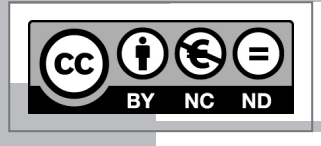
The Transborder Immigrant Tool

La herramienta transfronteriza para inmigrantes

2014 The Office of Net Assessment
The University of Michigan
Ann Arbor, Michigan



978-0-9915719-1-8

ACKNOWLEDGMENTS

The *Transborder Immigrant Tool* (TBT) by Electronic Disturbance Theater 2.0/b.a.n.g. lab is a last mile safety device designed to aid the disoriented of any nationality in a desert environment. The project's interactive platform was developed and tested in southern California's Anza-Borrego Desert State Park from 2009 to 2012. Its code, included in this volume, is executable when and if one adds the coordinates of functional water caches to its poetic program. Its poetry, another executable code included here, comprises TBT's desert survival series.

Electronic Disturbance Theater 2.0/b.a.n.g. lab: Micha Cárdenas, Amy Sara Carroll, Ricardo Dominguez, Elle Mehrmand, and Brett Stalbaum

Computer code: Walkingtools Laboratory (Brett Stalbaum and Jason Najarro)

Statement: Amy Sara Carroll

Statement translation: Natasha Hakimi Zapata

Poetry: Amy Sara Carroll

Poetry translations: Francheska Alers-Rojas, Julieta Aranda, Elizabeth Barrios, Iván Chaar-López, Orquídea Morales, Omar Pimienta, and Mary Renda

Book design: Parisa Ghaderi

ONA series coordinators: John Cheney-Lippold and Tung-Hui Hu

Mil gracias to the members of the University of Michigan's Rackham graduate student and faculty interdisciplinary workshop—the Border Collective—for their unflagging commitment to collaboration, to Paula Poole for her desert and design wisdoms, to Water Station and Border Angels for their humanitarian efforts, to John Cheney-Lippold, Parisa Ghaderi, and Wendy Sung for their critically creative coordination of this volume's publication, and to Bonnie Applebeet, Larry La Fountain-Stokes, and Kristy Rawson for their Spanish and English proofreading.

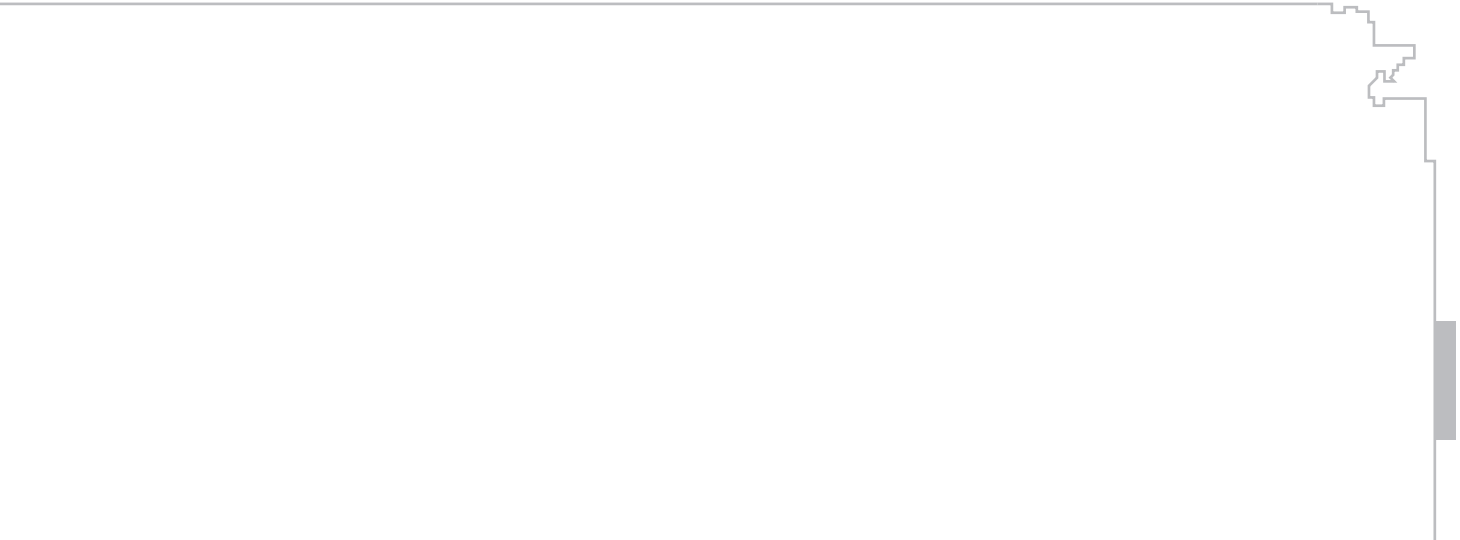


TABLE OF CONTENTS

1 Of Ecopoetics and Dislocative Media

5 De la ecopoética y los medios dislocativos

11 CoordinatesContainer.java

13 DowsingCompass.java

16 DowsingCompassListener.java

17 TBCoordinates.java

20 TBGpxParser.java

26 TBMIDlet.java

44 The Desert Survival Series/La serie de sobrevivencia del desierto

iv. Quién hablaría de la soledad del desierto
Raúl Zurita, "El Desierto de Atacama VII"

In *Exodus/Éxodo*, Ciudad Juárez photographer Julián Cardona writes of a large tree just outside of Sásabe, Sonora, where border-crossers discard their old clothes to put on new ones that make them look more "American."[1] In conversation with an undocumented worker in North Carolina, Cardona realized that he and the other man both had stopped beneath that same literal and symbolic marker of crossing. Cardona's reflections on the pair's exchange remind me of another commentary on trees, embedded in Leonard Schwartz's heartfelt observations on the great Chilean writer Raúl Zurita's "visionary poetics." Schwartz muses,

> Zurita's poems might be figured as an eco-poetry in which the space between nature and history is closed up, once we realize that the work reimagines the entirety of the ocean in such a way as to include those thrown from planes into that ocean. And reimagines the mountains in such a way as to include the Disappeared thrown from planes into their snows until one can only speak of those mountains as containing those people. And renders the desert no longer conceivable except if the voices and the deaths in the desert are made a part of that desert.[2]

Schwartz goes on to parse his references to the former Pinochet regime's practice of wedding natural environs to its methods of dealing with political dissent and that practice's effects on Zurita's poetics by way of an eco (echo). Schwartz recalls that in the introduction to the anthology *Black Nature: Four Centuries of African-American Nature Poetry*, Camille Dungy stakes the claim that African American poets have a distinct relationship to the conceit of nature.[3] In Dungy's formulation, a tree can never just be a tree, but also functions as a rememory

of lynching. Ecology holds trauma and promise simultaneously, is neither beautiful nor sublime per se, but becomes part of a larger built environment that regulates the policing and disciplining of ungrammatical bodies.[4]

A swath of border art and literature similarly has documented how the Sonoran Desert—on the southwestern border of the United States, on the northwestern border of Mexico—equals more than a desert, especially after 1994. Chantal Akerman's early high-definition, multilingual *De l'autre côté/From the Other Side*, for instance, bears witness to the fallout effects of the years 1994 to 2001 in the region and beyond. Formally, the documentary depends in equal measure upon testimony and upon a series of high-lonesome tracking shots.[5] *De l'autre côté* begins in Agua Prieta, Sonora, with border dwellers' firsthand accounts of losing relatives, of attempting to enter the United States. Akerman's focus subsequently shifts to neighboring Douglas, Arizona. The filmmaker interviews a local restaurant owner, a married couple on their ranch, who reflect on their border fears after 9/11 (2001), a Mexican consulate, and a sheriff, who explains the on-the-ground effects of the Clinton administration's Operation Gatekeeper.

Although all of the above segments of *De l'autre côté* powerfully tug on viewers' sympathies and antipathies, particularly in the wake of SB1070, another tactic of the film directly related to Operation Gatekeeper's reconfiguration of border wall-fence ecologies, lends even higher definition to Akerman's talking cure. Operation Gatekeeper, implemented in 1994, profoundly shifted Mexico-to-U.S. migrants' border-crossing patterns. The policy displaced would-be undocumented entrants from the Light-Up-the-Border campaign stretches of Baja California-California to the alternately treacherous, increasingly narco-driven, intemperate zones of Sonora-Arizona's "devil's highway." Akerman's film undocuments the affective ironies generated by this displacement's "privileged unknowing" production of the desert as border

patrol.[6] The filmmaker's sustained attentions to place, her relentless, interminable scrutiny of landscapes and long-winded "silences," in the style of her previous "images within images," in *De l'autre côté* are "arresting."[7] (They literally arrest the film's development, eco-ing another layer of meaning to the word "arrest.") *De l'autre côté* slows down to magnify, to amplify the Agua Prieta-Douglas join of the border's own alleged historical "emptiness" (in chronodevelopmentalist time) into a cacophonous roar. Or, more accurately, the film's double portraiture of inhabitants and landscapes dialectically indexes the divide between the border as "discursive vector" and "just weather."

In this regard, although seldom reckoned with in border, Latin@, and inter-American studies, *De l'autre côté* forms part of a repertoire of cultural production that personifies to politicize the Sonoran Desert. In works that range from Cardona's images of "brassieres hanging off bushes near Sásabe on the line at a place where guides are known to rape women coming north" to Luis Alberto Urrea's *The Devil's Highway: A True Story*, the desert environment becomes an active agent in border enforcement, in informal literal and symbolic economies of trafficking, in effect contributing to the proliferating NAFTA era connotations of the word disposability in the Mexican-U.S. borderlands.[8]

From this position I have pondered and grown into the form of the *Transborder Immigrant Tool* (TBT) as an aesthetic intervention, too. When I began collaborating on TBT, I found myself writing poems that functioned best in museum, gallery, and university contexts. I wondered, "What would I want from a poem in the desert? Would I want a poem there at all?" I turned to my collaborator Brett Stalbaum, who'd co-authored the project's code, with these questions, or rather with the intuition that the poems that I'd composed weren't quite right for the project—not yet. We began to discuss the hubris of the poetics attached to TBT, to ponder what kind/s of poetry could

and would engage or expand upon our collective vision of the tool as sustenance. Our conversations were connected to our ruminations' echolocation of the tool as dislocative media. We wondered and wandered perpetually: What would/does it mean to strip locative media of its implicit urbanity?

I returned to the drawing board, approaching all of the above as a conceptual or formal conundrum. In other words, I connected the "poetry problem" to the ambitions of the project's code, the specificities of the latter's remediation of a GPS platform. Meanwhile, I began to spend more time in the desert. I tested a prototype of the tool against the landscape, against my own disorientation and technological "impairments." Beneath the unforgiving glare of the landscape's daytime sun, beneath the luminosity of its nighttime stars, I thought, "What would I absolutely need to know if I were on my way to one of TBT's way-stations?" I turned to texts about desert survival: handbooks, military manuals, a guide for border-crossers briefly distributed by the Mexican government. I divvied up (with the help of Brett and others) the information I found—the indispensible from the expendable. I wrote pared down prose poems, ideologically neutral (Is any writing "ideologically neutral"?), procedural, if you will—a poem about locating the North Star, a poem about what to do if you are bitten by a rattlesnake or tarantula, poems that contained details about how to weather a sandstorm or a flash flood. I imagined one poem for every hour of the day, a series of twenty-four fragments in conversation with one another, in conversation with the previous museum-gallery-university series of poems that I'd composed.

At base, I worked from two assumptions. A desert is not just a desert. And, poetry-becoming-code/code-becoming-poetry could transubstantiate, translate into a lifesaving technology, sounding off.

[1] Charles Bowden and Julián Cardona, *Exodus/Éxodo* (Austin: University of Texas Press, 2008), 268.

[2] Leonard Schwartz, "Zurita: On Raul Zurita's Visionary Poetics," *Jacket2*, last modified January 15, 2012, accessed July 25, 2012, http://jacket2.org/commentary/zurita.

[3] Camille T. Dungy, ed., *Black Nature: Four Centuries of African-American Nature Poetry* (Athens: University of Georgia Press, 2009).

[4] Like many, when I think of grammar and the ungrammatical in this context, I return to Hortense J. Spillers, "Mama's Baby, Papa's Maybe: An American Grammar Book," *Diacritics* 17.2 (Summer 1987): 64-81.

[5] *De l'autre côté/From the Other Side*, directed by Chantal Akerman (Paris: ARTE France, 2002).

[6] I modify here Eve Kosofsky Sedgwick's phrase "privilege of unknowing." Sedgwick, "Privilege of Unknowing: Diderot's *The Nun" in Tendencies* (Durham: Duke University Press, 1993), 23-25.

[7] Terrie Sultan, ed., *Chantal Akerman: Moving through Time and Space*. (Houston and New York: Blaffer Gallery, the Art Museum of the University of Houston, 2008).

[8] Bowden and Cardona, *Exodus/Éxodo*, 223; and Luis Alberto Urrea, *The Devil's Highway: A True Story* (New York: Little, Brown, and Company, 2004).

