A Perpetual Calendar Made of LEGO® Parts

Chamberlain Fong

San Francisco, California, USA; chamberlain@alum.berkeley.edu

Abstract

The design of a perpetual calendar made completely out of standard LEGO® parts is presented and discussed. This desk calendar can be adjusted to represent any month in the Gregorian calendar. In addition, there is an ancillary discussion on calculating probabilities in the Gregorian calendar.

Introduction

During the great pandemic of 2020, I was stuck at home like most of the world. I decided to spend this time cooped up at home playing with LEGO®. Meanwhile, it was during this time that the eminent mathematician John Horton Conway passed away from complications of COVID-19. I have long been a fan of Conway's extensive body of work, especially his Doomsday algorithm [1,2,9] for calculating the day of the week for any date in the calendar. This directly inspired me to design and build a perpetual calendar out of LEGO®.

My perpetual calendar is shown in Figure 1 set to two different months. The configuration on the left is that of July 1998. This was the calendar month during the inaugural Bridges conference [7] in Winfield, Kansas USA. The specific dates for this historic event were July 28–30, 1998. The configuration on the right is that of February 2021. This is the hitherto current calendar month at the time of submission and review of this paper. The photographs in Figure 1 do not really do justice in showcasing the functionality and kinetic nature of the perpetual calendar. Please watch the accompanying video to get a better look of it in action: https://youtu.be/_Uz9VTIpgqE

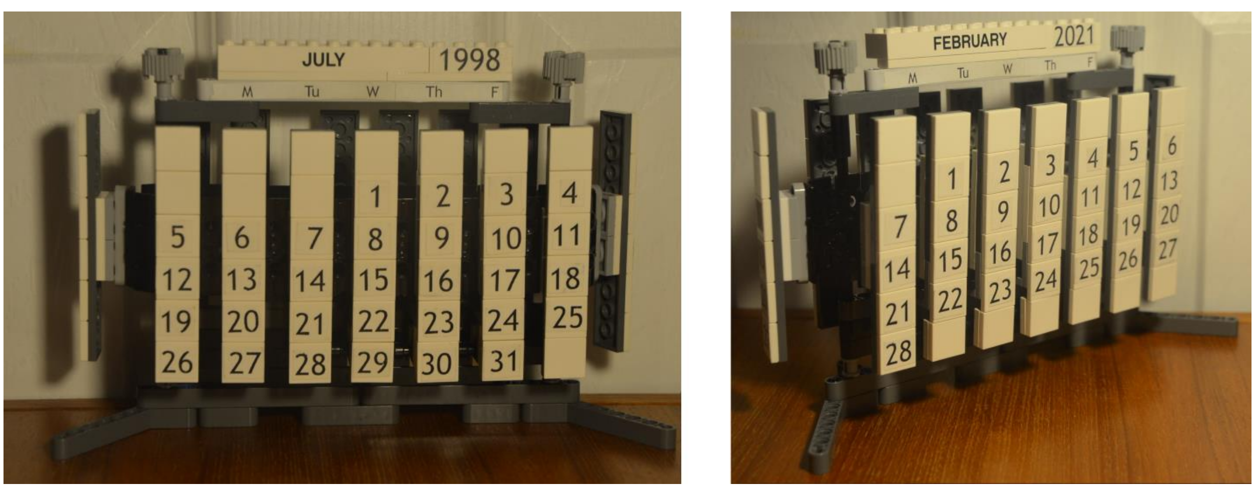

Figure 1: *A LEGO-based perpetual calendar configured to July 1998 and February 2021*

At the core of the perpetual calendar is a conveyor belt carrying digits that represent the days in the month. There are two knobs on top of the contraption that allow the user to scroll the conveyor belt. This simple scrolling mechanism enables adjustment of the device to represent any month in the calendar. Moreover, there are additional components in this contraption to turn on or turn off the digits representing the 29th, 30th, and 31st days, which are not present in some months such as February this year. The internal structure of this perpetual calendar will be discussed in more detail later in this paper.

