,j})$ is different from (0,0) the predator j is assigned to the capture of the prey i and when $(\alpha_{i,j}, \gamma_{i,j})$ is equal to (0,0) the predator j does not consider the prey i as one of his targets. For $j = 1,2,\cdots,m$ the constant β_j measures how many "resources" available to the predator j (*i.e.* the control vector $\underline{u}_j(t)$) can be spent to chase the prey or the preys assigned to it.

The choice of the constants $\alpha_{i,j}, \gamma_{i,j}, \beta_j, i = 1,2,\cdots,n,$ $j = 1,2,\cdots,m$, of the functions U_1, U_2, of the cost functional (5) and of the constraints (1), (2), (3), (4) determines the behaviour of the predators, that is, defines the "artificial intelligence" (AI) of the predators.

Moreover, when the functions U_1, U_2 and the constants $\alpha_{i,j}$, $\gamma_{i,j}$, $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$, are chosen appropriately making small the first and the third set of addenda of (5) corresponds to trying to realize the rendez vous, in fact this implies that the distance (in state space) between the actors of Set 1 and the actors of Set 2 measured by the functions U_1 and U_2 is small.

Note that when we choose $U_1(\cdot,\circ)=U_2(\cdot,\circ)=\|\cdot-\circ\|^2$ the first set of addenda of (5) is the sum of the squares of the distances in state space at time $t=T$ between the couples of Set 1 and Set 2 actors times some nonnegative constants while the third set of addenda of (5) is the sum of the mean values in the time interval $[t_0,T]$ of the square of the distances in state space between the couples of Set 1 and Set 2 actors times some nonnegative constants. The cost of the rendez vous operations is represented by the second set of addenda of (5), that, up to nonnegative multiplicative constants, is the sum of the mean value in the time interval $[t_0,T]$ of the squared norm of the control vectors. Making small the second set of addenda of (5) during the rendez vous operation means that the goal of achieving the rendez vous is pursued keeping the "consumption" of the control vectors small. Finally the values chosen for the nonnegative constants $\alpha_{i,j}$, β_j, $\gamma_{i,j}$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$, that appear in (5) determine the choice of the relative weights of the three sets of addenda of (5). This corresponds to the choice of the relative weights given to the goals represented by these three sets of addenda of the cost functional.

Given the fact that the Set 1 actors follow the trajectory implicitly defined by the initial value problem (1), (2), Problems (6), (1), (2), (3), (4), is a formulation of a "rendez vous" problem as an optimal control problem. In this rendez vous problem the Set 1 actors follow their trajectories defined by (1), (2) ignoring the presence and the action of Set 2 actors. Set 2 actors pursue both the goals of having a rendez vous with the Set 1 actors and of keeping the values of the control vectors low (they indicate the consumption of resources employed for the rendez-vous).

Let us consider the situation where a differential game is used as model of the rendez vous problem. We generalize the optimal control Problems (6), (1), (2), (3), (4) in a differential game problem. First we change the definition of the dynamics of Set 1 actors and we define the new dynamics of Set 1 actors as follows:

$$\frac{\mathrm{d}\underline{z}_i}{\mathrm{d}t}=G_i(t)\underline{z}_i(t)+D_i(t)\underline{v}_i(t),\qquad t>t_0, i=1,2,\cdots,n, \tag{7}$$

$$\underline{z}_i(t_0)=\underline{z}_{0,i}, i=1,2,\cdots,n, \tag{8}$$

where $\underline{v}_i(t)\in R^2$ is the control vector relative to the state vector $\underline{z}_i(t)\in R^4$ at time t and $D_i(t)$ is a matrix of the appropriate dimensions that acting on the control vector $\underline{v}_i(t)$ contributes to the definition of the dynamics of the state vector $\underline{z}_i(t)$ at time t, $t>t_0$, $i=1,2,\cdots,n$. The choice $\underline{v}_i(t)\in R^2$, $t>t_0$, $i=1,2,\cdots,n$, has the same motivation than the analogous choice made for the Set 2 actors.

The dynamics of the Set 2 actors remains unchanged and is described by Equations (3), (4).

Let $t_0, T, \alpha_{i,j}, \beta_j, \gamma_{i,j}, i=1,2\cdots,n,\ j=1,2,\cdots,m$, be as above and β_i^*, $i=1,2,\cdots,n$, be nonnegative constants, we consider the following cost functional:

$$\begin{aligned}&J_1(\underline{u}_1,\underline{u}_2,\cdots,\underline{u}_m,\underline{v}_1,\underline{v}_2,\cdots,\underline{v}_n)\\&=\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\alpha_{i,j}}{2}U_1(\underline{x}_j(T),\underline{z}_i(T))+\sum_{j=1}^{m}\frac{\beta_j}{2}\int_{t_0}^{T}\mathrm{d}t\|\underline{u}_j(t)\|^2\\&-\sum_{i=1}^{n}\frac{\beta_i^*}{2}\int_{t_0}^{T}\mathrm{d}t\|\underline{v}_i(t)\|^2+\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\gamma_{i,j}}{2}\int_{t_0}^{T}\mathrm{d}tU_2(\underline{x}_j(t),\underline{z}_i(t))\end{aligned} \tag{9}$$

The differential game considered is the following:

$$\begin{gathered}\text{minimize}\quad \text{maximize}\quad J_1(\underline{u}_1,\underline{u}_2,\cdots,\underline{u}_m,\underline{v}_1,\underline{v}_2,\cdots,\underline{v}_n)\\ \underline{u}_j\in V,\qquad \underline{v}_i\in W,\\ i=1,2,\cdots,n,\ j=1,2,\cdots,m,\end{gathered} \tag{10}$$

subject to the constraints (3), (4), (7), (8), where W is a suitable set of admissible controls and V has been introduced previously.