DE LA ECOPOÉTICA Y LOS MEDIOS DISLOCATIVOS

iv. Quién hablaría de la soledad del desierto
Raúl Zurita, "El Desierto de Atacama VII"

En *Éxodo/Exodus*, el fotógrafo Julián Cardona de Ciudad Juárez escribe sobre un gran árbol a las afueras de Sásabe, Sonora, bajo el cual quienes cruzan la frontera descartan su vieja ropa y se ponen ropa nueva que los hace parecer más "americanos".[1] En conversación con un trabajador indocumentado en Carolina del Norte, Cardona se da cuenta que tanto él como el otro hombre habían parado bajo el mismo marcador literal y simbólico del cruzamiento. Las reflexiones de Cardona sobre el intercambio me recuerdan a otro comentario sobre árboles, incrustado en las observaciones sinceras de Leonard Schwartz sobre la poética visionaria del gran escritor chileno Raúl Zurita. Schwartz reflexiona,

> Los poemas de Zurita pueden ser entendidos como una eco-poesía en el que el espacio entre la naturaleza y la historia está cerrado, en cuanto nos damos cuenta de que la obra reimagina la totalidad del océano de tal manera que incluye aquellos que fueron tirados de aviones para caer en él. Y reimagina las montañas de tal manera que incluyen a los Desaparecidos tirados de aviones para caer en sus nieves hasta que uno tan solo puede hablar de esas montañas de manera de que contengan a esa gente. Y hace que el desierto no sea concebible sin que las voces y las muertes que ocurrieron en el desierto formen parte del mismo.[2]

Schwartz después analiza las referencias a la práctica de unir el entorno natural con métodos para lidiar con la disidencia política que empleaba el antiguo régimen de Pinochet y los efectos que esa práctica tuvo en la poética de Zurita mediante un eco. Schwartz recuerda que en la introducción de la antología

Naturaleza negra: Cuatro siglos de poesía afroamericana sobre la naturaleza, Camille Dungy declara que los poetas afroamericanos tienen una relación distinta al recurso literario de la naturaleza.[3] En la formulación de Dungy, un árbol nunca puede ser simplemente un árbol, sino que también funciona como una rememoria del linchamiento. La ecología contiene el trauma y la promesa simultáneamente, no es ni bella ni sublime per se, pero se convierte en parte de un ambiente artificial más grande que regula la vigilancia y disciplina de "cuerpos gramaticalmente incorrectos".

Una gran cantidad de arte y literatura de la frontera de los Estados Unidos ha documentado cómo el desierto de Sonora—en la frontera suroeste de los Estados Unidos, en la frontera noroeste de México—equivale a más que un desierto, especialmente después de 1994. La obra multilingüe y de alta-definición de Chantal Akerman titulada *De l'autre côté/Del otro lado,* por ejemplo, atestigua a los efectos secundarios de los años 1994 a 2001 en la región y más allá de ella.[4] Formalmente, el documental depende en partes iguales sobre el testimonio y sobre una serie de viajes altamente solitarios. *De l'autre côté* comienza en Agua Prieta, Sonora, con relatos de primera mano de habitantes fronterizos que hablan sobre la pérdida de familiares y del intentar entrar a los Estados Unidos. El foco de Akerman se desplaza posteriormente al pueblo vecino de Douglas, Arizona. La cineasta entrevista al dueño de un restaurante local, a una pareja casada en su rancho, quienes reflexionan sobre sus temores fronterizos después del 9/11 (2001), y a un consulado mexicano y un alguacil quienes explican los efectos que tuvo la Operación Gatekeeper del gobierno de Clinton allí.

Aunque todos los segmentos mencionados de *De l'autre côté* llegan con fuerza a las simpatías y antipatías de los espectadores, especialmente a raíz de la ley SB1070, otra táctica de la obra directamente relacionada a la reconfiguración de las ecologías de la valla fronteriza de la Operación Gatekeeper, sin embargo, aporta una definición aun más alta a la cura del

habla que emplea Akerman. La Operación Gatekeeper, implementada en 1994, cambió de manera profunda los patrones que seguían los migrantes que cruzaban la frontera de México a EEUU. La póliza desplazó a los que vendrían a ser entrantes indocumentados de los trechos pertenecientes a la campaña Light-Up-the-Border de Baja California-California hacia las zonas también traicioneras, cada vez más narco-controladas y desaplacibles de la "carretera del diablo" en Sonora-Arizona. El documental de Akerman indocumenta las ironías afectivas de este desplazamiento generadas por la producción "privilegiada desconocedora"[5] del desierto como patrulla fronteriza. Las atenciones prolongadas de la cineasta hacia el lugar, su implacable, interminable escrutinio de los paisajes y "silencios" extensos, en el estilo de sus "imágenes dentro de imágenes" previas, en *De l'autre côté* son "detenedores".[6] (Literalmente detienen el desarrollo del documental, haciendo eco para añadir otro nivel de significado a la palabra "detener".) *De l'autre côté* desacelera para magnificar, para amplificar la unión de Agua Prieta-Douglas con el presunto "vacío" histórico de la frontera (en tiempo crono-desarrollista) y convertirlo en un rugido cacofónico. O, para ser más precisos, el doble-retrato de los habitantes y paisajes que logra el documental indexa dialécticamente la división entre la frontera como "vector discursivo" y "tan solo clima".

En este sentido, aunque rara vez se tome en cuenta en los estudios de la frontera, de los latinos, e interamericanos, *De l'autre côté* forma parte de un repertorio de producción cultural que personifica para politizar el desierto de Sonora. En obras que van desde las imágenes de Cardona de "sostenes colgando de arbustos cerca de Sásabe en la línea en un lugar donde se conoce que los guías violan a las mujeres viniendo al norte" a *La carretera del diablo: Una historia verídica*, el ambiente desértico se convierte en agente activo en la vigilancia fronteriza, en economías informalmente literales y simbólicas

del tráfico, en efecto, contribuyendo a la proliferación de las connotaciones de la palabra desechabilidad en la era del TLCAN en las zonas fronterizas de México-Estados Unidos.[7]

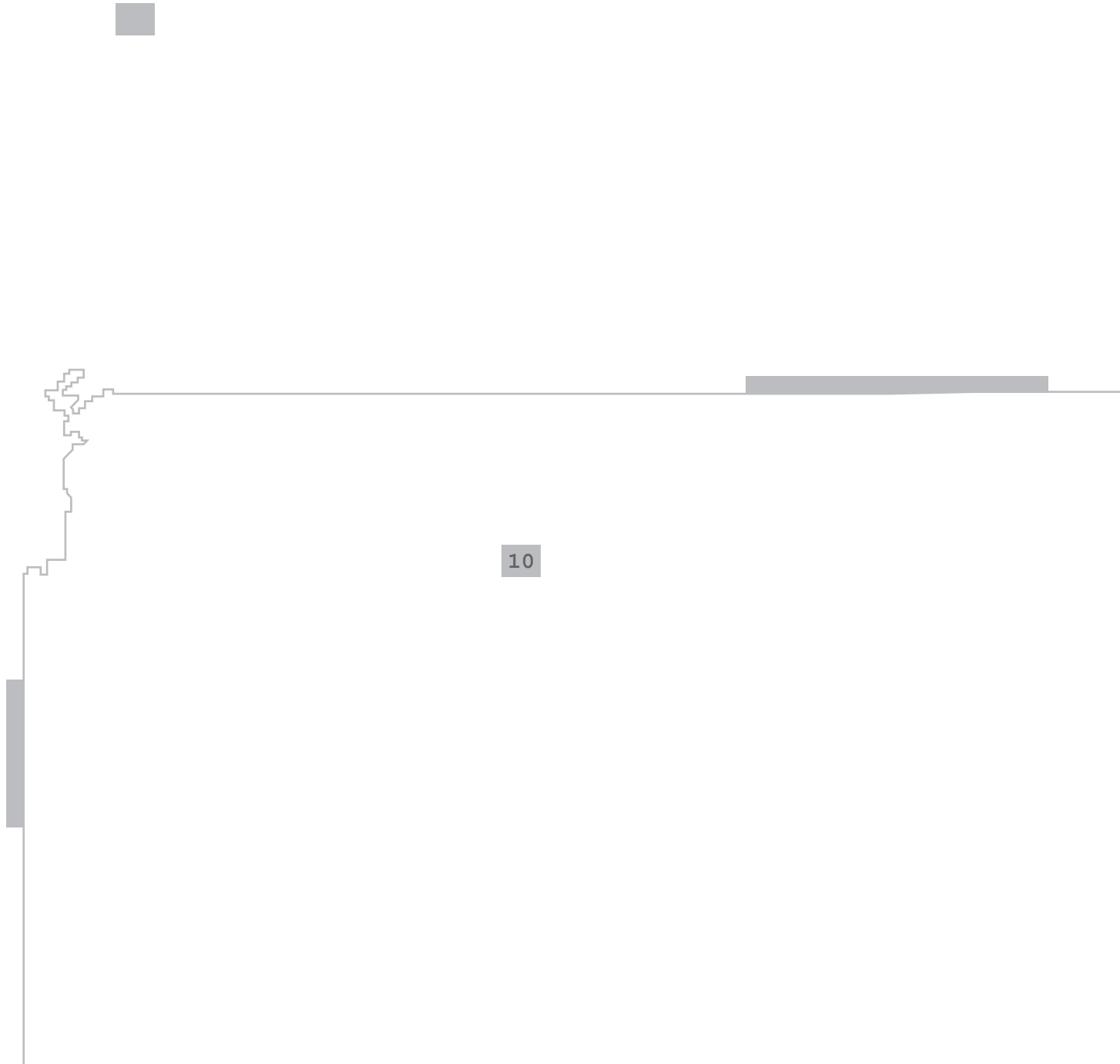
Desde esta posición, yo también he reflexionado y crecido para llegar a llenar la forma de *La herramienta transfronteriza para inmigrantes (*TBT) como una intervención estética. Cuando empecé a colaborar en TBT, me encontré escribiendo poemas que funcionaban mejor en un museo, una galería y en contextos universitarios. Pensé, "¿Qué querría yo de un poema en el desierto? ¿Querría, en lo más mínimo, un poema?" Me dirigí a mi colaborador Brett Stalbaum, quien co-escribió el código del proyecto, con estas preguntas, o más bien con la intuición de que los poemas que yo había compuesto todavía no eran adecuados para nuestro proyecto. Empezamos a hablar del hubris de la poética ligada al TBT, a reflexionar sobre que tipo/s de poesía podrían efectivamente expandir o entablar con nuestra visión colectiva de la herramienta como sustento. Nuestras conversaciones estaban conectadas a nuestras meditaciones y su ecolocación de la herramienta como medio dislocativo. Nos preguntábamos y deambulábamos perpetuamente: ¿Qué significa/ significaría despojar a los medios locativos de su urbanidad implícita?

Volví al punto de partida, abordando todo lo anterior como un enigma conceptual o formal. En otras palabras, conecté el "problema de poesía" a las ambiciones del código del proyecto, a las especificidades de remediación de este último como una plataforma GPS. Mientras tanto, empecé a pasar más tiempo en el desierto. Probé un prototipo de la herramienta contra el paisaje, en contra de mi propia desorientación e "impedimentos" tecnológicos.

Bajo el brillo implacable de su sol diurno, por debajo de la luminosidad de sus estrellas nocturnas, pensé: "Si yo estuviera en camino a una de las estaciones de paso de TBT, ¿qué sería absolutamente necesario saber?" Recurrí a textos sobre la

sobrevivencia en el desierto: guías, manuales militares, una guía para los que cruzan la frontera que brevemente fue distribuida por el gobierno mexicano. Repartí (con la ayuda de Brett y de otros) la información que encontré—lo indispensable de lo prescindible. Escribí poemas en prosa recortados, ideológicamente neutros (¿Alguna escritura es acaso "ideológicamente neutra"?), de procedimiento, por así decirlo—un poema sobre cómo localizar la estrella del norte, un poema sobre lo que se debe hacer en el caso de ser mordido por una serpiente de cascabel o una tarántula, poemas que contenían detalles sobre cómo capear una tormenta de arena o una inundación repentina. Imaginé un poema por cada hora del día, una serie de veinticuatro fragmentos en conversación el uno con el otro, en conversación con los poemas anteriores que había compuesto que pertenecían a un museo-galería-universidad.

En el fondo, trabajé a partir de dos suposiciones. Un desierto no es sólo un desierto. Y poesía-convirtiéndose-en-código/código-convirtiéndose-en-poesía podría transubstanciarse, traducirse en una tecnología que salva vidas, resonar.


10

CoordinatesContainer.java
available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/CoordinatesContainer.java


```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g. lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

package edu.ucsd.calit2.TransBorderTool;

/*
 * XML Parsing using kxml2 example by Naveen Balani
 */
import java.util.Vector;
import java.util.Hashtable;
import java.util.Enumeration;
import javax.microedition.location.Coordinates;

/** This class extends a Hashtable to add methods that can sort the data
 *  geographically or topologically to produce some notion of which of the
 *  "nearby" points might be "closest". Various methods are expected to be
 *  added to the class, which sort these differently.
 * @author Jason Najarro and Brett Stalbaum
 * @version 0.5.5
 */
public class CoordinatesContainer extends Hashtable {
```

```
/**
 * Default constructor
 */
public CoordinatesContainer() {
    super();
}

/** Gathers the TBCoordinates
 * @param currentPos the current position
 * @param range with in range meters
 * @return a Vector of TBCoordinates
 */
public Vector getNearestCoords(Coordinates currentPos, int range) {
    Vector nearestWaypoints = new Vector();

    // Get coordinates within range as Vector
    for (Enumeration e = this.elements(); e.hasMoreElements();) {
        TBCoordinates hCoords = (TBCoordinates) e.nextElement();
        float distance = currentPos.distance(hCoords);
        if (distance <= range) {
            nearestWaypoints.addElement(hCoords);
        }
    }

    // Sort the Vector by distance from current position
    int n = nearestWaypoints.size();
    for (int i = 1; i < n; i++) {
        for (int j = 0; j < n - i; j++) {
            float distance1 = currentPos.distance((TBCoordinates) nearestWaypoints.elementAt(j));
            float distance2 = currentPos.distance((TBCoordinates) nearestWaypoints.elementAt(j + 1));
            if (distance1 > distance2) {
                TBCoordinates temp = (TBCoordinates) nearestWaypoints.elementAt(j);
                nearestWaypoints.setElementAt(nearestWaypoints.elementAt(j + 1), j);
                nearestWaypoints.setElementAt(temp, j + 1);
            }
        }
    }

    //System.out.println("vector " + nearestWaypoints.size());
    return nearestWaypoints;
```

```
    }
}
```

