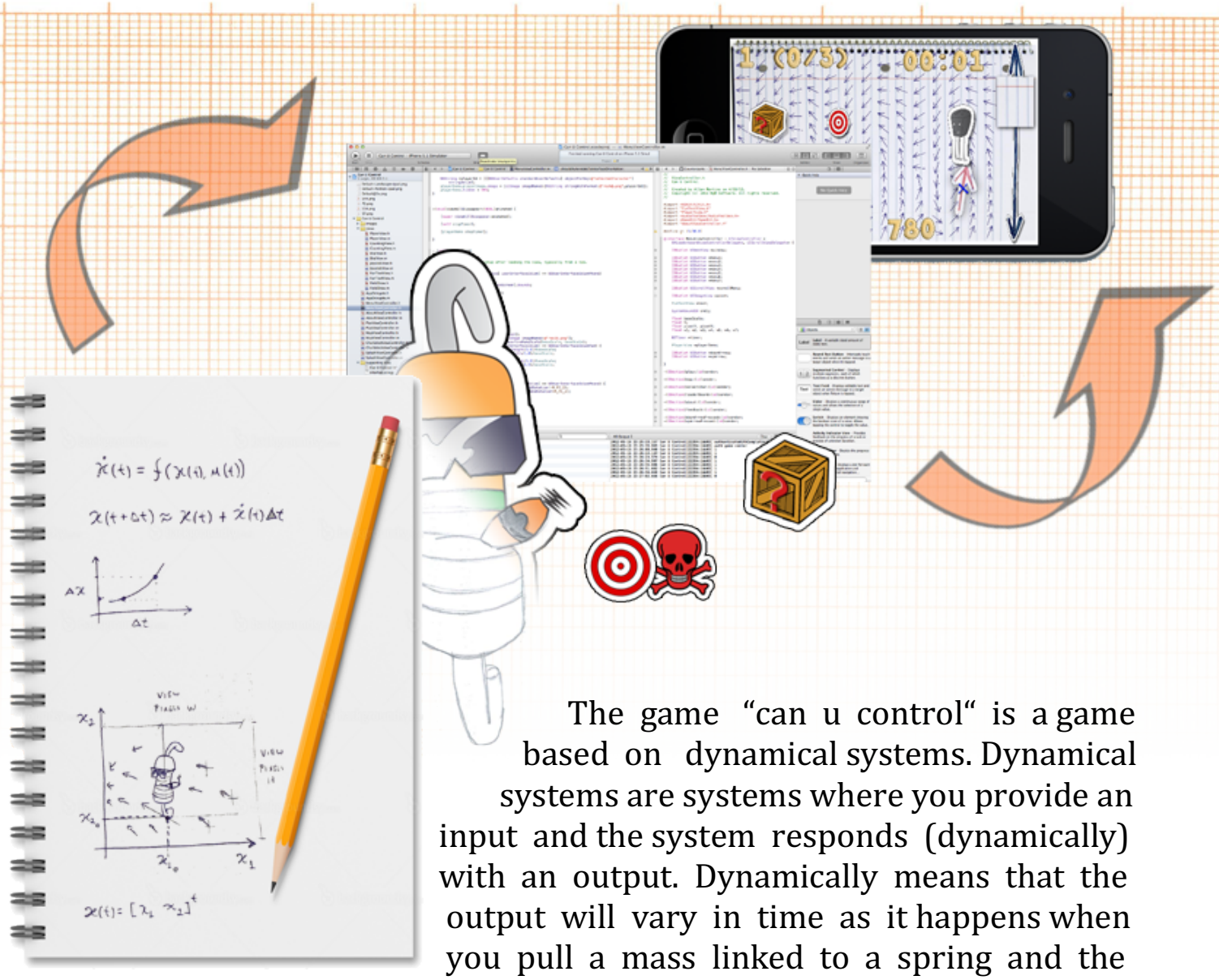

The game “can u control“ is a game based on dynamical systems. Dynamical systems are systems where you provide an input and the system responds (dynamically) with an output. Dynamically means that the output will vary in time as it happens when you pull a mass linked to a spring and the mass goes back and forth swinging because of the spring. A kind of dynamical systems that is very common are electrical circuits (so the use of all electronic characters in the game) but the concept applies to any system that have time (or any global variable) involved. There are several mathematical representations for dynamical systems. The most commonly used are differential equations, transfer functions and state space. In this game, we use the state space representation to give the dynamics of the game. In the state space representation, the system is represented by internal variables called “states” of the system. The states can be any variable that change in the system (like the position of a mass, current in a circuit, etc). The state space representation is nothing but a first order differential equation where the state variables are assembled into a unique vector representation, as shows in the following equation

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t), u(t)) \qquad (1)$$

In the equation, $\mathbf{x}(t)$ is the state vector, $u(t)$ is the input of the system and $F(.)$ is the function that governs how the system evolves in time. The dimension of the vector gives the order of the system. Systems of 2nd order for instance have two state variables. This equation simple tells us that at a particular moment in time, the state will "walk" in the direction given by its first derivative. For example, if F is a constant vector, say, pointing up (in a 2nd order system), the first state will always increase and the second will stay still.