Note that the Set 1 actors maximize the functional J_1 with respect to the control variables $\underline{v}_i$, $i=1,2,\cdots,n$, while the Set 2 actors minimize the functional J_1 with respect to the control variables $\underline{u}_j$, $j=1,2,\cdots,m$. When the constants $\alpha_{i,j},\gamma_{i,j},\beta_i^*,\beta_j, i=1,2,\cdots,n,\ j=1,2,\cdots,m$, and the functions U_1, U_2 satisfy some obvious conditions this implies that the Set 1 actors try to make large their distance (in state space) from the Set 2 actors making large the set of addenda of (9) containing the expressions $U_1(\underline{x}_j(T),\underline{z}_i(T))$ and the set of addenda containing the expressions $\int_{t_0}^{T}\mathrm{d}tU_2(\underline{x}_j(t),z_i(t))$, $i=1,2,\cdots,n, j=1,2,\cdots,m$, that is the Set 1 actors try to avoid the rendez vous with the Set 2 actors. Moreover, the Set 1 actors pursue the goal of avoiding the rendez vous with the Set 2 actors using a "small quantity" of the control variables $\underline{v}_i(t)$, $i=1,2,\cdots,n$, and this last objective is expressed by the attempt of maximizing the set of addenda of (9) containing the expressions $-\frac{\beta_i^*}{2}\int_{t_0}^{T}\mathrm{d}t\|\underline{v}_i(t)\|^2$, $i=1,2,\cdots,n$. That is with the choice of J_1 given in (9) the strategy pursued by the Set 1 actors that maximize J_1 is an "escape" strategy (*i.e.* the attempt of avoiding the rendez vous) that takes into account the need of keeping the "consumption" of the $\underline{v}_i$, $i=1,2,\cdots,n$, control variables small. On the other hand, the Set 2 actors minimize the functional J_1 with respect to the control variables $\underline{u}_j$, $j=1,2,\cdots,m$, which implies that the Set 2 actors try to realize the rendez vous keeping small, with respect to the variables $\underline{u}_j$, $j=1,2,\cdots,m$, the sets of addenda of (9) that contain the expressions $U_1\left(\underline{x}_j\left(T\right),\underline{z}_i\left(T\right)\right)$ and the expressions $\int_{t_0}^{T}\mathrm{d}tU_2\left(\underline{x}_j\left(t\right),\underline{z}_i\left(t\right)\right)$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$. Finally, keeping small the term $\sum_{j=1}^{m}\frac{\beta_j}{2}\int_{t_0}^{T}\mathrm{d}t\left\|\underline{u}_j\left(t\right)\right\|^2$ of (9) the Set 2 actors try to achieve the goal of having a rendez vous with the Set 1 actors using a "small quantity" of the control variables $\underline{u}_j\left(t\right)$, $j=1,2,\cdots,m$. Problems (10), (3), (4), (7), (8) is a formulation of a "rendez vous" problem as a differential game.

Note that when the Set 1 actors pursue not only the goal of avoiding the rendez vous (escape strategy) using a "small" quantity of the control vectors, but also the goal of reaching a location $\underline{\xi}_0$ in the state space we must add to (9) the terms $-\sum_{i=1}^{n}\frac{\delta_{1,i}}{2}\int_{t_0}^{T}\left\|\underline{z}_i\left(t\right)-\underline{\xi}_0\right\|^2\mathrm{d}t$ and/or $-\sum_{i=1}^{n}\frac{\delta_{2,i}}{2}\left\|\underline{z}_i\left(T\right)-\underline{\xi}_0\right\|^2$, where $\delta_{1,i}$, $\delta_{2,i}$, $i=1,2,\cdots,n$, are positive constants or more general terms (that is, for example, terms involving the functions U_1 and U_2) able to measure the "distance" between $\underline{z}_i$ and $\underline{\xi}_0$, $i=1,2,\cdots,n$. Note that, using the terms defined in the Introduction, $\underline{\xi}_0$ represents the home of the preys. The i-th actor reaches the location $\underline{\xi}_0$ when the distance between the state vector of actor i and $\underline{\xi}_0$ is smaller than a given tolerance, $i=1,2,\cdots,n$.