The Gregorian Calendar

The *Gregorian calendar* is the de facto calendar system currently in use worldwide. This calendar is named after Pope Gregory XIII who mandated its use in 1582. It is a slight alteration of the *Julian calendar* system which dates back to 46 B.C. from Julius Caesar. The Julian calendar has all the trappings of our modern calendar including the familiar 12 months along with the haphazard number of days in each month. In the Julian calendar, years that are divisible by 4 are leap years, which have an extra day falling on February 29th. The Gregorian calendar amended this rule so that a year divisible by 100 is only a leap year if it is also divisible by 400. For example, the year 2000 was a leap year, but the year 1900 was not, because when divided by 400, it does not give a whole number.

The reasoning behind the Gregorian calendar reform was to have a more accurate calendar [3,4] in sync with the seasons on Earth. Astronomically speaking, Earth spins around its axis approximately 365.24219 times annually. This rate is known as the *mean tropical year*. The true number of days in the mean tropical year actually varies slightly over time. The Julian calendar approximates the mean tropical year as 365.25 days. In contrast, the Gregorian calendar gives a better approximation of the mean tropical year as 365.2425 days.

Over the years, there have been many proposals to improve upon and replace the Gregorian calendar. One notable example of proposed calendar reform is the *Hanke-Henry permanent calendar* [5] by economist Steve Hanke and astronomer Richard Conn Henry. In the Hanke-Henry calendar, every date falls on the same day of the week, year after year. This property is achieved by reducing the common year to 364 days and adding a leap week every 5 or 6 years. Note that the number 364 is divisible by 7. However, in my opinion, it is unlikely for any calendar reform to happen in the near future. For one thing, such a change is very political and would require the whole world to agree on a drastic alteration of the calendar. Also, it will require modifications in the timekeeping mechanism of a significant number of software programs. This will be like the Y2K problem on steroids. The Gregorian calendar is here to stay.

Canonical Calendar Layout. The Gregorian calendar has a standard layout for displaying the full month. In this layout, the days of the month are arranged in a 6x7 table. This table comes with a top banner showing the days of the week sequentially from Sunday to Saturday. This layout is pretty much standard and universal in its use, be it in wall calendars or in computer software. The perpetual calendar shown in Figure 1 follows this convention.

Although each month can have at most 31 days, a grid with 6x7 entries is necessary to fit all the days in the canonical table layout. There are some configurations where a 5x7 grid is insufficient. This usually occurs when the 1st day of the month falls on a Friday or Saturday. To illustrate this, Figure 2 enumerates all the 28 possible configurations that can occur in the canonical calendar layout. There are three configurations that require a grid of 6x7 cells because they will not fit in a 5x7 grid. These three configurations are shown at the bottom left corner of Figure 2 enclosed in a dashed outline. Some calendars work around this problem by sharing calendar days like the 24th and the 31th in the same grid cell. However, such shared grid cells are inelegant and disallowed in the design of my perpetual calendar.

It is relatively easy to show why there are 28 distinct calendar configurations. Each calendar month can be classified by its number of days. There are 4 possibilities for this: 28, 29, 30, and 31. In addition, each month can be classified by the day of the week it starts with. There are 7 possibilities for this, corresponding to each day of the week. Since these two classes are independent of each other, there are a total of 4*7 different calendar configurations. Every month appearing in the calendar is equivalent to one of these 28 configurations. Months with equivalent configurations are interchangeable with each other.

The distribution of the 28 monthly configurations does not come evenly in a standard calendar. Some configurations are more likely to occur than others. For example, a month starting on a Monday with 31 days is much more likely to occur than a similar month with only 29 days. In fact, months with exactly 29 days only occur during February of leap years, which make them quite rare. Furthermore, configurations within a single vertical file of Figure 2, such as those months with 31 days, also do not come with equal likelihoods. This phenomenon will be discussed in more detail near the end of this paper.