13 **DowsingCompass.java**

available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/DrowsingCompass.java

```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g.lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

package edu.ucsd.calit2.TransBorderTool;

import edu.ucsd.calit2.TransBorderTool.international.*;
import java.util.Vector;
import javax.microedition.location.LocationProvider;
import javax.microedition.location.Location;
import net.walkingtools.javame.canvas.CompassCanvas;

/**
 * @author Brett Stalbaum and Jason Najarro
 * @version 0.5.5
 */
public class DowsingCompass extends CompassCanvas {

    //private DowsingCompassListener MIDlet = null;
    private CoordinatesContainer waypointList = null;
```

```
private Vector nearbyWPVector = null;
private boolean gotFirstFix = false;
private final byte TRIGGER_RADIUS = 30;
private DowsingCompassListener dowsingListener = null;

/**
 * Constructor for a DowsingCompass
 *
 * @param gpxFileName the GPX file name
 */
public DowsingCompass(String gpxFileName) {
    super(Translation.loadTranslation("en"));
    super.setTriggerRadius(TRIGGER_RADIUS);
    // simply making a GPXparser causes it to come to life an parse the file given.
    TBGpxParser gpxParser = new TBGpxParser(TBGpxParser.RES, "../../../../" + gpxFileName);
    // get the CoordinatesContainer (a Hashtable) from the parser - it should
    // contain all the the waypoints in the GPX file
    waypointList = gpxParser.getCoordsContainer();
    nearbyWPVector = new Vector();
}

public void setTarget(TBCoordinates targetCoords) {
    target = targetCoords;
    super.setTarget(target);
    navigating = true;
}

public void stopNavigation() {
    super.stopNavigation();
    navigating = false;
}

public Vector getNearbyWaypoints(int distance) {
    nearbyWPVector = waypointList.getNearestCoords(currentCoords, distance);
    return nearbyWPVector;
}

public void addNavigatorListener(DowsingCompassListener navigatorListener) {
    super.addNavigatorListener(navigatorListener);
```

```
        dowsingListener = navigatorListener;
    }

    public void removeNavigatorListener(DowsingCompassListener navigatorListener) {
        super.removeNavigatorListener(navigatorListener);
        dowsingListener = null;
    }

    // override CompassCanvas locationUpdated
    public void locationUpdated(LocationProvider provider, Location location) {
        // call to superclass
        super.locationUpdated(provider, location);
        if (gotFirstFix == false && location.isValid()) {
            gotFirstFix = true;
        } else if (!location.isValid()) {
            return;
        }

        if (!navigating && isMoving()) { // here we are dowsing, but only if moving
            //System.out.println("search waypoint list begun");
            for (int i = 0; i < nearbyWPVector.size(); i++) {
                TBCoordinates mc = (TBCoordinates) nearbyWPVector.elementAt(i);
                if (currentCoords.distance(mc) <= TRIGGER_RADIUS + 20) { // + a little
                    continue; // omits sites that we are already at
                }
                float directionPointerAzimuth = (int)currentCoords.azimuthTo(mc);
                //System.out.println(directionPointerAzimuth);
                if ((currentCompassHeading >= (directionPointerAzimuth - 5) &&
                        currentCompassHeading <= (directionPointerAzimuth + 5))) {
                    if (dowsingListener != null) {
                        dowsingListener.witchingEvent(mc);
                    }
                    break;
                }
            }
        }
    }
}
```

DowsingCompassListener.java
available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/DrowsingCompassListener.java

```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g.lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

package edu.ucsd.calit2.TransBorderTool;

import net.walkingtools.javame.canvas.NavigatorListener;

/**
 * This interface extends edu.ucsd.calit2.TransBorderTool.GPS.CompassListener
 * adding the witchingEvent method for the TransBorder tool
 * @author Brett Stalbaum
 * @version 0.5.5
 */
public interface DowsingCompassListener extends NavigatorListener {

      /**
       * Allows the implementing class to receive witching events
       * @param mc the MetaCoordinates object describing the witched site
       */
      public void witchingEvent(TBCoordinates mc);

}
```

TBCoordinates.java

available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/TBCoordinates.java

```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g. lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

package edu.ucsd.calit2.TransBorderTool;

import java.io.IOException;
import javax.microedition.location.Coordinates;
import javax.microedition.lcdui.Image;

/**
 * The TBCoordinates class extends javax.microedition.location.Coordinates
 * to include various MetaData
 * @author Brett Stalbaum and Jason Najarro
 * @version 0.5.5
 */
public class TBCoordinates extends Coordinates {

    public final int WATER = 0;
    public final int BEACON = 1;
    public final int CITY = 2;
    private String name = "";
```

```
private String type = "";

public TBCoordinates(double latitude, double longitude, float altitude, String name, String type) {
    super(latitude, longitude, altitude);
    this.name = name;
    this.type = type;
}

public TBCoordinates() {
    super(0, 0, 0);
    name = "";
    type = "";
}

// Convert Coordinates to TBCoordinates
//
/**
 * @param c
 * @return
 */
public static TBCoordinates toMetaCoords(Coordinates c, String name, String type) {
    TBCoordinates mc = new TBCoordinates(c.getLatitude(), c.getLongitude(),
      c.getAltitude(), name, type);
    return mc;
}

// Convert Coordinates to TBCoordinates
//
/**
 * @param c
 * @return
 */
public static TBCoordinates toMetaCoords(Coordinates c) {
    TBCoordinates mc = toMetaCoords(c, "", "");
    return mc;
}

public void setName(String n) {
    name = n;
}
```

```
    /** Set TBCoordinates type
     * e.g water station, safety beacon, etc.
     */
    public void setType(String t) {
        type = t;
    }

    public String getName() {
        return name;
    }

    public String getType() {
        return type;
    }

    /** Returns image pertaining to MetaCoordinate type
     */
    public Image getIcon() {
        Image icon = null;
        try {
            //System.out.println(this.getType());
            if (this.getType().toLowerCase().endsWith("water")) {
            // matching "Water", "Drinking Water", "N Water...."
                icon = Image.createImage("/img/water_icon_sm.png");
            } else if (this.getType().equalsIgnoreCase("beacon")) {
                icon = Image.createImage("/img/beacon_icon_sm.png");
            } else if (this.getType().equalsIgnoreCase("city")) {
                icon = Image.createImage("/img/generic_icon_sm.png");
            } else {
                icon = Image.createImage("/img/generic_icon_sm.png");
            }
        } catch (IOException ioe) {
            System.err.println(ioe);
        }
        return icon;
    }
}
```

TBGpxParser.java

available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/TBGpxParser.java

```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g. lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

package edu.ucsd.calit2.TransBorderTool;

/* KXML APIs - http://kobjects.org/kxml/index.php
 * USE of the Kxml packages require the distribution of the following license
 * Copyright (c) 2002-2007 Stefan Haustein, Oberhausen, Rhld., Germany
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or
 * sell copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
```

```
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
 * FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
 * IN THE SOFTWARE.
 */

import org.kxml2.io.KXmlParser;
import javax.microedition.io.HttpConnection;
import javax.microedition.io.Connector;
import java.io.InputStreamReader;
import java.io.InputStream;
import java.io.IOException;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
/**
 * Objects of this class serve the narrow
 * purpose of reading a .gpx file containing only waypoints. It was updated a great deal
 * as of version B3 to make it more robust-it should now handle any gpx file containing
 * waypoints, regardless of other junk in the file.
 * @author Jason Najarro and Brett Stalbaum
 * @version 0.5.5
 */
public class TBGpxParser implements Runnable {

    /**
     * Load from resource
     */
    public static final int RES = 0;
    /**
     * Load from HTTP connection
     */
    public static final int HTTP = 1;
    private int type;
    private String filePath;
    private CoordinatesContainer coordsList = null;
    // parse states
    boolean open = false;
    boolean waypointOpen = false;
    boolean gotName = false;
    boolean gotSym = false;
```

```
String sym = "";
String name = "";
double lat = -1;
double lon = -1;
private boolean loading = true;

/** This constructor accepts a source type (see the static final values of this class) and a
    String
 * representing the file path or URL to the source location.
 * @param sourceType as of version beta 3, HTTP or RES
 * @param sourceLocation a file path or URL
 */
public TBGpxParser(int sourceType, String sourceLocation) {
      coordsList = new CoordinatesContainer();
      type = sourceType;
      filePath = sourceLocation;
      Thread t = new Thread(this);
      t.start();
}


/** This constructor accepts a source type (see the static final values of this class), a
    String
 * representing the file path or URL to the source location, and a CoordinatesContainer if
 * you have one.
 * @param sourceType as of version beta 3, HTTP or RES
 * @param sourceLocation a file path or URL
 * @param container a CoordinatesContainer, left out, this class will produce its own.
 */
public TBGpxParser(int sourceType, String sourceLocation, CoordinatesContainer container) {
      type = sourceType;
      filePath = sourceLocation;
      coordsList=container;
      Thread t = new Thread(this);
      t.start();
}

/** returns true if objects of this class are busy loading the file
 * @return true if still loading
 */
```

```
public boolean loading() {
	return loading;
}

public void run(){
	try {
		KXmlParser parser = new KXmlParser();

		if(type==HTTP){
			//Open http connection
			HttpConnection httpConnection = (HttpConnection) Connector.open(filePath);

			//Initilialize XML parser
			parser.setInput(new InputStreamReader(httpConnection.openInputStream()));
		}else if(type==RES){
			InputStream GPXInStream = null;
			//System.out.println(filePath);
			GPXInStream = this.getClass().getResourceAsStream(filePath);
			//Initilialize XML parser
			parser.setInput(new InputStreamReader(GPXInStream));
		}
		//System.out.println("calling parse");
		parseGPXData(parser);

	}catch (Exception e) {
		//e.printStackTrace ();
		//System.out.println("Error:");
		//System.out.println(e.toString ());
		return;
	}
	loading = false;
}

/* Completely rebuilt this part on May 10, 2008, in order to make the class
* more robust. Before it assumed that certain elements would be contained in
* a particular order, without any intervening xml tags that might be added
* by various applications. (Such as G7towin, which produced gpx files that
* this class threw up on.) I tried to make it more robust by looking for tags
* and accepting their data if they contained "wpt" data, and just ignoring
* anything else it finds. Brett.
```

```
*/
private void parseGPXData(KXmlParser parser) throws IOException {
      try{
            parser.nextTag();
            if (parser.getEventType() != XmlPullParser.END_DOCUMENT) open = true;
            while (open) {
                  /*
                  // elegant testing!
                  int tabs = parser.getDepth();
                  StringBuffer buf = new StringBuffer();
                  for (int i = 0; i < tabs; i++ ) {
                        buf.append("\t*");
                  }
                  */
                  // parsing the Start tags has all the stuff we care about
                  if (parser.getEventType() == XmlPullParser.START_TAG) {
                        String currentName = parser.getName();
                        //System.out.println(buf.toString() + currentName);
                        if (currentName.equals("wpt")) {
                              lat = Double.parseDouble(parser.getAttributeValue(0));
                              lon = Double.parseDouble(parser.getAttributeValue(1));
                              waypointOpen = true;
                        }
                        if (waypointOpen && currentName.equals("sym")) {
                              parser.next();
                              sym = new String(parser.getText());
                              gotSym = true;
                        }
                        if (waypointOpen && currentName.equals("name")) {
                              parser.next();
                              name = new String(parser.getText());
                              gotName = true;
                        }
                        if (waypointOpen && gotSym && gotName) {
                              //System.out.println("*** " + name + " " + sym);
                              String localName = new String(name.toCharArray());
                              String localSym = new String(sym.toCharArray());
                              //System.out.println("*** " + localName.hashCode() + " " + lo
                                //calSym.hashCode());
                              TBCoordinates wp = new TBCoordinates(lat, lon, 0, localName,
```

```
                        localSym);
                    waypointOpen = false;
                    gotSym = false;
                    gotName = false;// Add MetaCoordinate to CoordinatesContainer
                    //System.out.println(wp.getLatitude() + " "+ wp.getLongitude() + " "
                      //+ wp.getName() + " " + wp.getType());
                    coordsList.put(localName, wp);
                    //System.out.println(coordsList.hashCode() + " "
                      //+ coordsList.size());
                }
                //System.out.println(buf.toString() + waypointOpen + " " + gotName + " " +
                  //gotSym);
                parseGPXData(parser);
            }

            parser.nextTag();
        }
        // we are at an end tag
        //System.out.println("got out");

    }catch(XmlPullParserException xe){
        //xe.printStackTrace();
        return; // not so great, but there seems to be no other choice
    }

}

    /** Returns the CoordinantsContainter
     * @return the CoordinatesContainer representing the coordinates
     */
    public CoordinatesContainer getCoordsContainer(){
        //System.out.println("getCoordsContainer() " + coordsList.hashCode());
        return coordsList;
    }

}
```