In this last case, the cost functional (9) can be rewritten as:

$$\begin{aligned}
&J_2\left(\underline{u}_1,\underline{u}_2,\cdots,\underline{u}_m,\underline{v}_1,\underline{v}_2,\cdots,\underline{v}_n\right)\\
&=\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\alpha_{i,j}}{2}U_1\left(\underline{x}_j\left(T\right),\underline{z}_i\left(T\right)\right)\\
&-\sum_{i=1}^{n}\frac{\delta_{2,i}}{2}U_1\left(\underline{z}_i\left(T\right),\underline{\xi}_0\right)\\
&+\sum_{j=1}^{m}\frac{\beta_j}{2}\int_{t_0}^{T}\mathrm{d}t\left\|\underline{u}_j\left(t\right)\right\|^2-\sum_{i=1}^{n}\frac{\beta_i^*}{2}\int_{t_0}^{T}\mathrm{d}t\left\|\underline{v}_i\left(t\right)\right\|^2\\
&+\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\gamma_{i,j}}{2}\int_{t_0}^{T}\mathrm{d}tU_2\left(\underline{x}_j\left(t\right),\underline{z}_i\left(t\right)\right)\\
&-\sum_{i=1}^{n}\frac{\delta_{1,i}}{2}\int_{t_0}^{T}\mathrm{d}tU_2\left(\underline{z}_i\left(t\right),\underline{\xi}_0\right),
\end{aligned}\tag{11}$$

where the constants $\alpha_{i,j},\beta_j,\beta_i^*,\delta_{1,i},\delta_{2,i},\gamma_{i,j}$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$, satisfy the conditions stated previously. The differential game considered is:

$$\begin{aligned}
&\text{minimize}\quad\text{maximize}\quad J_1\left(\underline{u}_1,\underline{u}_2,\cdots,\underline{u}_m,\underline{v}_1,\underline{v}_2,\cdots,\underline{v}_n\right)\\
&\quad\underline{u}_j\in V,\qquad\underline{v}_i\in W,\\
&i=1,2,\cdots,n,\quad j=1,2,\cdots,m,
\end{aligned}\tag{12}$$

subject to the constraints (3), (4), (7), (8), where W and V have been defined previously.

Note that Equations (3), (4) are the constraints of the optimization problem (6) and that similarly Equations (3), (4), (7), (8) are the constraints of the optimization problems (10) and (12).

Sometimes the cost functionals J, J_1, J_2 given respectively by (5), (9), (11) are called objective or utility functionals.

Different choices of the functionals J, J_1 and J_2 obtained, for example, by changing the functions U_1 and U_2, changing the values of the constants $\alpha_{i,j},\beta_j,\beta_i^*,\gamma_{i,j}$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$, and of the constants $\delta_{1,i}$, $\delta_{2,i}$, $i=1,2,\cdots,n$, or a different choice of the dynamical equations obtained by changing the matrices $A_j\left(t\right)$, $B_j\left(t\right)$, $j=1,2,\cdots,m$, $G_i\left(t\right)$, $D_i\left(t\right)$, $i=1,2,\cdots,n$, result in different games. In particular changing the values of the constants $\alpha_{i,j},\beta_j,\beta_i^*,\gamma_{i,j}$, $\delta_{1,i},\delta_{2,i}$, $i=1,2,\cdots,n$, $j=1,2,\cdots,m$, assigns different weights to the goals pursued by the actors of the game and, as a consequence, results in different behaviours of the actors during the game. Recall that the strategies adopted by the game actors are (approximate) solutions of the mathematical models adopted for the "rendez vous" problem. That is, they are the (approximate) solutions of the optimal control problems or of the differential games presented previously.

We can conclude that the different mathematical problems obtained by changing the cost functionals and/or the dynamical equations and/or the rendez vous definition lead to different video games, that have the same goals/rules, but are characterized by a different mixtures of parameters. From the point of view of the player, some of these video games are easy and some of them are difficult to play.

In Section 5, given the functions U_1 and U_2, we discuss in a special case how to change the parameter values that appear in the cost functional and in the dynamic equations to change the effectiveness of the actors in pursuing their goals.

Recall that, in order to be able to play with the video games derived from these mathematical models, it is necessary to be able to compute their solution (or their approximate solution) in closed loop form. In fact, "real time" reaction is needed when playing a video game.

It is important to notice that when we choose $U_1\left(\cdot,\circ\right)=U_2\left(\cdot,\circ\right)=\left\|\cdot-\circ\right\|^2$ the optimal control problems (6), (1), (2), (3), (4) and the differential games (10), (3), (4), (7), (8) or (12), (3), (4), (7), (8) due to their special form (*i.e.*: linear constraints and quadratic cost functional) can be approached analytically and numerically with elementary tools. A closed loop solution of these problems can be easily computed. That is, the Hamilton Jacobi equations satisfied by the value function associated to these models can be reduced to a matrix Riccati equation and this last equation can be solved numerically at an affordable computational cost. However the video game presented in Section 3 and Section 4 does not contain the solution of this matrix Riccati equation. In fact, for simplicity, we prefer finding easy-to-compute suboptimal solutions of the mathematical problems mentioned above, that "define" video game actors acting according to the goals stated in the game declaration.