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29
```

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
```

```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
```

```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29
```

```
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28
```

```
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

```
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

```
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29
```

```
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28
```

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29
```

```
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28
```

```
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

```
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

```
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29
```

```
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28
```

```
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

```
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

```
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

```
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28
```

```
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

```
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

```
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
```

```
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28
```

Figure 2: *The 28 different calendar configurations*

Finding Patterns in the Calendar. Even though the days, weeks, and months of the Gregorian calendar appear irregular and haphazard, there are actually some patterns in it. One striking pattern in the calendar appears when examining the number of days in each month within a year. If leap years are ignored, the sequence of day counts goes like this: 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31. There are 7 months in the year with 31 days. These are January, March, May, July, August, October, and December. Notice that every other month in the year has 31 days except for a discrepancy with July and August. July and August are two consecutive months that both have 31 days. This makes them the exception to the alternating pattern of day counts for months within a year. These two exceptional months can be called Caesarean months because both are named after Roman emperors: Julius Caesar and Augustus Caesar.

Meanwhile, Conway exploited the alternating pattern to come up with effective mnemonics for his Doomsday algorithm. Specifically, Conway [2,9] noticed that April 4, June 6, August 8, October 10, and December 12 always fall on the same day of the week as one another for any year. This is a consequence of the aforementioned alternating pattern; and can be verified by counting the number of days in between the given dates. Conway's observation can be summarized by these equations:

$$\text{for the months of April and June:} \quad (30-n)+31+(n+2)=63 \quad \equiv\ 0 \pmod{7}$$
$$\text{for August, October \& December:} \quad (31-n)+30+(n+2)=63 \quad \equiv\ 0 \pmod{7}$$

Perpetual Calendars. A perpetual calendar is a calendar that is valid for many years. People usually use it to look up the day of the week for a given date in the past or in the future. Perpetual calendars are not new and are probably as old as the calendar itself. There have been many different designs of perpetual calendars. The most common ones use some sort of sliding mechanism to adjust a table of calendar digits that align with a top banner containing the days of the week. Figure 3 shows some examples of perpetual calendars. At the left is a design from the Museum of Modern Art in New York. It consists of two slabs of plexiglass that slide against each other to produce different calendar configurations. At the center is a rotary type of perpetual calendar that is adjusted by rotating a piece of laminated cardboard over another. The last one is a calendar that uses a sliding piece of black plastic to adjust its configuration. This calendar is made and sold by the Container Store. Note that all these examples are strictly mechanical devices. Electronic calendars which use microprocessors, pixelated displays, and external power sources are beyond the scope of this paper.

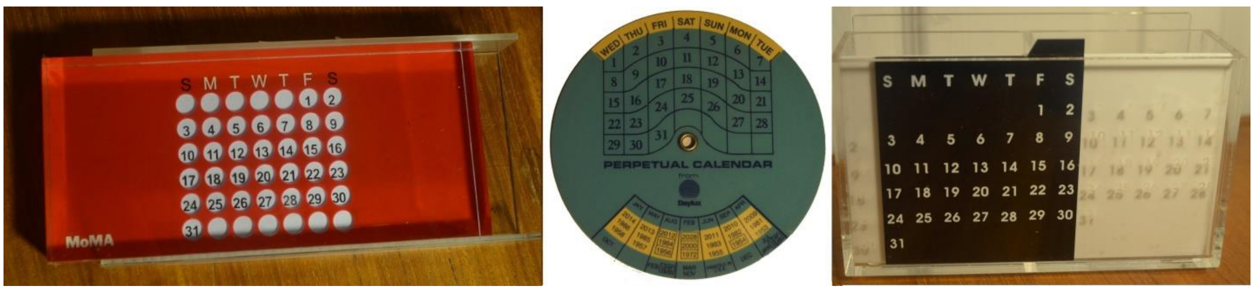

Figure 3: *Some examples of perpetual calendars*

Conditional Days: 29, 30, 31. In order to faithfully represent a calendar configuration, it is ideal to display the correct number of days in a month. This means that the calendar should only display the 29th, 30th, and 31st days if the month in question actually has those days. However, most designs of mechanical perpetual calendars just ignore the number of days in a month. This is evident in all of the calendars in Figure 3, which display 31 days regardless of the month. On the other hand, my perpetual calendar shown in Figure 1 has additional mechanisms that allow it to represent February 2021 with the correct number of days. This is the main innovation of this paper.