TBMIDlet.java
available at: http://sourceforge.net/p/walkingtoolsgpx/code/HEAD/tree/tbtool/src/edu/ucsd/calit2/TransBorderTool/TBMIDlet.java

```
/*  WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
    Copyright (C) 2007-2012 Walkingtools project/b.a.n.g. lab UCSD

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU Affero General Public License for more details.

    You should have received a copy of the GNU Affero General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

*/

// note: good idea to extend the dowsing interface to include (or redirect users)
// to sites that are within 500 meters.

package edu.ucsd.calit2.TransBorderTool;

import edu.ucsd.calit2.TransBorderTool.international.*;
import javax.microedition.MIDlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.location.*;
import java.io.*;
import java.util.*;
import javax.microedition.media.*;
import net.walkingtools.javame.util.AudioArrayPlayer;

/**
 * @author Brett Stalbaum and Jason Najarro
 * @version 0.5.5
 */
```

```
public class TBMIDlet extends MIDlet implements DowsingCompassListener, CommandListener {

    private Display display = null;
    // current displayable will normally be the tbDowsingCompass, but if expired, an alert.
    private DowsingCompass tbDowsingCompass = null;
    private Vector nearbyWPList = null;
    private static final int SEARCH_DISTANCE = 10000; // 10K
    private List targetList = null;
    private Alert arrivedAlert = null;
    private Alert waypointAheadAlert = null;
    private Alert expired = null;
    private boolean isExpired = false;
    private Alert expirationWarning = null;
    private boolean expireWarning = false;
    private Alert startUpDisplay = null;
    private boolean startUpAlert = false;
    private Alert noNearbyWaypoints = null;
    private Alert minimalistInfoAlert = null;
    private Command exit = null;
    private Command ignore = null;
    private Command cancel = null;
    private Command listNearbyWPs = null;
    private Command setTargetManual = null;
    private Command setTargetAuto = null;
    private AudioTimer audioTimer = null;
    // private String URL = "http://internetjunkee.com/transborder/GPScourseFinal.gpx";
    private LocationProvider lp = null;
    private TBCoordinates aheadCoords = null;
    // using only one audio player for two kinds of sound. The first is the
    // poems, the second is the Audio UI elements
    private static final String[] audioStrings = {
      "1-GAIN_04-02.wav", "2-GAIN_02-03.wav", "2-GAIN_03-01.wav",
      "3-GAIN_01.wav", "3-GAIN_02-02.wav", "5-GAIN_02-07.wav",
      "6-GAIN_04-02.wav", "7-GAIN_02-03.wav", "7-GAIN_03-01.wav",
      "8-GAIN_01.wav", "8-GAIN_02-02.wav", "10-GAIN_02-07.wav",
      "11-GAIN_04-02.wav", "12-GAIN_02-03.wav", "12-GAIN_03-01.wav",
      "13-GAIN_01.wav", "13-GAIN_02-02.wav", "15-GAIN_02-07.wav",
      "16-GAIN_04-02.wav", "17-GAIN_02-03.wav", "17-GAIN_03-01.wav",
      "18-GAIN_01.wav", "18-GAIN_02-02.wav", "20-GAIN_02-07.wav", // 24 poems
      "arriving.wav", "expiration.wav", "expired.wav", "found.wav", // AudioUI
```

```
  "lowgps.wav", "move.wav", "nosites.wav", "pointing.wav", // AudioUI
  "read.wav", "searching.wav", "startup.wav", "beep.wav" // AudioUI
};
private static final int NUMBER_OF_POEMS = 24;
private boolean running = false;
private boolean navigating = false;
// if the MIDlet is getting an update
// interval which is adequate for
// dymanic navigation, dynamicNavigation should be true
private boolean dynamicNavigation = false;
private Ticker minimalistTicker = null;
private net.walkingtools.international.Translation translation = null;
private AudioArrayPlayer audioPlayer = null;
private byte moveWarningEnervator = 1;

/**
 * Constructor for a TransBorderMIDlet
 */
public TBMIDlet() {
    // load the translation
    translation = Translation.loadTranslation(getAppProperty("language"));

    // get the display
    if (display == null) {
        display = Display.getDisplay(this);
    }

    // test value for jad file... can delete
    //System.out.println(System.currentTimeMillis()+1000*60*60*24*8);

    // set up the test alert first (for debugging on phone)
    exit = new Command(translation.translate("Exit"), Command.EXIT, 0);
    // get the gpx file
    String gpxFile = this.getAppProperty("GPXFile");

    tbDowsingCompass = new DowsingCompass(gpxFile);

    int width = tbDowsingCompass.getWidth();
    Image errorImage = null;
    Image tbImage = null;
```

```
if (width < 150) {
    errorImage = loadImage("error_sm.png");
    tbImage = loadImage("tb_sm.png");
} else {
    errorImage = loadImage("error.png");
    tbImage = loadImage("tb.png");
}

// first, validate the expiration value
ignore = new Command(translation.translate("Ignore"), Command.CANCEL, 0);
String expirationDate = this.getAppProperty("Expiration-Date");
long exp = Long.parseLong(expirationDate);

if (exp <= System.currentTimeMillis()) {
    expired = new Alert(translation.translate("Data expired"),
            translation.translate("The data is expired,
              TBTool is not safe to use."),
            errorImage,
            AlertType.ERROR);
    expired.setTimeout(Alert.FOREVER);
    expired.addCommand(exit);
    expired.addCommand(ignore);
    expired.setCommandListener(this);
    isExpired = true;
} else if (exp <= System.currentTimeMillis() +
   (1000 * 60 * 60 * 24 * 7) ) { // 7 day warning
    Date date = new Date(exp);
    expirationWarning = new Alert(translation.translate("Expiration Warning"),
            translation.translate("\nThe data will expire on:\n") + date.toString() +
            translation.translate("\nTransborder Immigrant Tool\
              nEDT/BANGLAB/CRCA/CALIT2/VISARTS/UCSD\n\n"),
            errorImage,
            AlertType.WARNING);
    expirationWarning.addCommand(ignore);
    expirationWarning.setTimeout(15000);
    expirationWarning.setCommandListener(this);
    expireWarning = true;
} else {
    Date date = new Date(exp);
    startUpDisplay = new Alert(translation.translate
```

```
      ("Transborder Immigrant Tool"),
            translation.translate("\nExpires: ") + date.toString() +
            translation.translate("\nTransborder Immigrant Tool\
              nEDT/BANGLAB/CRCA/CALIT2/VISARTS/UCSD\n\n"),
            tbImage,
            AlertType.INFO);
    startUpDisplay.addCommand(ignore);
    startUpDisplay.setTimeout(10000);
    startUpDisplay.setCommandListener(this);
    startUpAlert = true;
}

noNearbyWaypoints = new Alert(translation.translate("No Nearby Points"),
        translation.translate("There are no sites within ") +
        (int)((SEARCH_DISTANCE / 1000.0) + .5)
        + translation.translate(" Kilometers."),
        tbImage,
        AlertType.WARNING);

noNearbyWaypoints.addCommand(ignore);
noNearbyWaypoints.setTimeout(10000);
noNearbyWaypoints.setCommandListener(this);

minimalistInfoAlert = new Alert(translation.translate("Minimal Mode"),
        "",
        tbImage,
        AlertType.WARNING);

minimalistInfoAlert.addCommand(ignore);
minimalistInfoAlert.setTimeout(Alert.FOREVER);
minimalistInfoAlert.setCommandListener(this);

nearbyWPList = new Vector();
cancel = new Command(translation.translate("Cancel"), Command.CANCEL, 0);
listNearbyWPs = new Command(translation.translate("Find"), Command.SCREEN, 1);
setTargetManual = new Command(translation.translate("Select"), Command.SCREEN, 1);
setTargetAuto = new Command(translation.translate("Set Target"), Command.SCREEN, 1);

/* through a lot of tedious testing, I discovered that these
constructors of the TextFields were throwing an IllegalArgumentException
```

```
when using TextField.DECIMAL or TextField.NUMERIC constraints. The
following is from the javadoc. It seems not to contradict the
use of TextField.DECIMAL or NUMERIC given that I was setting
the forms to a decimal/numeric value... hmmmmm... this must be an
issue in iden implementation.

"Some constraints, such as DECIMAL, require the implementation to
perform syntactic validation of the contents of the text object.
The syntax checking is performed on the actual contents of the text
object, which may differ from the displayed contents as described
above. Syntax checking is performed on the initial contents passed
to the constructors, and it is also enforced for all method calls
that affect the contents of the text object. The methods and
constructors throw IllegalArgumentException if they would result
in the contents of the text object not conforming to the required
syntax."
 */
tbDowsingCompass.addCommand(exit);
tbDowsingCompass.addCommand(listNearbyWPs);
tbDowsingCompass.setCommandListener(this);
tbDowsingCompass.addNavigatorListener(this);

targetList = new List(translation.translate("Select a Target"), List.IMPLICIT);
targetList.addCommand(cancel);
targetList.addCommand(setTargetManual);
targetList.setCommandListener(this);

waypointAheadAlert = new Alert(translation.translate("Site Ahead!"),
        translation.translate("Site Ahead!"),
        tbImage, AlertType.INFO);
waypointAheadAlert.setTimeout(Alert.FOREVER);
waypointAheadAlert.addCommand(ignore);
waypointAheadAlert.addCommand(setTargetAuto);
waypointAheadAlert.setCommandListener(this);

arrivedAlert = new Alert(translation.translate("Arrived at Site"),
        translation.translate("Arrived at Site"),
        tbImage, AlertType.INFO);
arrivedAlert.setTimeout(Alert.FOREVER);
arrivedAlert.addCommand(ignore);
```

```
arrivedAlert.setCommandListener(this);
minimalistTicker = new Ticker(
            translation.translate("Minimal or no GPS signal.
              Alert will give direction and distance information if possible.")
        );
dynamicNavigation = true;
  // assume active navigation at startup of gps to give it a chance to fix

// set up location provider
// Set criteria for selecting a location provider:
// accurate to 50 meters horizontally
try {
    Criteria cr = new Criteria();
    cr.setHorizontalAccuracy(50);
    // we can set other criteria that we require
    cr.setSpeedAndCourseRequired(true);
    cr.setPreferredResponseTime(2000);
    cr.setAltitudeRequired(true);
    try {
        // Get an instance of the provider
        lp = LocationProvider.getInstance(cr);
    } catch (LocationException e) { // if this happens, lp could not get a location
        display.setCurrent(new Alert(translation.translate
          ("Exception on getting location provider"),
                translation.translate("Exception on getting location provider")
                  + ':' + e.toString(),
                null,
                AlertType.INFO));
    }
    // register this with the location listener
    // the second argument is the interval. -1 is a flag that says,
    // "whatever works best for you"
    // the third arg is the timeout, or, how many seconds past the
    // interval defined in arg 2
    // the provider should wait before it returns an invalid Location
    // the fourth is the maxAge of a valid location.
    // The provider may provide a valid location
    // in lieu of a current location as long as it is not older than this.
    lp.setLocationListener(tbDowsingCompass, 2, 2, 2);
} catch (SecurityException se) {
```

```
            Alert noLocationService = new Alert(translation.translate
             ("TBtool requires location"),
                    translation.translate("The Transborder Immigrant Tool
                      needs access to location services.") +
                    translation.translate("Try answering \"Yes\"
                     on startup to grant TBtool access."),
                    errorImage,
                    AlertType.INFO);
            noLocationService.setTimeout(Alert.FOREVER);
            noLocationService.addCommand(exit);
            noLocationService.setCommandListener(this);
            display.setCurrent(noLocationService);
        }
    }

    protected void startApp() throws MIDletStateChangeException {
        // get the display
        if (display == null) {
            display = Display.getDisplay(this);
        }

        // this thread to randomly play audio file
        try {
            audioPlayer = new AudioArrayPlayer("audio", audioStrings, true);
            // true, in audio cueing mode
            //InputStream in = getClass().getResourceAsStream("/audio/beep.wav");
            audioTimer = new AudioTimer();
            running = true;
            audioTimer.start(); // start audio thread
        } catch (IOException e) {
            Alert bailOnAudioException = new Alert(translation.translate
             ("Could not load audio"),
                    translation.translate("Could not load audio"),
                    loadImage("error_sm.png"),
                    AlertType.INFO);
            bailOnAudioException.setTimeout(Alert.FOREVER);
            bailOnAudioException.addCommand(exit);
            display.setCurrent(bailOnAudioException);
        } catch (MediaException e) {
            Alert bailOnAudioException = new Alert(translation.translate
```

```
            ("Could not play audio"),
                  translation.translate("Could not play audio"),
                  loadImage("error_sm.png"),
                  AlertType.INFO);
            bailOnAudioException.setTimeout(Alert.FOREVER);
            bailOnAudioException.addCommand(exit);
            display.setCurrent(bailOnAudioException);
        }

        // make sure the data is not expired
        if (isExpired) {
            display.setCurrent(expired);
            display.vibrate(1000);
            playAudioFile("expired.wav", true);
        } else if (expireWarning) {
            display.setCurrent(expirationWarning, tbDowsingCompass);
            display.vibrate(1000);
            playAudioFile("expiration.wav", true);
        } else if (startUpAlert) { //first time only
            startUpAlert = false;
            display.setCurrent(startUpDisplay, tbDowsingCompass);
            display.vibrate(1000);
            playAudioFile("startup.wav", true);
        } else { // we are good to go
            display.setCurrent(tbDowsingCompass);
        }
    }

    /**
     * edu.ucsd.calit2.TransBorderTool.CompassListener interface method
     * Called when user is facing a waypoint
     * Displays waypointAheadAlert pertaining to type of waypoint
     */
    public void witchingEvent(TBCoordinates mc) {
        aheadCoords = mc;
        if (display.getCurrent().equals(tbDowsingCompass)) {
            waypointAheadAlert.setString(tbDowsingCompass.getInfo(mc));
            waypointAheadAlert.setImage(aheadCoords.getIcon());
            double distance = tbDowsingCompass.distanceTo(mc);
            if (distance > SEARCH_DISTANCE) {
```

```
            display.vibrate(100);
        } else if (distance > 1000) {
            display.vibrate(300);
        } else if (distance > 500) {
            display.vibrate(500);
        } else if (distance > 100) {
            display.vibrate(800);
        }
        display.setCurrent(waypointAheadAlert);
        display.vibrate(1000);
        playAudioFile("found.wav", false);
    }
}

/**
 * NavigatorListener interface method
 * Displays alert when user arrives within range of target
 */
public void arrivedAtTarget(int distance) {
    navigating = false;
    // stop the compass from navigating
    tbDowsingCompass.stopNavigation();
    display.setCurrent(arrivedAlert);
    display.vibrate(1000);
    playAudioFile("arriving.wav", false);
}
```



```
// all of the UI audio files are played through this method
// the poems are not played through this method,
// see second arg in playFileName below
private void playAudioFile(String name, boolean interrupt) {
    try {
        audioPlayer.playFileName(name, interrupt); // true will interrupt a poem if playing
    } catch (MediaException e) {
        try {
            audioPlayer.playFileName("beep.wav", true);
        } catch (MediaException ex) {
            return;
        } catch (Exception eb) {
            eb.printStackTrace();
```

```
        }
    }
}

/**
 * NavigatorListener interface method
 * Called to populate nearby waypoint vector
 * once the CompassCanvas detects a valid location
 * so user may begin "dowsing" for waypoints
 * @param ready true for ready to navigate
 */
public void navigationReady(boolean ready) {
    if (ready) {
        nearbyWPList = tbDowsingCompass.getNearbyWaypoints(SEARCH_DISTANCE);
        if (navigating) {
            tbDowsingCompass.removeCommand(listNearbyWPs);
            tbDowsingCompass.addCommand(cancel);
        } else {
            tbDowsingCompass.addCommand(listNearbyWPs);
            tbDowsingCompass.removeCommand(cancel);
        }
    } else {
        if (!navigating) {
            tbDowsingCompass.removeCommand(listNearbyWPs); // can't use
        }
    }
}

/**
 * NavigatorListener interface method tells the MIDlet the GPS refresh rate
 * of the Navigator (DowsingCompass...) If the MIDlet is getting an update
 * interval which is adequate for dymanic navigation then dynamic
 * (compass based) navigation should be on.
 * Otherwise the phone enters into a minimalist mode that can still provide
 * an occasional alert, useful with less capable phones or in place where
 * GPS coverage is poor. In these cases the user may still be able to navigate
 * with a magnetic compass.
 * @param milliseconds reported milliseconds since last update
 */
public void updateInterval(long milliseconds) {
```

```
// if the device is without update for 10 minutes, enter minimal mode
if (milliseconds > 1000*60*10) { // signal is not good
    if (dynamicNavigation) { // entering non dynamic mode from dynamic
        tbDowsingCompass.setTicker(minimalistTicker);
        if (!navigating) {
            tbDowsingCompass.removeCommand(listNearbyWPs); // can't use
        }
        display.vibrate(1000);
        playAudioFile("lowgps.wav", true);
    }
    dynamicNavigation = false;
 } else { // we have a good signal
    if (!dynamicNavigation) { // we are now returning from a bad signal
        // because dN is set to true in the constructor
        // we must be returning from non-dynamic to dynamic, not just starting
        // restore interface to last state
        tbDowsingCompass.setTicker(null);
        // Offer any available help to user
        // get closest point data into alert string if available
        nearbyWPList = tbDowsingCompass.getNearbyWaypoints(SEARCH_DISTANCE);
        if (nearbyWPList != null && !nearbyWPList.isEmpty()) {
            TBCoordinates target = (TBCoordinates)nearbyWPList.elementAt(0);
            Coordinates current = tbDowsingCompass.getCurrentCoords();
            float distance = current.distance(target);
            String distanceStr = null;
            if (distance >= 1000) {
                distanceStr = (int)(distance/1000) + translation.translate(" Kilometers");
            } else {
                distanceStr = distance + translation.translate(" Meters");
            }
            // create minimalist info alert (if it is just an intermittent single report
            // then at least this info will be left on screen as the system goes
            // back into non dynamic navigation mode.
            minimalistInfoAlert.setString(
                    translation.translate("Nearest Site: Distance ") + distanceStr + ", " +
                    translation.translate("Azimuth ") + (int)current.azimuthTo(target)
                    + translation.translate(" degrees, General Direction ") +
                    tbDowsingCompass.directionTo(target)
                );
            display.setCurrent(minimalistInfoAlert, tbDowsingCompass);
```

```
                display.vibrate(1000);
                playAudioFile("read.wav", false);
            } else {
                display.setCurrent(noNearbyWaypoints);
                display.vibrate(1000);
                playAudioFile("nosites.wav", false);
            }
        }
        dynamicNavigation = true;
    }
}

/** (non-Javadoc)
 * @param arg0
 * @throws MIDletStateChangeException
 * @see javax.microedition.MIDlet.MIDlet#destroyApp(boolean)
 */
protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
    // TODO Auto-generated method stub
}

/**
 *
 */
protected void pauseApp() {
    // TODO Auto-generated method stub
}

private Image loadImage(String str) {
    Image image = null;
    try {
        image = Image.createImage("/img/" + str);
    } catch (IOException e) {

        image = null;
    }
    //System.out.println(image);

    return image;
}
```

```
public void commandAction(Command c, Displayable d) {
    if (c == exit) { // exit
        running = false;
        notifyDestroyed();
    } else if (c == cancel) { // stop navigation and reset softkey commands
        navigating = false;
        tbDowsingCompass.stopNavigation();
        tbDowsingCompass.removeCommand(cancel);
        if (dynamicNavigation) {
            tbDowsingCompass.addCommand(listNearbyWPs);
        }
        display.setCurrent(tbDowsingCompass);
    } else if (c == ignore) {       // Returns to compass interface if user chooses not
                                    // to set a dowsingEvent as a target
        if (navigating) {
            tbDowsingCompass.removeCommand(listNearbyWPs);
            tbDowsingCompass.addCommand(cancel);
        } else {
            tbDowsingCompass.removeCommand(cancel);
            if (dynamicNavigation) {
                tbDowsingCompass.addCommand(listNearbyWPs);
            }
        }
        display.setCurrent(tbDowsingCompass);
    } else if (c == listNearbyWPs) {    // Display a List of waypoints within range
                                        // from which user can manually choose a target
                                        // Update nearby waypoint vector
        nearbyWPList = tbDowsingCompass.getNearbyWaypoints(SEARCH_DISTANCE);
        if (nearbyWPList != null && !nearbyWPList.isEmpty()) {
            targetList.deleteAll();
            // Loop through waypoint vector adding waypoint
            // image and information to list
            for (int i = 0; i < nearbyWPList.size(); i++) {
                TBCoordinates mc = (TBCoordinates) nearbyWPList.elementAt(i);
                targetList.append(tbDowsingCompass.getInfo(mc), mc.getIcon());
            }
            display.setCurrent(targetList);
        } else {
            display.setCurrent(noNearbyWaypoints);
```

```
            playAudioFile("nosites.wav", true);
        }
    } else if (c == setTargetAuto) {
    // Set a waypoint detected by a dowsingEvent as the target
        navigating = true;
        tbDowsingCompass.setTarget(aheadCoords);
        // Change commands on tbDowsingCanvas
        tbDowsingCompass.removeCommand(listNearbyWPs);
        tbDowsingCompass.addCommand(cancel);
        display.setCurrent(tbDowsingCompass);
        // Set a waypoint selected from nearby waypoint List as the target
    } else if (c == setTargetManual) {
        navigating = true;
        int index = targetList.getSelectedIndex();
        tbDowsingCompass.setTarget((TBCoordinates) nearbyWPList.elementAt(index));
        //Change Commands on tbDowsingCanvas
        tbDowsingCompass.removeCommand(listNearbyWPs);
        tbDowsingCompass.addCommand(cancel);
        display.setCurrent(tbDowsingCompass);
    }
}

public void motionStatusUpdate(boolean isMoving) {
    if (isMoving) { // updated to moving
        nearbyWPList = tbDowsingCompass.getNearbyWaypoints(SEARCH_DISTANCE);
        // so update nearby points
    } else { // updated not moving
        display.vibrate(200);
        if (moveWarningEnervator % 5 == 0) { // only play this file ~ every 5th time
            playAudioFile("move.wav", false);
            // the "move for compass" message can be too frequent
        }
        moveWarningEnervator++;
    }
}

// inner class to control audio
class AudioTimer extends Thread {

    Random rand = new Random();
```

```
        public void run() {
            while (running) {
                try {
                    Thread.sleep(1000 * 60 * (rand.nextInt(19) + 1)); // sleep random min-
utes
                    //Thread.sleep(1000 * 60); // sleep one min (test)
                } catch (InterruptedException e) {
                    running = false;
                }
                try {
                    int randIndex = rand.nextInt(NUMBER_OF_POEMS);
                    // poems at the top of the audio array
                    audioPlayer.play(randIndex, false);
                    // false means to cue the audio if something else is playing
                } catch (MediaException ex) {
                    Alert bailOnAudioException =
                      w      new Alert(translation.translate("media exception"),
                            ex.getMessage(),
                            null,
                            AlertType.INFO);
                    bailOnAudioException.setTimeout(Alert.FOREVER);
                    display.setCurrent(bailOnAudioException);
                }
            }
        }

        public void finalize() {
            running = false;
        }
    }
}
```