3. A Heuristic Solution of the "Rendez Vous" Problem

Let us specify the mathematical models presented in Section 2 to the case of the video game that will be presented in detail in Section 4 and let us explain the heuristic meaning of the suboptimal solution of the models implemented in the software. The case presented in the video game of Section 4 can be summarized as follows: in a region Ω of the Euclidean plane there are two predators and one prey. The physical boundaries of the region Ω limit the space where the action of the game takes place. In order to simplify the exposition we choose the region Ω equal to the Euclidean plane, so that there are no boundaries to consider. At the end of this Section, we discuss briefly what to do when Ω has nonempty boundaries. The predators want to capture (*i.e.* to have a rendez vous with) the prey. The prey wants to escape the capture and wants to reach (with velocity not necessarily zero) its home. The prey home is a point belonging to the region Ω marked on the computer screen with a cross. The prey and the predators are the actors of the game. Each actor (prey or predator) is represented as a point mass that moves in the region Ω to pursue its goals. The movement of the point masses satisfies suitable equations of motion expressed by ordinary differential equations. The equations of motion express Newton's dynamical principle for the point masses (actors) considered. The control variables are forces acting on the appropriate point masses.

Let us introduce some notation:

- $t_0 = 0$ initial time;
- $T > t_0$ time horizon of the rendez vous problem;
- $m_x > 0$ mass of the prey;
- $m_j > 0$ mass of the j-th predator, $j = 1,2$;
- $\underline{x}^h(t) = \left(x_1^h(t), x_2^h(t)\right)^{\mathrm{T}} \in R^2$ prey position (*i.e.*: position of the point mass animated by the human player) at time t, $t_0 \le t \le T$. Note that the superscript T means transposed and should not be confused with the time horizon of the rendez vous problem also denoted with T and that the superscript h denotes the actor animated by the "human" player;
- $\underline{y}_j(t) = \left(y_{1,j}(t), y_{2,j}(t)\right)^T \in R^2$, position of the j-th predator at time t, $t_0 \le t \le T$, $j = 1,2$;
- $\underline{x}_0^h = \left(x_{1,0}^h, x_{2,0}^h\right)^T \in R^2$ initial position of the prey;
- $\underline{y}_j^0 = \left(y_{1,j}^0, y_{2,j}^0\right)^T \in R^2$ initial position of the j-th predator, $j = 1,2$;
- $\underline{\dot{x}}_0^h = \left(\dot{x}_{1,0}^h, \dot{x}_{2,0}^h\right)^T \in R^2$ initial velocity of the prey;
- $\underline{\dot{y}}_j^0 = \left(\dot{y}_{1,j}^0, \dot{y}_{2,j}^0\right)^T \in R^2$ initial velocity of the *j*-th predator, $j = 1,2$;
- $\alpha_x \ge 0$ friction coefficient of the prey;
- $\alpha_j \ge 0$ friction coefficient of the j-th predator, $j = 1,2$;
- $K_j \ge 0$ elastic constant of the force attracting the j-th predator to the prey, $j = 1,2$;

The dynamics of the game actors (7), (8), (3), (4) in the special case considered here is defined by the following system of ordinary differential equations (*i.e.*: equations of motion) and initial conditions:

$$m_x \frac{\mathrm{d}^2 \underline{x}^h}{\mathrm{d}t^2} + \alpha_x \frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t} = \underline{v}(t),\ t_0 < t < T, \tag{13}$$

$$\underline{x}^h(t_0) = \underline{x}_0^h, \tag{14}$$

$$\frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t}(t_0) = \underline{\dot{x}}_0^h, \tag{15}$$

$$m_j \frac{\mathrm{d}^2 \underline{y}_j}{\mathrm{d}t^2} + \alpha_j \frac{\mathrm{d}\underline{y}_j}{\mathrm{d}t} = \underline{u}_j(t), \quad t_0 < t < T,\ j = 1,2, \tag{16}$$

$$\underline{y}_j(t_0) = \underline{y}_j^0,\ j = 1,2, \tag{17}$$

$$\frac{\mathrm{d}\underline{y}_j}{\mathrm{d}t}(t_0) = \underline{\dot{y}}_j^0,\ j = 1,2, \tag{18}$$

where the control $\underline{v}(t)$, $t_0 < t < T$, is chosen by the human player using the joypad, and the control functions $\underline{u}_j(t)$, $t_0 < t < T$, $j = 1,2$, are chosen by the game engine. These last control functions will be chosen later using a simple idea based on elastic forces. Note that rewriting Equations (13) and (16) as systems of first order equations transforms them in the forms (7), (3). These equations are simple expressions of Newton's dynamical principle of classical mechanics. In fact they are simple examples of the physical law $m\underline{a} = \underline{F}$ where m is the mass, $\underline{a}$ is the acceleration of the point mass considered and $\underline{F}$ is the force acting on it. That is, for example, Equation (13) can be interpreted as follows:

given the fact that the prey is a point mass of mass m_x and that $\underline{x}^h(t)$ is the prey position at time t, the term $m_x \frac{\mathrm{d}^2 \underline{x}^h}{\mathrm{d}t^2}$ represents the mass of the prey multiplied by its acceleration and the force acting on the prey is given by the vector $\underline{F} = \underline{v} - \alpha_x \frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t}$, where $-\alpha_x \frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t}$ is a friction force and the control variable $\underline{v}$ is a force chosen by the human player using the joypad. For $j = 1,2$, the predator j is a point mass of mass m_j and Equation (16) is the equation of motion of predator j. It is easy to see that Equation (16) can be interpreted as Equation (13).

The control function $\underline{v}(t)$, $t_0 < t < T$, is chosen by the human player having in mind the prey goals. Looking at the computer screen where the game scene is displayed in real time the human player chooses this control function acting on a joypad.