Although my perpetual calendar can represent any month in the Gregorian calendar, it has no internal mechanism that allows the user to determine which calendar configuration to use for a given month. For this, the user still has to look up or calculate the day of the week for the 1st day of the month. Of course, Conway's Doomsday algorithm is perfect for such a task! In a way, my perpetual calendar can be considered as a complementary device useful for displaying results of the Doomsday algorithm.

LEGO® Bricks and Beams

The Bridges community of mathematical artists has a long history of using interlocking plastic parts to form geometric shapes. There have been several papers about Zometool [6] and Polydron [10]. LEGO® is not particularly popular among mathematical artists. Although LEGO® has appeared in the Bridges proceedings [8], it has largely been ignored by mathematical artists. One of the goals of this paper is to promote LEGO® as a possible medium for geometric and mechanical modeling.

Why LEGO®? LEGO is among the most popular toy brands in the world, which makes it quite ubiquitous. Most retail stores that sell toys carry LEGO® products. LEGO's popularity and success has enabled it to amass a very large database of parts. In fact, there are currently about 4000 different parts in its inventory. This makes it quite versatile in modeling a wide variety of objects, including perpetual calendars. Also, there is a large second hand market where used LEGO® parts can be purchased relatively cheap. This makes it quite suitable and economical for rapid prototyping.

The LEGO® *brick* is the most basic building block of the toy line. It was first patented in January 1958. Over the years, it has been altered and refined to many variations of different color, size, and measurements. One important variant is the LEGO® *plate* which comes at ⅓ of the thickness of the brick. Another important variant is the LEGO® *tile* which is essentially a plate with no connector studs on top. These constituent parts are all used in the perpetual calendar presented in this paper. To be specific, the calendar digits representing days of the month were all printed on top of tiles mounted on a central conveyor belt. These tiles serve as the facade of the perpetual calendar.

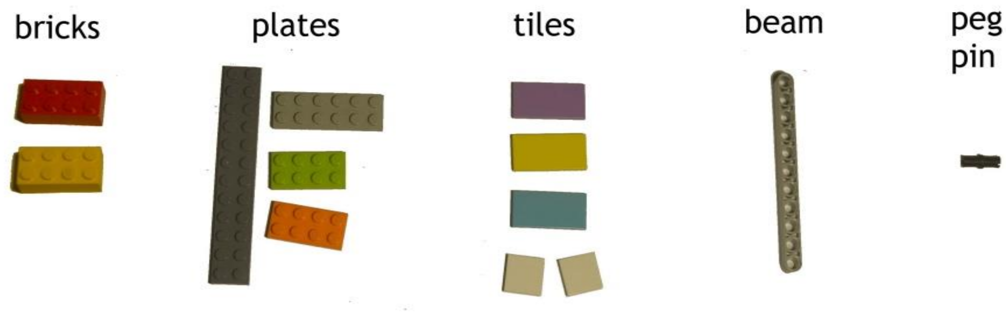

Figure 4: *Some common LEGO® parts*

LEGO® Technic. In 1977, LEGO® introduced an advanced line of toys that are capable of elaborate and complicated motions. Prior to this, LEGO® toys were mostly made up of interconnected bricks that form static shapes akin to sculpted blocks. Toys in the Technic line are dynamic and exhibit rich technical functionality. These toys consist of interconnecting rods instead of the familiar LEGO® bricks. The basic building block in this toyline is a plastic rod known as the LEGO® *beam*. Beams connect with other beams using a pivoting part known as the *peg pin*. Figure 4 shows some of these Technic parts. Other common Technic parts include gears, ball joints, and axles.