The desert is an ecosystem with a logic of sustainability, of orientation, unique unto itself. For example, if the barrel cactus—known otherwise as the compass cactus—stockpiles moisture, it also affords direction. As clear as an arrow or a constellation, it leans south. Orient yourself by this mainstay or by flowering plants that, growing toward the sun, face south in the Northern Hemisphere.

El desierto es un ecosistema único en si mismo, con su propia lógica de sustentabilidad y orientación. Por ejemplo, el cactus de barril o biznaga—conocido como cactus brújula—acumula el rocío y la humedad, también proporciona direcciones. Tan claro como una flecha o una constelación, se inclina hacia el sur. Oriéntese haciendo uso de este recurso o por las plantas que florecen, creciendo hacia al sol, ya que éstas siempre apuntan hacia el sur en el Hemisferio Norte.

Camine o escale en las mañanas. Descanse al mediodia bajo un mezquite, una hediondilla o gobernadora para protegerse de la abrasadora temperatura del suelo. Recuerde—aun la víbora de cascabel aerodesliza la mayor parte de su cuerpo sobre la ardiente arena mientras deja su característica huella en forma de "J" a lo largo de las dunas del desierto.

Climb or walk in the morning. Rest midday beneath creosote bush or mesquite, insulating yourself from the superheated ground. Remember—even the sidewinder hovercrafts, the bulk of its body above the scalding sand as it leaves its trademark J-shaped tracks across the desert dunes.

Justo antes del amanecer, los beduinos volteaban piedras semienterradas
Just before sunrise, Bedouins turned over half-buried stones in the desert
para atrapar el rocío que el frío de la noche había condensado en su
to catch the dew that the night's coolness had condensed on the stones'
superficie. Los viajeros indígenas en el corredor de México-Estados Unidos
surfaces. Indigenous travelers in the Mexican-U.S. corridor searched
buscan las anchas hojas de la yuca y el agave para abastecerse de agua.
the broad leaves of yucca and agave. Rainwater collected at each plant's
El agua de la lluvia se acumula en la base de cada planta—el ápice de las
base—the leaves' apex—remaining there up to a few days after a summer or
hojas—permaneciendo ahí hasta unos días después de una llovizna veraniega
winter shower. Proceed from the simple premise: The desert caches water in
o invernal. La premisa a seguir es simple: el desierto acumula agua en
unlikely places that it resists divulging. Do not expend all your energies
lugares inesperados que resiste divulgar. No agote sus energías buscando sus
searching for its secret stashes, but likewise do not assume that its pockets
escondites secretos, pero igualmente no presuma que sus bolsillos de humedad
of moisture are nonexistent. Restrict your water reconnaissance to early or
son inexistentes. Limite su búsqueda de agua al principio y al final del día
late in the day when your liquid net-gain will outweigh the perspiration you
de manera que el líquido que encuentre sea más que el sudor que transpire al
expend. A thirst is seldom quenched; it morphs to reappear on the horizon.
buscarlo. La sed es rara vez saciada: se transforma para reaparecer en el
Meanwhile, the desert reflects the sun back like a mirror. You are caught in
horizonte. Mientras tanto, el desierto refleja el sol como en un espejo. Usted
that pair's uneven, inconsummate exchange.
está atrapad@ en el medio de este intercambio desigual e inconsumado.

El avión piloteado por el Teniente Edwin Zolnier del Cuerpo de Marines de EE.UU. realizó un aterrizaje forzoso en el Desierto de Sonora. Al ser rescatado cinco días más tarde, Zolnier reconoció que el cactus de barril—biznaga—fue lo que lo mantuvo con vida. Técnicamente, se puede sobrevivir de la humedad de ciertos cactus, pero para esto hay que reconocer la diferencia entre lo sostenible y lo insostenible. La pulpa y la savia de algunos cactus son tan tóxicas, que tendrá que ser hospitalizad@ en caso de beberlas y comerlas. Algunos cactus no l@ matarán, pero pueden dejarle muy enferm@. La regla de oro: Solo corra el riesgo de comer o tomar cactus si la única alternativa que tiene es morir de sed. El saguaro y el órgano envenenan. Punto. Y no todos los cactus de barril son iguales. Para hacer las cosas más difíciles, los saguaros jóvenes se parecen mucho a las biznagas. Así que no debe concentrarse en buscar solamente cactus chaparros y redondos, tiene que diferenciarlos, piense en los anzuelos. Las espinas en forma de "J," utilizadas por los indígenas seris para pescar, marcan y distinguen los cactus que verdaderamente le salvarán la vida. Cuando haya encontrado el cactus correcto, corte la parte superior con un cuchillo o piedra afilada, pero no espere encontrar una fuente de líquido en su interior. Céntrese, corte un pedazo de la pulpa interior blancuzca que está en el centro de la planta. Mastíquela. Deje que el jugo baje por su garganta. Escupa la pulpa una vez que le haya chupado todo el líquido. No se trague el bagazo. Descanse, digiera. (Cualquier esfuerzo después de comer, podría causar que perdiera lo poco que ha ganado en el proceso.) Si el sabor de la pulpa le provoca ganas de vomitar, coloque la pulpa bajo la sombra literal o improvisada (e.g., cúbrala por poco tiempo con su cuerpo). Al enfriarse, aunque sea un par de grados, el líquido sabrá súper concentrado, más "vegetal" que la mayoría de los vegetales apetitosos para los seres humanos, pero puede ser más fácil de tragar.