The control functions $\underline{u}_j(t)$, $t_0 \le t \le T$, $j = 1,2$, appearing in (16) are chosen "solving" the mathematical model of the rendez vous problem and this choice defines the behaviour of the predators, that is defines the artificial intelligence of the predators. For example let us consider the functional:

$$J_3(\underline{u}_1, \underline{u}_2) = \sum_{j=1}^{2} \frac{\nu_j}{2} \int_{t_0}^{T} \left\| \underline{u}_j(t) \right\|^2 \mathrm{d}t + \sum_{j=1}^{2} \frac{\mu_j}{2} \int_{t_0}^{T} U_3\left(\underline{y}_j(t), \underline{x}^h(t) \right) \mathrm{d}t, \tag{19}$$

where ν_j, μ_j, $j = 1,2$, are positive constants. The functional J_3 defined in (19) can be seen as a variation of the functional J defined in (5). The constants ν_j, μ_j, $j = 1,2$, play the same role as the constants β_j, $j = 1,2,\cdots,m$, and $\gamma_{i,j}$, $i = 1,2,\cdots,n$, $j = 1,2,\cdots,m$, appearing in (5) and the function U_3 depends on the distance between the two position vectors and its minimum value is reached when this distance is zero. The simplest choice of U_3 is $U_3(\cdot,\circ) = \|\circ - \cdot\|^2$. The fact that U_3 is chosen as a function of the position vectors of the actors and not of the state vectors (made of position and velocity) is a consequence of the interpretation of the rendez vous as "approximately in the same position" adopted in this section, instead of the interpretation of the rendez vous as "approximately in the same state" adopted in Section 2.

The control problem that we consider in the video game as mathematical model of the rendez vous problem can be written as follows:

$$\underset{\underline{u}^1, \underline{u}^2 \in S}{\text{minimize}} \quad J_3\left(\underline{u}^1, \underline{u}^2\right), \tag{20}$$

subject to the constraints (13), (14), (15), (16), (17), (18), where S is a suitable set of admissible controls. Note that, as already said, for the purposes of the video game we need a closed loop solution of the control problems (20), (13), (14), (15), (16), (17), (18), in fact for $t_0 \le t \le T$ when at time t we must choose the control variables $\underline{u}_j(t)$, $j = 1,2$, the control variable $\underline{v}(\tau)$ has been chosen (by the human player) only for $t_0 \le \tau \le t$, and the corresponding vector $\underline{x}(\tau)$ is known (to the game engine) only for $0 < \tau < t$.

A suboptimal solution of the control problems (20), (13), (14), (15), (16), (17), (18) can be found solving the following system of differential equations and initial conditions:

$$m_x \frac{\mathrm{d}^2 \underline{x}^h}{\mathrm{d}t} + \alpha_x \frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t} = \underline{v}, \quad t_0 < t < T, \tag{21}$$

$$\underline{x}^h(t_0) = \underline{x}_0^h, \tag{22}$$

$$\frac{\mathrm{d}\underline{x}^h}{\mathrm{d}t}(t_0) = \underline{\dot{x}}_0^h, \tag{23}$$

$$m_j \frac{\mathrm{d}^2 \underline{y}_j}{\mathrm{d}t^2} + \alpha_j \frac{\mathrm{d}\underline{y}_j}{\mathrm{d}t} = -K_j\left(\underline{y}_j - \underline{x}^h\right), \quad t_0 < t < T, \quad j = 1,2, \tag{24}$$

$$\underline{y}_j(t_0) = \underline{y}_j^0, \quad j = 1,2, \tag{25}$$

$$\frac{\mathrm{d}\underline{y}_j}{\mathrm{d}t}(t_0) = \underline{\dot{y}}_j^0, \quad j = 1,2, \tag{26}$$

where K_j, $j = 1,2$ are positive constants. Note that Equations (21), (22), (23) are simply Equations (13)-(15). These equations are repeated here to simplify the exposition. This corresponds to choose $\underline{u}_j(t) = -K_j\left(\underline{y}_j - \underline{x}^h\right)$, $j = 1,2$, as suboptimal solution of (20), (13), (14), (15), (16), (17), (18). The force $\underline{u}_j(t) = -K_j\left(\underline{y}_j - \underline{x}^h\right)$ is the elastic force generated by a string of elastic constant K_j that pushes $\underline{y}_j$ toward $\underline{x}^h$, $j = 1,2$. The previous choice of the control functions $\underline{u}_j(t)$, $t_0 \le t < T$, $j = 1,2$, defines predators that behave coherently with their goal, that is predators that try to have a rendez vous with the prey. The solution $\left(\underline{x}^h(t), \underline{y}_1(t), \underline{y}_2(t)\right)$, $t_0 \le t \le T$, of problems (21),(22), (23), (24), (25), (26) is determined once chosen the control function $\underline{v}(t)$, $t_0 \le t \le T$. This solution describes a scene where for $j = 1,2$, and $t_0 \le t \le T$ the predator j located at time t in $\underline{y}_j(t)$ tries to make small its distance from the prey while the prey located at time t in $\underline{x}^h(t)$ tries using the force $\underline{v}$ chosen by the human player, to make small its distance from the home represented as a cross on the screen and to make large its distance from the predators. The human player pursues the prey goals looking on the computer screen the game scene displayed in real time and choosing consequently the control function $\underline{v}(t)$, $t_0 < t < T$, acting on the joypad. In fact the trajectory implicitly defined by (21), (22), (23) is determined by the human player choosing $\underline{v}$. The trajectories implicitly defined by (24), (25), (26) are determined by the choice $\underline{u}_j = -K_j\left(\underline{y}_j - \underline{x}^h\right)$, $j = 1, 2$, made in (24) of the control variable $\underline{u}_j$ appearing in (16). This choice of the control variables $\underline{u}_j$, $j = 1, 2$, guarantees that the forces acting on the predators try to reduce the distance between the predators and the prey. That is the choice made of the control variables defines a suboptimal solution of problems (20), (13), (14), (15), (16), (17), (18) such that the behaviour of the game actors is coherent with their goals in the game. Moreover this suboptimal solution of problems (20), (13), (14), (15), (16), (17), (18) can be computed as a closed loop suboptimal solution through explicit formulae. In particular for the prey position, solution of (21), (22), (23), the following formula holds:

$$\begin{aligned}\underline{x}^h(t) &= \underline{x}_0^h + \frac{m_x}{\alpha_x}\underline{\dot{x}}_0^h + \frac{1}{\alpha_x}\int_{t_0}^{t}\underline{v}(\tau)\,\mathrm{d}\tau \\ &\quad - \mathrm{e}^{-\alpha_x(t-t_0)/m_x}\left(\frac{m_x}{\alpha_x}\underline{\dot{x}}_0^h + \frac{1}{\alpha_x}\int_{t_0}^{t}\underline{v}(\tau)\,\mathrm{e}^{\alpha_x(\tau-t_0)/m_x}\,\mathrm{d}\tau\right), \\ &t_0 \le t \le T.\end{aligned} \tag{27}$$

Similar formulae hold for the predator positions, solution of (24), (25), (26).

Note that for $j = 1, 2$, increasing the value of the elastic constant K_j, or decreasing the value of the mass m_j correspond to increasing the effectiveness of the predator j in chasing the prey.

Finally let us note that the force $\underline{v}(t)$, $t_0 < t < T$, chosen by the human player using the joypad and the forces $\underline{u}_j = -K_j\left(\underline{y}_j - \underline{x}^h\right)$, $t_0 < t < T$, $j = 1, 2$, can be seen as a suboptimal solution of the differential game defined by:

$$\begin{aligned}&\text{minimize} \quad \text{maximize} \quad J_4\left(\underline{u}^1, \underline{u}^2, \underline{v}\right), \\ &\underline{u}^1, \underline{u}^2 \in U \quad \underline{v} \in W\end{aligned} \tag{28}$$

subject to the constraints (13), (14), (15), (16), (17), (18) where:

$$\begin{aligned}&J_4\left(\underline{u}_1, \underline{u}_2, \underline{v}\right) \\ &= \sum_{j=1}^{2}\frac{\nu_j}{2}\int_{t_0}^{T}\left\|\underline{u}_j(t)\right\|^2\mathrm{d}t + \sum_{j=1}^{2}\frac{\mu_j}{2}\int_{t_0}^{T}U_3\left(\underline{y}_j(t), \underline{x}^h(t)\right)\mathrm{d}t \\ &-\frac{\delta_1}{2}\int_{t_0}^{T}\left\|\underline{\Lambda}_0 - \underline{x}^h(t)\right\|^2\mathrm{d}t - \frac{\beta}{2}\int_{t_0}^{T}\left\|\underline{v}(t)\right\|^2\mathrm{d}t,\end{aligned} \tag{29}$$

where $\underline{\Lambda}_0 = \left(\Lambda_{1,0}, \Lambda_{2,0}\right)^T$ is the location of the prey home in the Euclidean plane and ν_j, μ_j, $j = 1, 2$, δ_1, β are positive constants. Moreover this suboptimal solution of problems (28), (13), (14), (15), (16), (17), (18) can be computed using elementary formulae in closed loop form. This means that for $0 = t_0 < t < T$ the suboptimal solution of time t can be computed using only information available at time t.

In a future work instead of building a video game starting from a suboptimal solutions of a mathematical model such as (20), (13), (14), (15), (16), (17), (18) or (28), (13), (14), (15), (16), (17), (18) as we have done here, we will consider a video game based on the closed loop solution of the mathematical model going through the reduction of the corresponding Hamilton Jacobi equations to a matrix Riccati equation and the numerical solution of this last equation.

We conclude this Section discussing briefly the case when Ω, instead of being the two dimensional Euclidean plane, is a proper subset of the Euclidean plane. When Ω is a subset of the plane with non empty boundary in order to keep the trajectories of the game actors inside Ω we add, to the already considered forces, also a force generated by a strong short range repulsive potential infinite on the boundary of Ω. In this case it is necessary to integrate numerically the resulting equations of motion of the game actors. In fact, the presence of the force generated by the repulsive potential inhibits the possibility of writing explicit formulae for the solutions of the equations of motions.