Technic is now a mainstay of the LEGO® product line. When LEGO® introduced its own robotics toyline in 1998, Technic parts were heavily used as constituent parts in the robotic kits. Consequently, the Technic toyline is particularly popular among students of mechanical engineering. For my perpetual calendar, Technic parts are used in the chassis frame that serves as the base stand holding the conveyor belt in place.

Treads and Sprockets. Even more advanced parts from the LEGO® Technic line are needed for my perpetual calendar. These come in the form of LEGO® *treads* and *sprockets*. These parts were first introduced in 2007 for use in toy vehicles navigating rugged terrain. Examples of such vehicles include bulldozers, excavators, and armored tanks. More recently, LEGO® designers have successfully repurposed these parts to serve as a conveyor belt.

Treads are thin rectangular elements that can be linked together to form a long chain. Sprockets are gear-like parts that mesh together with treads in order to propel them in motion. Figure 5 shows treads and sprockets as well as their use in a toy vehicle. Note that the bottom of the toy has continuous caterpillar tracks made of treads used for vehicular locomotion. Treads and sprockets are key elements in the conveyor belt at the heart of my perpetual calendar.

Figure 5: *Treads & sprockets along with their use in a toy snow groomer*

Design Details

My perpetual calendar is made of 3 main components. These are the chassis, the conveyor belt, and the individual numeric columns. The chassis is built out of interconnected LEGO® beams. It also includes two spinning axles with sprockets attached. The conveyor belt is a looped chain of 31 treads driven by sprockets on the chassis. There are several numeric columns that mount on top of the conveyor belt. These numeric columns are used to display individual days on the calendar. Figure 6 shows a picture of the chassis on the left along with different views of a numeric column.

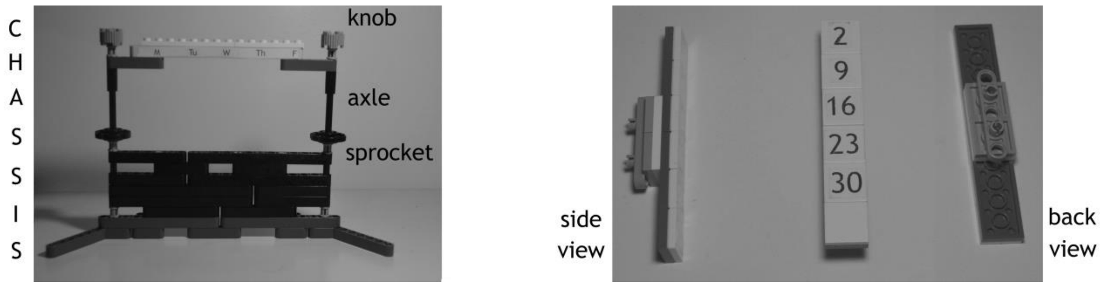

Figure 6: *Some components of the perpetual calendar*

***Modulo 7 Numeric Columns*.** A total of 15 numeric columns serve as the front-facing part of the calendar. These columns are primarily made up of LEGO® plates along with white square tiles that have printed digits. Each column has 6 tiles that represent cells in the calendar grid. The full layout of 15 columns mounted on top of the conveyor belt is shown in Figure 7. Of course, only 7 of these columns are visible in the front part of the calendar at any one time.

Note that all the numbers printed on a single column are congruent to each other in modulo 7 arithmetic. For example, the single column in Figure 6 contains the numbers: 2, 9, 16, 23, and 30. All these numbers give a remainder of 2 when divided by 7. This phenomenon follows from the fact that all these numbers fall on the same day of the week, and must therefore differ from each other by multiples of 7. Each column in the calendar behaves as a cohesive unit that stays together regardless of the calendar configuration. Furthermore, the design of each column is essentially identical except for 3 in the middle of Figure 7. These middle 3 columns contain the conditional days of the month at the bottom: 29, 30, and 31. Not all months in the calendar go up to these many days. Hence, my perpetual calendar has extra components to allow for these days to be turned on or turned off depending on the calendar month.