If we have a 2nd order system, it is possible to "draw" the state space equations by drawing arrows representing each derivative at the positions corresponding to the state positions. This "plot" is called vector field and it looks like the following picture

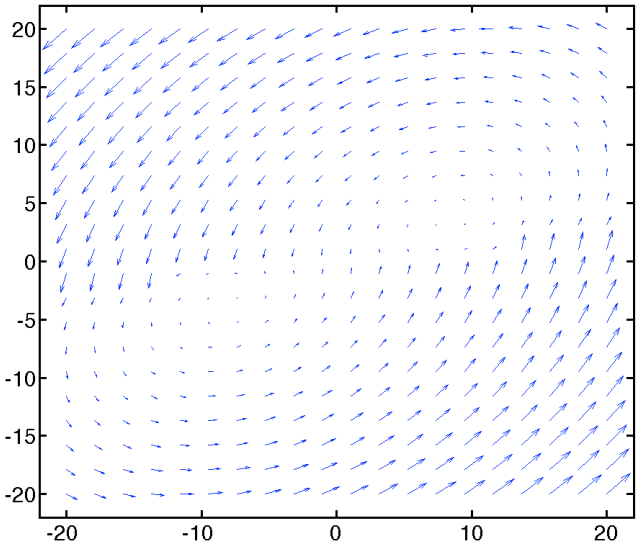

Figure 1: Example of the plot of a vector field. The arrows are the velocity that the state will have if it is located where the arrows is.

Additionally we can draw the state and make it "walk" in the vector field. As each arrow is the derivative of the state position, it represents the state velocity and as such, the state will "follow the arrows".

This representation, although graphical and simple, can mean and represent a lot of important stuff, from currents in a circuit to positions of atoms and its particles. The "walk" of the state can be controlled by changing the vector field (changing $u(t)$ in the equation (1)). The problem is that, by changing u, we do not change the vector field freely. The vector field will only change one aspect of it, either changing the

overall direction of the arrows or changing the location to where the arrows converge to. So, it is a challenge to control the position where we want the state to be in real systems. There are several kinds of systems (depending on the function $F(.)$). Systems can be linear, non-linear, discontinuous, quantized, discrete, etc. Each one can even be classified according to its stability (stable or instable), presence of oscillations, cycles, etc.

In order to simulate the state spare equations in a computer (in this case the device you are playing the game), it is necessary to numerically "integrate" the states. That will be how the character will "walk" in the vector field. To do that, there are several methods in the literature that can be used. In the case of the game, we used the Euler Method because it is simple and suited to gaming purposes (where precision is not that important). The Euler's method is simple, it basically starts from the approximation we use to define the derivative

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t+\Delta t)-\mathbf{x}(t)}{\Delta t} \tag{2}$$

Here, Δt is the step in time that we want to integrate. The smaller this step, the more precise is the result, but slower the solution is computed (more points to compute). Now we just plug the equation 1 and rewrite the "next" state as a function of the "previous" states as

$$\begin{aligned} \mathbf{x}(t+\Delta t) &= \mathbf{x}(t)+\dot{\mathbf{x}}(t)\Delta t \\ \mathbf{x}(t+\Delta t) &= \mathbf{x}(t)+\mathbf{F}(\mathbf{x}(t),u(t))\Delta t \end{aligned} \tag{3}$$

So, as $\mathbf{x}(t)$ is the position of the character, the only thing we can do to change its movement is to increase or decrease $u(t)$. In this manner, playing the game, you will be able to "feel" how difficult (or not) it is to control the state of a dynamical system only by using the "control" input $u(t)$.

The game then consists in stages or levels, each one represented by a different system vector field or dynamics. There is a little character representing the state position and it only moves following the vector field (tied to the dynamics of the system). Your goal is to change the

input of the system (sliding up or down a joystick) in order to change the vector field, guiding the character to a target.

In the game, there are several “lands“ each one representing a kind of system (linear, non-linear, etc). Each land has its degree of difficult due to the nature of the dynamics of the system it represents. For example, there are levels that present a characteristic called limit cycles, some other are instable and so on.

With all this, the game is not only fun (I hope) but teaches us a little about state space representation and how difficult (or easy) it is to control a dynamical system using only one variable! Have fun with it and do not forget to go back to the lands after finishing them to try the new challenges (moving targets, skulls, etc).