4. The Video Game

4.1. Game Description

The video game presented here consists of two predators chasing a prey animated by a player using a joypad. The prey to save his life and to increase his point score must reach its home, that is a target represented with a cross on the computer screen and must avoid the predators. The game field Ω is the entire plane and the prey is always on the computer screen, since the screen window automatically scrolls up, down, left, right according with the movement of the prey. The video game can be downloaded from the website: http://www.ceri.uniroma1.it/ceri/zirilli/w9.

The prey is a point mass represented as a black disk on the computer screen and its movement is implicitly defined by the equations of motion and initial conditions (21), (22), (23). The equation of motion (21) depends on the force $\underline{v}$ chosen by the human player acting on the joypad. The predators are point masses represented as red disks on the computer screen and their dynamics is described by the equations of motion and initial conditions (24), (25), (26). **Figure 1** shows the game window, with

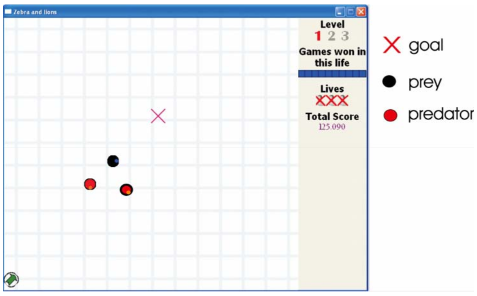

Figure 1. A scene taken from the video game.

the prey, the predators (black and red disks) and the prey home (a red cross).

The human player to move the prey uses the washer on the left of the joypad choosing the force that will determine the movement of prey (*i.e.* this corresponds to choose the control function $\underline{v}$ of Section 3). The movement of the (left) washer to the right provides a force that pushes the prey to the right on the screen, the movement of the washer up provides a force that pushes the prey upwards on the screen and so on. The force magnitude is directly proportional to the washer slope.

For all practical purposes the time horizon of the game is infinite, that is in the actual implementation of the video game T has been chosen as a large positive constant. The game stops when the prey reaches its home (victory of the prey) or when the predators reach the prey (victory of the predators). The human player (the prey) can escape from the predators indefinitely without being penalized, for example, he can escape until he presses the "back" button stopping the game execution. The game is made of three levels of increasing difficulty. The predators of the three levels feature an increasing effectiveness in chasing the prey.

The effectiveness of the predators is controlled by changing the values of some of the constants in the motion equations presented in Section 3. In particular, decreasing the value of the mass parameter increases the effectiveness of the corresponding predator in chasing the prey.

The values of these constants change at each level of the game. The choice of the value of the time step used in the computation of the trajectories is done through a calibration procedure that takes into account the hardware where the video game is running. The values of the constants that increase or decrease the effectiveness of the predators in chasing the prey (*i.e,* for example, the elastic constant, or the mass and the friction coefficient of the predators) have been chosen calibrating the difficulty of each level of the game on the ability of a sample of five experienced players aged between six and thirteen years.

To move to the next level the human player must reach the target (*i.e.* its home) ten consecutive times without being intercepted by the predators. Each time the human player reaches the target, his point score increases by an amount (1-2-3 points) equal to the level where he is and a counter computes the number of consecutive times the player has reached the target. When the player reaches the target, a graphical window appears on the screen stating that the target has been reached. When the player closes the message window, a new target appears on the screen, prey and predators are repositioned, and the game starts again. The repositioning of the prey and of the predators on the screen is random and is obtained using a random number generator. If the predators reach the prey three consecutive times the game ends with the message "game over" and player's final score is displayed. When the player moves to the next level of the game receives a bonus of 1000 points. In the third level of the game the player can play indefinitely.

4.2. Software Implementation of the Game

The software implementation of the video-game consists of three parts: 1) the joypad interface (XboxGamepads.dll) that handles the communications between the joypad and the game engine; 2) the game logic software that computes the displacement of the prey and of the predators; 3) the graphical interface (XboxGame.dll) that handles the communications between the game engine and the computer screen.

The two interfaces are implemented as dll codes and are written in Delphi. Delphi is a programming language that can be seen as an evolution of the Pascal language. The game software is implemented in C++ code.

1) Joypad interface: XboxGamepads.dll

This interface interprets commands coming from the Xbox 360 joypad via a USB port and converts the hexadecimal data transmitted by the joypad into suitable records containing computer logical variables (true/false) or double precision numbers.

The used joypad buttons are "start" (game start), "end" (game end), "green" (game pause) and the top left washer.

The signal transmitted by the top left washer key is interpreted as an ordered couple of real numbers α_1, α_2 belonging to the interval $[-1,1]$. The vector $(\alpha_1,\alpha_2)^T$ coming from this washer is the force that the human player provides to the prey. To make it easy for the human player to understand the correspondence between his action on the washer and the force provided to the prey, the vector $(\alpha_1,\alpha_2)^T$ points in the direction of the tilted washer projected on the plane of the computer screen when the joypad is kept "parallel to the computer screen" with the washer on the side of the human player sitting in front of the computer screen and the norm of the vector $(\alpha_1,\alpha_2)^T$ is proportional to the slope of the tilted washer.

In Section 3 this force is denoted with $\underline{v} = (v_1, v_2)^T$, where v_1, and v_2 are the Cartesian components of the two dimensional vector representing the force. By appropriately choosing the reference frames we have: $\underline{v} = (v_1, v_2)^T = (\alpha_1, \alpha_2)^T$.