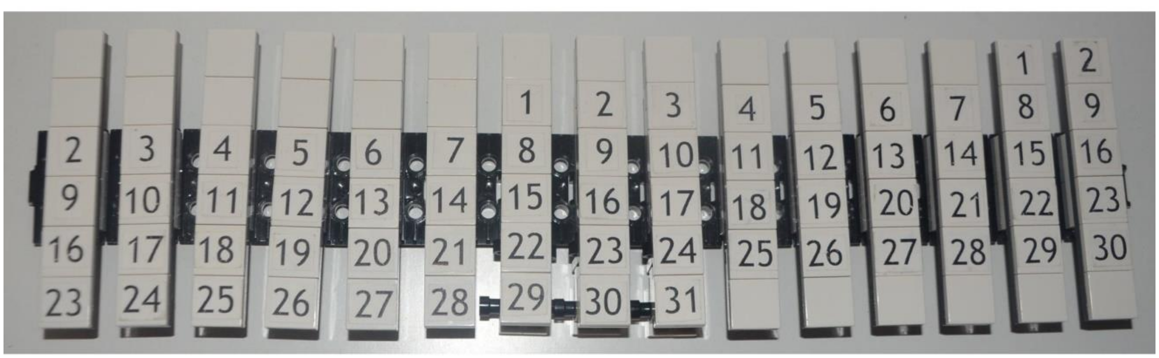

Figure 7: *Full layout of numeric columns on the conveyor belt*

Swiveling Bottom Part. In order to faithfully represent months that do not go up to 31 days, additional mechanical elements have been added to 3 numeric columns in my perpetual calendar. These come in the form of a swiveling part that attaches at the bottom of columns with the 29th, 30th, and 31st days. The user can spin these parts around to activate or deactivate them. This spinning action is demonstrated in Figure 8. Each swiveling part consists of a stack of plates and tiles that are attached to a pivoting pinhole used for rotation.

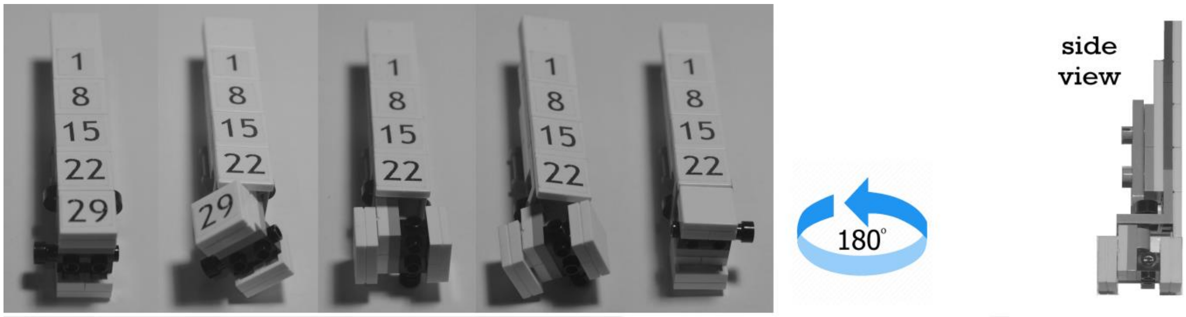

Figure 8: *A swiveling bottom part for representing a conditional 29th day*

Why Halloween is More Likely to Fall on a Saturday than a Sunday

This penultimate section will discuss some probability quirks in the calendar. The Gregorian calendar was designed to be periodic over a span of 400 years. This means that after 400 years, the pattern of days, weeks, months, and years repeats. For this to be true, the number of days within a period of 400 years must be divisible by 7. Otherwise, the weekly pattern cannot be periodic. For example, the periodicity of the Gregorian calendar necessitates that the day after Christmas in 1937 and 2337 must fall on the same day of the week. Indeed, it can be checked that both 12/26/1937 and 12/26/2337 fall on a Sunday.