U.S. Marine Corps pilot Lieutenant Edwin Zolnier's plane crash-landed in the Sonoran Desert. Rescued five days later, Zolnier credited the barrel cactus with keeping him alive. Technically you can survive on moisture from select cacti, but you have to recognize the difference between the tenable and the untenable. Some cacti's sap and pulp are so toxic you will need to be hospitalized afterward if you drink or eat them. Other cacti won't kill you, but will leave you sick as a dog. Baseline rule: Only take the risk of eating or drinking cacti if the alternative you face is dying of thirst. Saguaro and organ poison. Punto. And, not all barrel cacti are created equal. To make matters worse, young saguaros easily could be mistaken for barrels. So, don't just look for squat, rounded cacti; differentiate, think fishhook. J-shaped, outer "fishhook" spines, literally used by the Seri Indians for fishing, mark and distinguish the true rescue cactus from its peers. When you've found the right plant, cut off its top with a knife or sharp rock but, don't expect to find a font of liquid. Center yourself, cut out a chunk of the whitish inner pulp from the cactus' correspondent center. Chew it. Let the juice run down your throat. Spit out the pulp when you've sucked it dry. Don't swallow the pithy fiber. Rest, digest. (Exertion after eating could cause you to forfeit the little you gained in the process.) If the taste of the juice makes you gag, place the pulp in literal or makeshift shade (e.g., shield it for a short time with your body). Cooler by even a few degrees, its liquid still will taste super-concentrated, more "vegetal" than most vegetables palatable to human beings, but it might be easier to choke down.

You can survive without eating anything for three weeks in hot weather. But, the body's need for hydration is a different matter entirely. Consume the fruit of prickly pear, saguaro, organ pipe, yucca, or cholla for their moisture alone.

In the summertime, pitahaya dulce, the fruit of the organ pipe cactus, ripens to red and drops its spines. The prickly pear cactus' tuna reddens to purple, but never loses its needles. Dethorned, dethroned, both are delectably edible. Peel their skins.

En climas cálidos, puede sobrevivir hasta tres semanas sin comer nada. No obstante, la necesidad de hidratación del cuerpo humano es un problema completamente distinto. Consuma las tunas del nopal, saguaro, órgano, yuca y cholla, aunque sólo sea por su contenido líquido.

En el verano, la pitaya dulce, la fruta del cactus órgano, se torna roja y pierde sus espinas al madurar. La tuna se torna un color morado, mas nunca pierde sus púas. Destronadas, desespinadas, ambas son deliciosas. Quíteles la piel.

En cuestiones de vida o muerte, podrían casi todos afirmar, "el gusto es relativo." (Tal es el destino de la poesía como artificio, arte o sustento—una no-cuestión si no se puede beber, comer o respirar, aun si "la poesía no es un lujo".) Sin embargo, el sabor de los cactus presenta un dilema particularmente espinoso. Aún no mezcal o tequila, muchos cactus retienen humedad, pero también toxinas. De nuevo, la norma básica: sólo tome el riesgo de comer o beber cactus si la alternativa es morir de sed. Pruebe, pruebe, pruebe. El barril de anzuelo es quizás su mejor opción, pero asegúrese de que pueda identificarlo antes de ingerir su contenido. Si no está segur@ de que ha encontrado la planta correcta, ponga una pequeña porción de la pulpa en su boca, pruébelo antes de tragar su savia. El sabor debe parecerse al de vegetales súper-saturados. Escupa todo lo que sea acre, amargo o tan desagradable que provoque ahogo sin control o vómito. Espere aproximadamente treinta minutos para medir la tolerancia de su cuerpo a este experimento—es mejor quedarse sedient@ cerca del nocivo saguaro que envenenarse o acelerar las etapas de la deshidratación. ¡Ánimo! Un paisaje que sustenta al saguaro es igualmente favorable para el barril de anzuelo.

In matters of life or death, most would contend, "Taste is relative." (Such is the fate of poetry as artifice, art, or sustenance—a non-issue if one cannot drink, eat, or breathe, even if "poetry is not a luxury.") Still, the taste of cacti presents a particularly thorny conundrum. Not yet mezcal or tequila, many cacti hold moisture, but also harbor toxins. Again, the baseline rule: Only take the risk of eating or drinking cacti if the alternative is dying of thirst. Test, test, test. The fishhook barrel is perhaps your best bet, but make sure you can identify it positively before ingesting its contents. If you're not sure you've found the right plant, put a small portion of its pulp in your mouth, taste it before you swallow its sap. Expect the flavor of super-saturated vegetables. Spit out anything that is acrid, bitter, or so unsavory that it makes you feel as if you will choke uncontrollably or vomit. Wait approximately thirty minutes to gauge your body's tolerance of this experiment—better to stay thirsty within arm's reach of noxious saguaro than poison yourself or speed up your stages of dehydration. ¡Ánimo! A landscape that sustains the saguaro is equally amenable to the fishhook barrel.

Cottonwoods spread a welcome—shade. Clusters also indicate a desert stream or an underground spring close to the surface. Cut the tracks of animals. Walk in the footprints of coyote or fox to the freshest water available. Flies and mosquitoes rarely move far from a source. Javelina divine permanence, traveling in the corridors of washes. The flight paths of birds, like pigeons and doves, indicate the proximity of an oasis. They drink in the evenings. If they are flying low and slow, follow their direction. Where they came from may be where you need to go to refill bottles or canteens.

Los álamos negros ofrecen una bienvenida—la sombra. Las arboledas también indican un arroyo en el desierto o un manantial subterráneo cercano a la superficie. Rastree las pistas de las animales. Camine en las huellas de los coyotes o los zorros hacia el agua más fresca. Las moscas y los mosquitos rara vez se alejan de una fuente de agua. La divina permanencia de los jabalís, viajando por las zanjas. Las trayectorias del vuelo de los pájaros, como los pichones y las palomas, también indican la cercanía de un oasis. Ellos beben por las tardes. Si vuelan bajo y despacio, siga su dirección. De donde vienen quizás sea donde necesite ir para llenar su botella o frasco.

Según Heródoto, el rey Cambises perdió todo su ejército persa (cincuenta mil hombres) en una tormenta de arena egipcia hace dos mil quinientos años. Los desiertos guardan sus secretos; no hay evidencia arqueológica que corrobore la versión de Heródoto. Repentinos haboobs locales—a veces el resultado de las corrientes de aire frío generadas por las tormentas eléctricas—todavía amenazan a los viajeros del desierto. Los instrumentos en el desierto de Mojave han registrado velocidades de viento de hasta ciento sesenta kilómetros por hora. En una tormenta de arena o polvo, el calor sofoca; los vientos transubstancian el paisaje en objetos voladores no identificados. La arena se convierte en papel de lija contra la piel. De la espalda al viento. Acuéstese de costado contra un cerro o duna. Cubra su cara. Tápese la nariz y la boca. Guarde sus lentes en su camisa o en el bolsillo de la chaqueta (para que el viento no grabe sus soliloquios en ellos). Espere a que pase. Ningún Godot, una tormenta de arena seguirá el camino de un berrinche.

According to Herodotus, King Cambyses, twenty-five hundred years ago, lost his entire Persian army (fifty thousand men) in an Egyptian sandstorm. Deserts guard their secrets; no archaeological evidence corroborates Herodotus's account. Sudden, local haboobs—sometimes the result of thunderstorms' cool air outflows—still menace desert travelers. Instruments in the Mojave Desert have clocked wind speeds up to one hundred sixty kilometers per hour. In a sand or dust storm, heat suffocates; winds transubstantiate the landscape into unidentified flying objects. Sand becomes sandpaper against the skin. Turn your back to the wind. Lie on your side against a hill or dune. Wrap your face. Cover your nose and mouth. Tuck your eyeglasses into your shirt or jacket pocket (so the wind will not etch its soliloquies into them). Wait it out. No Godot, a sandstorm will go the way of a temper tantrum.

Africanized honey bees (AHB) present a formidable challenge to anyone traveling the Americas. Other bees might circle a person, but would refrain from stinging him or her without provocation. Sweat bees, for instance, will land on a hand to drink their fill of perspiration. Left to their own devices, they will not attack, but will fly away when they're satiated. But, "killer," or "assassin bees," the descendants of migrants (themselves the descendants of twenty-six Tanzanian bees accidentally released in southeast Brazil), are aggressively territorial. Almost identical in appearance to kinder and gentler bee populations, Africanized bees—now the reigning queens and workers of the Sonoran Desert—congregate near water holes and flowering cacti. Killer bees will defend their hives against perceived threats, attacking by the thousands. Do not pass within thirty meters of their colonies (eminent domains), constructed in veritable earthworks (mounds and cavities), cacti trunks, creosote, mesquite, former travelers' lay-up sites. (They've even been known to nest beneath the lids of water cache barrels!) If you see large numbers of bees' patrolling or hear their drone, move on. Killer bees read dark or bright colors—clothing or hair—as the sign of a predator. If AHBs swarm, do not wave your arms or swat their bodies. Run! Or, wrap yourself tightly in a blanket. Africanized bees motion-detect enemies, measure the latter's expelled carbon monoxide. Your first priority must be to protect your face (swelling of the mouth and nose inhibits breathing). If stung, find a safe space. Then, use your fingers, an earring, a shard of granite to remove stingers and venom sacks lodged beneath the skin (otherwise, even those fragments will keep pumping venom into you).

Las abejas africanas son un reto formidable para los que viajan por las Américas. Otras abejas pueden rodear a una persona, pero rara vez la picarían sin provocación. Las abejas del sudor, por ejemplo, aterrizan en la mano para beber hasta saciarse de la transpiración. Si no las perturba, no atacarán y se irán volando cuando estén satisfechas. Sin embargo, las abejas asesinas, descendientes de migrantes(que a su vez vienen de veintiséis abejas de Tanzania liberadas accidentalmente al sureste de Brasil), son agresivamente territoriales. Son casi idénticas a otras abejas mas dóciles y tranquilas. Las abejas africanas—ahora las reinas y trabajadoras del desierto de Sonora—se agrupan cerca de los posos de agua y de los cactus que han florecido. Las abejas asesinas defenderán sus panales en contra de cualquier amenaza: miles atacarán. Mantenga una distancia de no menos de treinta metros de sus colonias (dominios eminentes), construidas sobre un terraplén (lomas y huecos), cactus, creosotas, mezquites, los acampamentos abandonados de otros viajeros.(¡Hasta han anidado debajo de las tapas de los barriles de agua!) Si ve muchas abejas patrullando o escucha su zumbido, siga caminando. Las abejas asesinas perciben los colores oscuros o brillantes—en la ropa o el cabello—como un predador. Si las abejas forman una nube, no agite sus brazos y no las golpee. ¡Corra! O abríguese con una sabana. Las abejas africanizadas siguen los movimientos del enemigo, midiendo el monóxido de carbono que suelta. La prioridad debe ser cubrirse la cara (la hinchazón de la nariz y boca impide su respiración). Si le pican, busque un lugar seguro. Después, utilice sus dedos, un arete, una trozo de granito para quitarse los aguijones y los sacos de veneno alojados bajo la piel (si no lo hace, estos fragmentos continuarán inyectándole veneno).

Las tarántulas hembras envuelven sus madrigueras con seda. Emprendedoras, explayan sus elaboradamente tejidas entradas por la noche para atrapar a los inocentes insectos cercanos. Aunque algunos tejanos del sur sostienen que el masivo éxodo nocturno de las tarántulas son los muertos que salen de sus tumbas; de hecho, las tarántulas macho, ligeramente venenosas, rutinariamente se aventuran durante el ocaso. Si cruzan su camino, viva y deje vivir. De lo contrario, estos arácnidos con pulmones laminares le morderán o chasquearán, soplando los pelos de sus panzas al aire. Como las astillas, sus misivas entrarán en su piel, provocando gradaciones de irritación (picor, hinchamiento, topetones rojizos levantados). Peor aún, si sus pelos flotan como esporas a sus ojos, espere una irritación similar a la inducida por el spray de pimienta o el gas lacrimógeno. Pasará; pero, considere que en Italia muchos creyeron una vez que la única cura para una mordedura o el vertimiento de la tarántula era la danza ahora conocida como la tarantela.

Female tarantulas line their burrows with silk. Enterprising, they swing their elaborately webbed entrances open at night to spring on guileless insects nearby. Although some South Texans contend that nocturnal mass exoduses of tarantulas are the dead rising from their graves; in fact, male tarantulas, mildly venomous, routinely venture forth at dusk. If they cross your path, live and let live. Otherwise, these book-lunged arachnids will bite or flick you off, blowing hairs from their bellies into the air. Like slivers, their missives will enter your skin, provoking gradations of irritation (itching, swelling, raised reddish bumps). Worse yet, if their hairs float like spores into your eyes, expect a reaction comparable to that induced by pepper spray or tear gas. It will pass, but consider, in Italy, many once believed that the only cure for a tarantula’s bite or shedding was the dance now known as the tarantella.