2) Game logic software—C++ code game_body_def.cpp

The game logic software serves two purposes in the game. The first purpose is the initialization of the game, which consists in assigning the values of the game parameters, opening the communications between the engine and the joypad and calibrating some parameters depending on the CPU of the computer that runs the game. The second purpose is to check the input of the buttons, in particular of the end game button, to handle the data acquisition from the joypad washer in order to compute the new positions of the prey and of the predators and to compute the new positions of the prey and of the predators.

The computation of these positions is done for the prey using the data relative to the force transmitted by the joypad and Formula (27), and for the predators using formulae analogous to Formula (27) derived from Equations (24), (25), (26). As already said, the time interval used in the computation of the solution is calibrated on the hardware that hosts the video game. This calibration is done when the video game starts. Indeed, on the hardware commonly available today, the time required to compute the new positions from the data available is negligible in comparison to the reaction time of the player and the calibration chooses a time interval such that the movements on the computer screen of the game actors are compatible with an effective human intervention in the game.

3) Graphical interface: XboxGame.dll

The dll XboxGame graphic interface allows displaying the scene of the game on the screen every time that the engine communicates new data.

5. Conclusions

In this work, a class of video games has been presented that relies on the solution of optimal control problems and differential games. We have shown how ideas taken from classical mechanics and elementary calculus can be used to build a video game. In fact, the mathematical models employed and the methods used to solve the resulting problems are able to simulate the intelligent behaviour of the predators, thus can be used to implement a simple AI form.

The AI form implemented in the game presented Section 4 is very simple and can be enhanced in many different ways. For example, it is possible to use adaptive choices of the masses m_j and of the elastic constants K_j, $j = 1, 2$, appearing in the suboptimal solution presented in Section 3. That is, the elastic constants and the masses of the predators can be adapted to the ability of the human player. This kind of adaptivity is motivated by the need of making the game enjoyable for players having different skill levels.

As mentioned in the Introduction, we have done an experiment on a sample of a few dozens of young players. This experiment shows that, after some matches, children aged between five and eight years do understand the presence of restrictions to their movements and to the movements of their opponents due to Newton dynamical principle. An interesting finding is that even the youngest players after few attempts do develop an intuitive comprehension of the physical laws and consequently are able to improve their movements and their ability to escape from the predators. The same experiment shows that teenager players get easily bored by the video game. That is the video game is too simple for teenagers players. For example the graphic and the environment of the game is too simple for players already acquinted with commercial videogames. These players may be interested in the game as a virtual laboratory where to do experiments about Newton's dynamical principle in a recreational environment.

The previously considered differential game models allow to develop many different video games. For example teams of actors moving in three dimensional space can be considered. Moreover, the rendez vous definition can be changed taking into account both positions and velocities instead of only positions and other forces (e.g. gravity) could be added to the physical constraints. Finally the ability of the automatic characters can be calibrated on the ability of the human players.

In conclusion the idea of simulating intelligent behaviour using models taken from mathematical analysis and elementary physics that has been exploited in this paper is a rather general idea that can be used in many circumstances and that deserves further investigation.

REFERENCES

[1] F. Bellotti, R. Berta and A. De Gloria, "Designing Effective Serious Games: Opportunities and Challenges for Research," *Special Issue*: *Creative Learning with Serious Games*, *Journal of Emerging Technologies in Learning*, Vol. 5, No. 3, 2010, pp. 22-35.

[2] F. Bellotti, R. Berta, A. De Gloria and L. Primavera, "Adaptive Experience Engine for Serious Games," *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 1, No. 4, 2009, pp. 264-280. doi:10.1109/TCIAIG.2009.2035923

[3] M. Athans, "On Optimal Allocation and Guidance Laws for Linear Interception and Rendez Vous Problems,"

IEEE Transactions on Aerospace and Electronics Systems, Vol. 7, No. 5, 1971, pp. 843-853. doi:10.1109/TAES.1971.310324

[4] R. Isaacs, "Differential Games," Dover Publication, New York, 1999.

[5] A. A. Melikyan, "Primary Strategies of Simple Pursuit in Differential Games on Two-Sided Plane Figures," *Journal of Applied Mathematics and Mechanics*, Vol. 68, No. 4, 2004, pp. 545-554. doi:10.1016/j.jappmathmech.2004.07.007

[6] V. Kokkeviv, "Practical Physics for Articulated Characters," *Game Developers Conference*, San Jose, 24-26 2004, pp. 1-16. http://www.red3d.com/cwr/games/#ai-papers

[7] I. Millington, "Game Physics Engine Developments," Elsevier Inc., San Francisco, 2007.

[8] S. I. Nishimura and T. Ikegami, "Emergence of Collective Strategies in a Prey-Predator Game Model," *Artificial Life*, Vol. 3, No. 4, 1997, pp. 243-260. doi:10.1162/artl.1997.3.4.243

[9] I. Millington, "Artificial Intelligence for Games," Morgan Kaufmann Publications, Elsevier, San Francisco, 2006.

[10] V. Y. Glizer and V. Turetsky, "Complete Solution of a Differential Game with Linear Dynamics and Bounded Controls," *Applied Mathematics Research Express*, Vol. 2008, No. 1, 2008, 49 p. doi:10.1093/amrx/abm012