Western Diamondbacks—light to medium brown (liable to reach a length of two meters)—densely populate North American deserts. Most often found in brushy washes and shady recesses, in the springtime, they emerge from hibernation, blooming like flowers. At this, their venomous peak, they sunbathe, flashing their diamond-shaped patches. Their prey of choice: mice and kangaroo rats. Diamondbacks hold a dubious record—the majority of the continent's snakebite cases. Listen for their rattle. Look for the shallow circular marks they leave in locations where they've coiled up to sleep. Diamondbacks, too, are creatures of habit, returning to rest stops. Poisonous snakes fall into two categories (dependent upon their venom). Diamondbacks pack a hemotoxic punch, albeit not as concentrated as that of some snakes. They make up for this supposed lack in their powerful mode of delivery, injecting large quantities of venom with single surgical strikes. Hemotoxins flood the bloodstream, destroy blood cells, damage tissue, catalyze internal hemorrhaging. Yes, reactions to snakebites vary. But, if you are bit by a diamondback—especially in March or April—within three minutes the area around the wound will redden and swell. And, left untreated, your body will not deviate from a well-rehearsed script: massive swelling and blistering, a steady lowering of blood pressure, headache, severe pain, blood in the urine. Do not block circulation or make incisions to the wound. Do not take painkillers or sedatives. Do not eat. Remove all clothing and jewelry beneath the area. Remain motionless; keep the bite below your heart. Ideally, get someone else to cleanse the wound. And, most importantly—borders be damned—call 9-1-1 or 0-6-6; seek medical attention immediately.

La cascabel diamantada del oeste—color café claro a oscuro (capaz de alcanzar una longitud de dos metros)—puebla densamente los desiertos de América del Norte. Se le encuentra con mayor frecuencia en los deslaves de maleza y nichos con sombra, en la primavera, emergen de la hibernación, brotando como las flores. Ahí, en su pico venenoso, toman el sol, destellando sus parches adiamantados. Su presa preferida: los ratones y ratas canguro. La cascabel diamantada tiene un récord dudoso—la mayoría de los casos de mordedura de serpiente en el continente. Preste atención a su cascabel. Busque las superficiales marcas circulares que dejan en los lugares donde se han enrollado a dormir. Las cascabeles también son criaturas de hábito, volviendo a sus lugares de reposo. Las serpientes venenosas se clasifican en dos categorías (en relación a su veneno). Las cascabeles diamantadas van cargadas de hemotoxinas, aunque no tan concentradas como el de algunas serpientes. Compensan esta supuesta carencia con su potente modo de entrega, inyectando grandes cantidades de veneno con ataques individuales y quirúrgicos. Las hemotoxinas inundan el torrente sanguíneo, destruyen las células sanguíneas, dañan tejido, catalizan la hemorragia interna. Sí, las reacciones a las picaduras de víboras varían. Pero, si usted es mordid@ por una cascabel diamantada—especialmente entre marzo y abril—en tres minutos el área alrededor de la herida se enrojecerá e hinchará. Y, si no se trata, su cuerpo no se desviará de un guión bien ensayado: hinchazón masiva y formación de ampollas, disminución constante de la presión arterial, dolor de cabeza, dolor severo, sangre en la orina. No obstruya la circulación o haga incisiones sobre la herida. No tome analgésicos o sedantes. No coma. Retire la ropa o joyas debajo del área. Permanezca inmóvil, mantenga la mordedura debajo del nivel de su corazón. Lo ideal es que consiga que alguien más limpie la herida. Y, lo más importante—malditas fronteras—llame al 9-1-1 o 0-6-6; busque atención médica inmediatamente.

Para la mayoría, la mordedura del alacrán de corteza de Arizona, de un color amarillo translúcido y estriado café, no es letal. Los escorpiones, caminantes nocturnos, buscan escondites frescos, oscuros y húmedos durante el día. Si le pica un escorpión, limpie la herida; no coma nada por ocho horas. Es posible que se recupere. Por el contrario, la mordedura de una araña parda reclusa es literalmente "una fuerza de la naturaleza". Espere escalofríos, fiebre, náusea, vómitos, sarpullido, irritación, y una ulcera que seguirá creciendo a menos que la herida sea extirpada y tratada médicamente. Hallada en la madera seca y en escombros, la araña parda reclusa lleva en su espalda una marca distintiva y oscura en forma de violín.

For most, the bite of the translucent yellow and brown-striped Arizona bark scorpion is not lethal. Scorpions, night-walkers, seek cool, dark, moist hiding places during the day. If you are stung by a scorpion, clean the wound; do not eat anything for eight hours. It is possible for you to recover. In contrast, a brown recluse spider's bite is literally "a force of nature." Expect chills, fever, nausea, vomiting, rashes, hives, an ulcer that will continue to grow in size unless the wound is excised and medically treated. Found in dead wood and debris, the brown recluse carries a dark, violin-shaped trademark on its back.

Los calambres por calor, relativamente leves, son señal de deshidratación y pérdida de sodio. Beba agua, descanse en la sombra, busque agua durante el crepúsculo. La enfermedad por calor es una lesión cuyos síntomas incluyen fatiga, mareos, desmayos, náuseas, vómitos. Repita: Beba agua, descanse en la sombra, busque agua durante el crepúsculo. El agotamiento por calor produce sudoración, piel húmeda y pegajosa, elevación del pulso y la respiración, debilidad, más desmayos, náuseas y vómitos. PARE. Las decisiones que tome a partir de ahora dictarán si vive o muere. La insolación ocurre cuando una persona continúa a pesar del agotamiento. Sobreviene el trauma—colapso físico, pérdida de conciencia, pulso y respiración acelerados, un incremento vertiginoso de la temperatura corporal, desorientación severa, destrezas motoras afectadas, micción involuntaria, pupilas dilatadas. A medida que la insolación progresa, usted experimentará dolor de pecho y brazo, convulsiones, entrará en coma. Usted no estará equipad@ para enfrentar estos síntomas si se presentan. Llame al 9-1-1 o 0-6-6 de antemano.

Heat cramps, relatively mild, signal dehydration and loss of sodium. Drink water, rest in the shade, seek water at twilight. Heat illness is an injury whose symptoms include fatigue, dizziness, fainting, nausea, vomiting. Redux: Drink water, rest in the shade, seek water at twilight. Heat exhaustion produces sweating, clammy skin, increased pulse and respiration rate, weakness, more fainting, nausea and vomiting. STOP. The choices you make from now on will dictate whether you live or die. Heatstroke happens when a person pushes on despite heat exhaustion. Trauma ensues—physical collapse, loss of consciousness, rapid pulse and respiration, a skyrocketing body temperature, severe disorientation, impaired motor skills, involuntary urination, dilated pupils. As heatstroke progresses, you will experience chest and arm pain, convulse, go into a coma. You will not be equipped to deal with these symptoms as they present themselves. Call 9-1-1 or 0-6-6 beforehand.

La cholla, o cactus saltador, se agarra. Con la más leve caricia, un botón de espinas se despega. Quite la cholla de su piel y ropa en incrementos, con una piedra, un palo, un cuchillo: el botón… grandes espinas que quedaron atrás… pequeñas espinas o gloquidias. Las agudas agujas que quedan se saldrán solas en unos cuantos días.

Cholla, or jumping cactus, attaches. A bud of spines breaks off at the slightest hint of touch. Remove cholla from your skin and clothing in increments, with a rock, a stick, a knife: the bud… large spines left behind… small spines or glochids. Needling needles that remain will work themselves out in the days ahead.

Arborescent monocot. “Mothers of the Disappeared.” “I Still Haven’t Found What I’m Looking For.” “Bullet the Blue Sky”: “Put El Salvador through an amplifier.” Seldom free-standing (Mexican as the Irish in the United States), the Joshua tree sends out yellow-green flowers—like flares—in the spring. Mormons referred to the trees—actually shrubs—as “praying plants.” Anthropomorphizing each’s branches, they compared the largest of the yucca evergreens to the Old Testament prophet Joshua as he pointed toward the promised land. Use the Joshua tree’s lightweight wood to splint broken limbs. Chew the plant’s roots for the steroid-like compound released (in cases of allergic reaction or swelling).

Monocotiledónea arborescente. “Mothers of the Disappeared.” “I Still Haven’t Found What I’m Looking For.” “Bullet the Blue Sky”: “Put El Salvador through an amplifier.” Pocas veces se sostienen por si mismos (mexicanos como los irlandeses en los Estados Unidos), el árbol de Josué suelta flores verde-amarillas—como bengalas—en la primavera. Los mormones se referían al árbol—en realidad arbusto—como “las plantas que oran”. Antropomorfizando sus ramas, comparaban el más grande de los árboles de yuca de hoja perenne al profeta Josué del Antiguo Testamento mientras señalaba hacia la tierra prometida. Utilice la madera liviana del árbol de Josué para entablillar extremidades fracturadas. Mastique las raíces de la planta para extraer su compuesto similar al corticoesteroides (en los casos de inflamación o reacción alérgica).

Un antioxidante, un medicamento antiinflamatorio, el ubicuo arbusto de la creosota se rejuvenece a través de un proceso de clonación. Puede vivir hasta 9,000 años y los Tohono O'odham pensaban que era el ser viviente más antiguo de la tierra. Diez años después que la explosión termonuclear de 1962 destruyó veintiún arbustos de la creosota en Nevada, otros veinte retoñaron casi en forma de protesta. "No nos moverán". *Hiroshima mon amour*—sin fantasmas, sin quemaduras, sin sombras. El arbusto de creosota simplemente crece—un artista de la sobrevivencia.

An antioxidant, an anti-inflammatory medication, the ubiquitous creosote bush rejuvenates itself via a cloning process. With a life span upwards of 9,000 years, it is esteemed by the Tohono O'odham to be the oldest living thing on earth. Ten years after a 1962 Nevada thermonuclear explosion destroyed twenty-one creosote bushes, twenty re-sprouted as if to protest, "We will not be moved." *Hiroshima mon amour*—no ghosts, no burns, no shadows. The creosote bush simply grows—a survival artist.

Approach rocky trails with caution, especially in the dark or after a thunderstorm. Falls from steep slopes represent the second leading cause of injury and death in the desert (the first being dehydration). Sacrifice momentum for surer footing. Learn to read the terrain: Anticipate loose rock and gravel. Scramble by way of larger, less mobile boulders. Choose routes with anchoring patches of vegetation. Realize that descent, like dissent, is always more dangerous than ascent. Use a walking stick to test what lies ahead. With others, travel at an angle; avoid walking directly below someone else on steep slopes. (Otherwise, expect what that person knocks loose to impact you). Like rocky trails, desert dunes are walkable. Learn to read this terrain, too. Traverse dune desert arrhythmically—Fremen-style. Eschew dunes' angled leeward bases. Hike their hallowed hollows in between, keeping a lookout for small holes (that mark tunnels) and conical depressions of silt. Or, stay high on the dunes' surreal slopes of coarser, tightly packed sand particles. Peaks provide stability if you resist repeating the dunes' rhythm—a movement, like breath, that rises and falls.

Aproxímese a caminos pedregosos con cautela, especialmente en la oscuridad o después de una tormenta eléctrica. Las caídas de pendientes empinadas representan la segunda mayor causa de heridas y muertes en el desierto (siendo la primera, deshidratación). Sacrifique impulso por un punto de apoyo más seguro. Aprenda a leer el terreno: anticipe rocas sueltas y gravilla. Trepe por piedras más grandes y difíciles de desplazar. Elija rutas con parches de vegetación anclados. Dese cuenta que el descenso, como la disidencia, siempre es más peligroso que el ascenso. Use un bastón para probar lo que está adelante. Con otros, viaje en diagonal; evite caminar directamente debajo de alguien más en cuestas empinadas (Si no, espere que lo que la persona suelte al bajar le impacte.) Al igual que los caminos pedregosos, las dunas del desierto son transitables. Aprenda también a leer este terreno. Atraviese el desierto de dunas arrítmicamente—estilo Fremen. Rehúya las dunas con bases anguladas hacia sotavento. Camine entremedio de sus huecos sagrados, fijándose en los hoyos pequeños (que señalan túneles) y depresiones cónicas de limo. O manténgase en lo alto de las laderas surrealistas de las dunas hechas de partículas de arena ásperas, apretadas. Las cimas brindan estabilidad si usted se resiste repetir el ritmo—un movimiento, como el aliento, que sube y baja.

Tormentas eléctricas, revelaciones, pueden parecer como un consuelo para los que sufren de calor y sed. Pero las lluvias torrenciales, comunes en el DF desde mayo hasta septiembre, pueden ser tan mortales como el sol imperturbable y sin gríngolas en los desiertos de Sonora y Chihuahua. (Cada aguacero, una zona cero.) En minutos, el agua rio arriba baja como un muro de agua y escombros. Inundaciones repentinas, deslazamientos de tierra y monzones devoran. Deslaves en zonas bajas, arroyos y barrancos se desbordan. La temperatura del aire—como los papalotes—cae en picada. Cuando los nubarrones van juntándose, busque terreno elevado, evadiendo cactus altos y árboles, colinas y cadenas montañosas. Relámpagos, siempre a la vanguardia, anticipan el diluvio. Cuando el agua retroceda, entrecruce su obra de arte—patrones en la arena, esculpida como un mapa topográfico o un libro de Braille abierto.

Thunderstorms, a release, might sound like a relief to the overheated and thirsty. But, torrential rains, ordinary from May to September in Mexico City, can be as deadly as the unblinking, blinder-free sun in the Sonoran and Chihuahuan Deserts. (Each aguacero, a ground zero.) Within minutes, drainage upstream comes down like a wall of water and debris. Flash floods, mudslides, monsoons swallow. Low-lying washes, arroyos, canyons overflow. Air temperatures—like kites—nosedive. As storm clouds gather, seek higher ground, avoiding tall cacti and trees, hilltops and ridgelines. Lightning, always the avant-garde, precedes downpours. When the waters recede, criss-cross their artistry—patterns in the sand, sculpted as a topographic map or an open book in Braille.

The desert cold also kills. Unpredictable winter weather includes ice, sleet, hail, and snow. In an ideal world, you'd carry a wind- and waterproof jacket with you. In a less-than-ideal world, use large boulders and rock formations as windbreaks. They retain the sun's heat for hours. Don't lie down on the sandy ground or dig shallow holes into it. Faster than air, sand—a heat sink—will drain warmth, like a liquid, from your body.

El frío del desierto también mata. El impredecible clima de invierno incluye hielo, aguanieve, granizo y nieve. En un mundo ideal, llevaría consigo una chaqueta a prueba de viento y agua. En un mundo menos que ideal, use peñascos y formaciones rocosas para protegerse del viento. Éstas mantienen el calor del sol por horas. No se acueste en el suelo arenoso ni cave hoyos poco profundos en él. Más rápido que el aire, la arena—un disipador—drenará, como un líquido, el calor de su cuerpo.

No entre en pánico. No entre en pánico. Si está demasiad@ cansad@ o desorientad@ para continuar, dese cuente que probablemente no va a estar pensando claramente. El calor revuelve el cerebro como huevos. “Es completamente noble salir de una pose”. Reconozca sus propias limitaciones. Prenda su teléfono. Busque una señal. (Camine sólo si no está dentro de la cobertura. Luego, apague el teléfono para conservar la vida de la batería para salvar la suya. Camine, repita. Camine, repita, hasta que encuentre señal.) Llame al 9-1-1 o 0-6-6. Razone—es mejor vivir para cruzar el desierto mañana que dejar que el desierto le crucifique hoy.

Do not panic. Do not panic. If you are too tired or disoriented to continue, realize that you probably will not be thinking clearly. Heat scrambles the brain like eggs. “It is perfectly noble to come out of a pose.” Know your own limitations. Turn your phone on. Search for a signal. (Walk only if you are not in range. Then, power the phone down to save the battery life to save your own. Walk, retest. Walk, retest, until you secure reception.) Call 9-1-1 or 0-6-6. Reason—it’s better to live to cross the desert tomorrow than to let the desert cross you today.

Keep your body as covered as comfort permits. Sunburn not only inhibits travel, it can aggravate, accelerate heat exhaustion and heatstroke. Sand and rock reflect warmth and light as if you were walking on a metallic liquid or "ashes of time." (Protect your eyes from sun blindness, too!) Keep clothing, loose-fitting, light in weight and color. Think drab—tan, off-white, khaki, olive. Think controlled evaporation (clothing regulates). If presented with the opportunity, soak your shirt in water that's too dirty to drink. The sight or smell of you afterward might be unsavory, but damp clothing will slow down your rate of perspiration. The key here being conservation—conserve your resources whenever possible.

Mantenga su cuerpo tan cubierto como la comodidad lo permita. Las quemaduras de sol no sólo inhiben viajes, pueden agravar, acelerar la deshidratación e hipertermia. La arena y las rocas reflejan el calor y la luz como si se estuviera caminando sobre un líquido metálico o "cenizas del tiempo". (¡También proteja sus ojos de la ceguera por el sol!) Mantenga la ropa suelta, ligera de peso y de color. Piense pardo—canela, blancuzco, caqui, verde oliva. Piense evaporación controlada (la ropa regula). Si se le presenta la oportunidad, empape su camisa en el agua que sea muy sucia para beber. Puede que luego su aspecto o su olor sea desagradable, pero la ropa húmeda desacelerará su grado de sudoración. La clave aquí es la conservación—conserve sus recursos cuando sea posible.

Al fin y al cabo, muchos argumentarían que la naturaleza establece el estándar de neutralidad. A diferencia de los seres humanos, la naturaleza no mantiene lealtades a una nación, familia, negocio, religión. Usted ya sabe que el peligro más grande al que se enfrenta en el desierto no es necesariamente el clima o el terreno. Algunos no tendrán su mejor interés en mente. Los rescatistas se comprometen a asistir a los que necesiten ayuda; aténgalos a cumplir esa promesa. No le confíe su vida a ningun@ otr@ extrañ@.

Ultimately, many would argue that nature sets the standard for neutrality. Unlike human beings, nature maintains no allegiances to nation, family, business, religion. You already know that the greatest danger you face in the desert may not be the climate or the terrain. Some will not have your best interests in mind. Rescue workers pledge to offer assistance to those in need; hold them to that promise. Trust no one else who’s a stranger with your life.

When everything—including this cell phone—fails, build a signal fire in dirt or sand, away from brush and trees. Use dead cacti and mesquite. A fire in the shape of an "X"—the international symbol of distress—needs no translation.

Cuando todo—incluyendo este teléfono móvil—falle, construya una señal de fuego en la tierra o en la arena, lejos de arbustos y árboles. Use cactus muertos y mezquite. Un fuego en forma de "X"—el símbolo internacional de peligro—no necesita traducción.

I.

Dos horas antes o después de medio día, inserte un palo (de un metro de largo) en la tierra.

Marque la punta de la sombra que produzca con una piedra.

Espere veinte minutos.

Marque la segunda sombra con otra piedra.

Dibuje una línea recta de la primera piedra a la segunda, extendiendo la línea otros dos metros. Este es su eje de oeste a este.

Dibuje una línea perpendicular que cruce por el centro del eje.

Párase en la intersección de estas dos líneas. Mantenga el oeste a su izquierda para que mire hacia el norte geográfico.

Escoja un punto de referencia en línea con esa orientación.

Camine hacia ese punto de referencia; repita este ejercicio.

II.

En la noche, busque la Osa Mayor.

Fíjese en las dos estrellas que forman el contorno.

Imagine una línea desde esas estrellas a un punto como cinco veces la distancia entre ellas. Ahí debe encontrar a Polaris, la estrella del Norte, esperándole (al final del mango de la Osa Menor).

Imagine una línea recta entre Polaris y el horizonte. Este es su segundo punto de orientación hacia el norte geográfico.

I.

Two hours either side of noon, insert a stick (one meter long) into the ground.

Mark the tip of the shadow it casts with a stone.

Wait twenty minutes.

Mark its second shadow with another stone.

Draw a straight line from the first stone to the second, extending the line two additional meters. This is your west to east axis.

Draw a perpendicular line that crosses its center.

Stand at the lines' intersection. Keep west to your left to face true north.

Pick a visible landmark in line with that orientation.

Walk to that landmark; repeat this exercise.

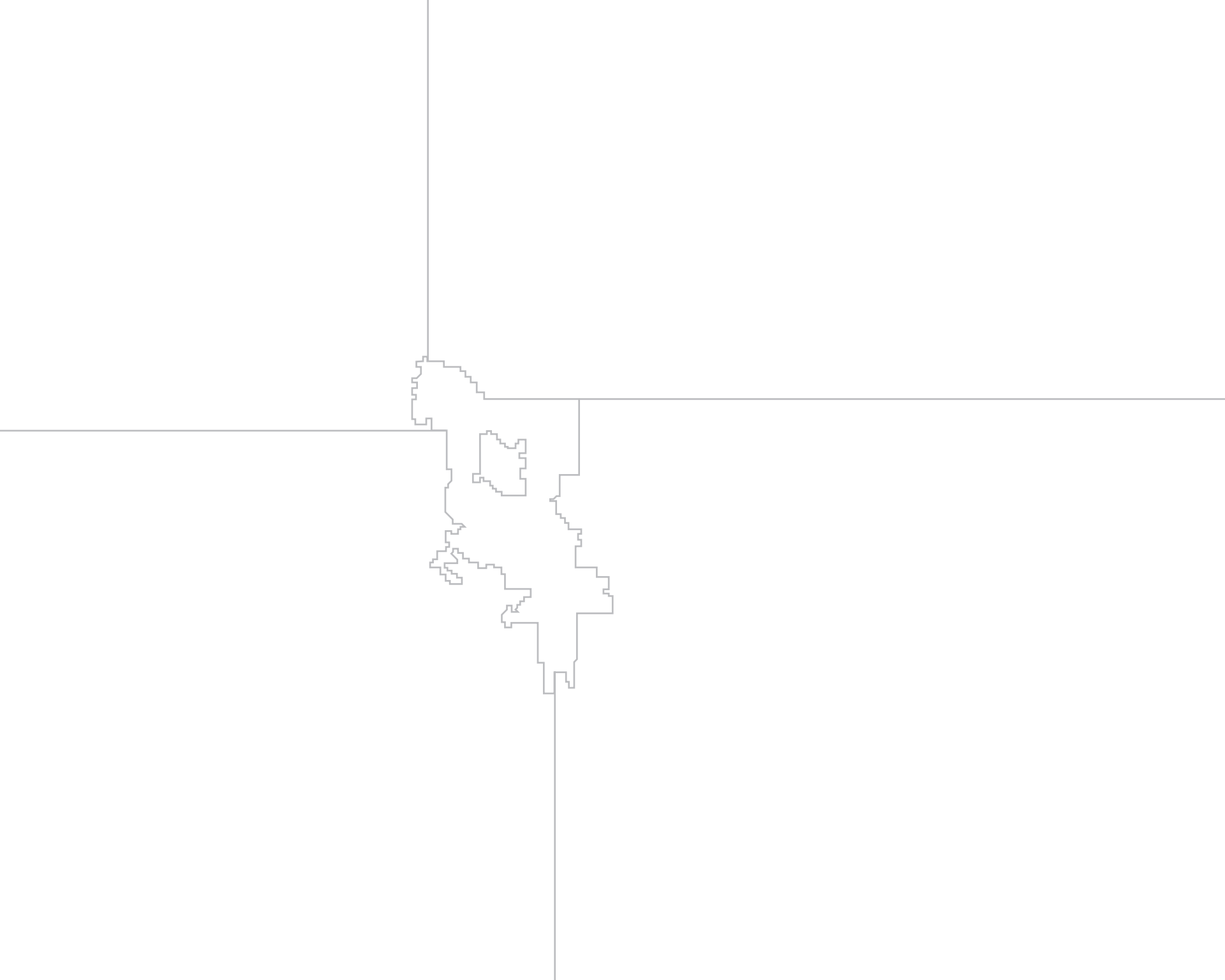
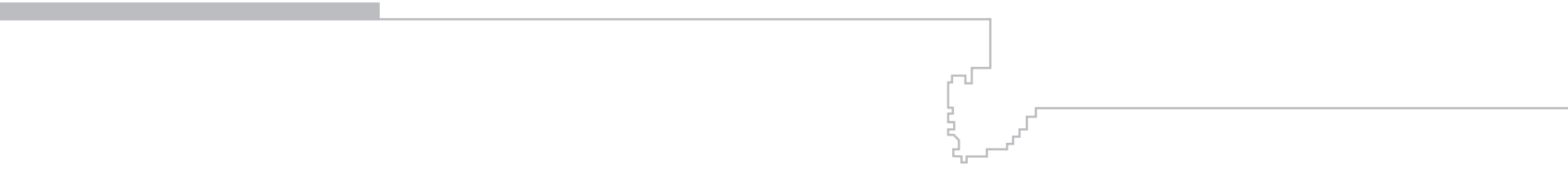
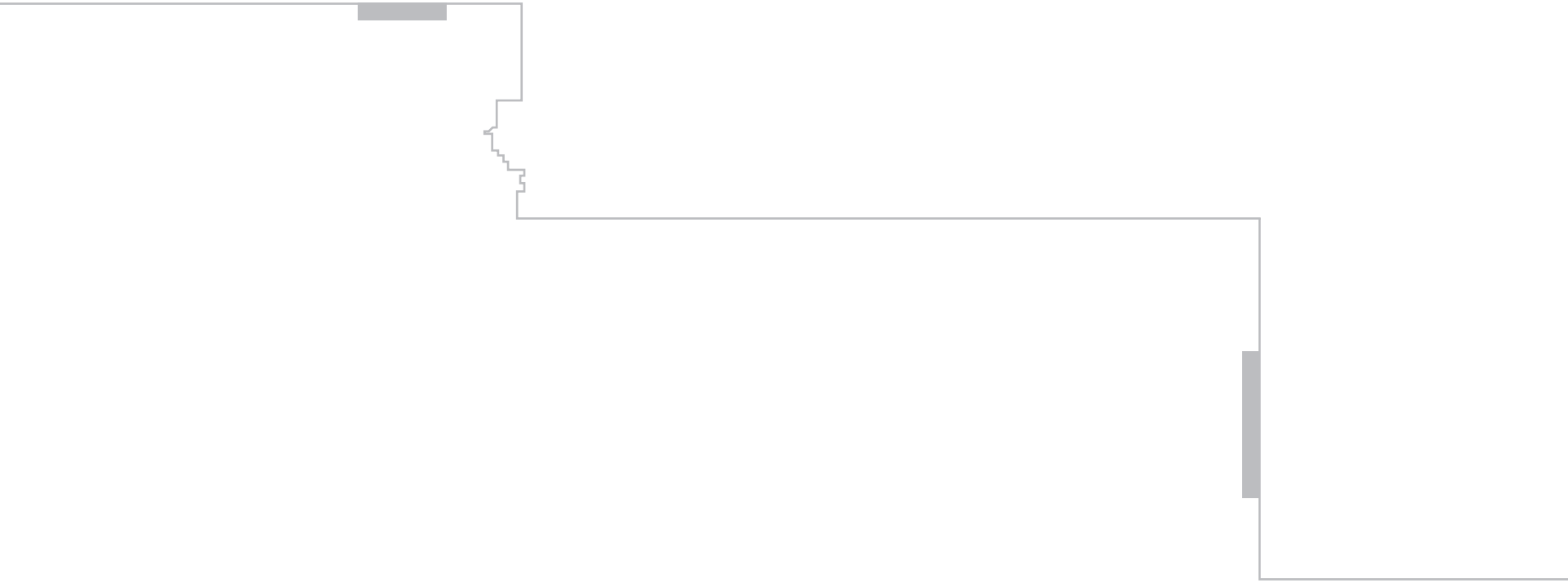
II.

At night, look for the Big Dipper.

Note the two stars that form its outer edge.

Imagine a line from those stars to a point about five times the distance between them. You should find Polaris, the North Star, waiting for you there (the end of the Little Dipper's handle).

Imagine a straight line between Polaris and the horizon. This is your second true north bearing.



The Office of Net Assessment was a publishing collective of students, faculty, and staff at the University of Michigan, Ann Arbor, funded by MCubed from 2012 to 2014.