Calculating the total number of days in the span of 400 years is relatively easy, but one has to be careful with leap years. The leap year rule of the Gregorian calendar designates the year 2000 as a leap year, but not for the years 2100, 2200, and 2300. In the span of 400 years, exactly 97 are leap years. Hence, the total number of days within a 400 year period is $97 * 366 + 303 * 365 = 146097$. This number is indeed divisible by 7. Furthermore, $^{146097}/_{400}$ = 365.2425, i.e. the Gregorian year.

It is natural to wonder about the probability that a calendar day will fall on a certain day of the week. For example, consider the likelihood that Halloween will fall on a Saturday. Intuitively, one might guess that the answer is $1/7$. This is a good approximation, but it is incorrect. In fact, it is easy to show that the probability cannot be exactly $1/7$. Recall that Gregorian calendar has a period of 400 years. In that span of time, there must also be 400 Halloweens. Each of these 400 Halloweens can be classified by its day of the week. Since the number 400 is not divisible by 7, the days of the week for Halloween in this time span cannot be uniformly distributed.

A computer program was written in C++ to count the days of the week for Halloween over any 400 year period. The central subroutine of the program repeatedly performs Conway's Doomsday algorithm 400 times. The results are listed below.

Sunday count	Monday count	Tuesday count	Wednesday count	Thursday count	Friday count	Saturday count
56	58	56	58	57	57	58

This tabulation shows that the distribution is not uniform within the 400 year period.

If one randomly picks a Halloween day over a 400-year period, the probability that it will fall on a Saturday is $^{58}/_{400}$. This follows directly from the tabulated values above. Likewise, the probability that Halloween will fall on a Sunday is $^{56}/_{400}$. Therefore, Halloween is more likely to fall on Saturday than a Sunday; i.e.

$$\frac{58}{400} > \frac{1}{7} > \frac{56}{400}$$

In terms of percentages, the main inequality can be rephrased as $14.5\% > 14\%$. This is a small but non-negligible difference that could be exploited by gambling houses. Table 1 shows a distribution table for other noteworthy days in the year. Note that the numbers in each row of the table have a sum total of 400, except for the last row which has a total of 97. This is expected because the last row corresponds to the Leap day.

Table 1: *Table of recurrences for certain days over a 400 year period*

Day	Description	Sunday count	Monday count	Tuesday count	Wednesday count	Thursday count	Friday count	Saturday count
01/01	New Year's day	58	56	58	57	57	58	56
02/14	Valentine's day	58	56	58	56	58	57	57
03/14	π day	56	58	56	58	57	57	58
07/04	Independence day (USA)	56	58	56	58	57	57	58
10/31	Halloween	56	58	56	58	57	57	58
12/25	Christmas day	58	56	58	57	57	58	56
12/26	Conway's birthday	56	58	56	58	57	57	58
02/29	Leap day	13	15	13	15	13	14	14

Summary

An innovative perpetual calendar build out of LEGO® parts is presented. This device uses a central conveyor belt along with several swiveling parts to represent any month in the Gregorian calendar. The device can be considered as a tangible physical complement to Conway's Doomsday algorithm. A list of parts for this calendar is available in
https://doomsdayrule.blogspot.com/2020/09/a-perpetual-calendar-made-of-lego-parts.html

In Memoriam: John Horton Conway (1937-2020)

References

[1] S. Abdali. "Finding the Year's Share in Day-of-Week Calculations." *Recreational Mathematics Magazine*, volume 3, issue 6, December 2016.

[2] J. Conway. "Tomorrow is the Day After Doomsday." *Eureka*, October 1973, pp. 28-32.

[3] C. Fong. "What Day is Doomsday? How to Mentally Calculate the Day of the Week for Any Date." *Scientific American*, October 2011.
https://www.scientificamerican.com/article/calendar-algorithm/

[4] C. Fong, M. Walters. "Methods for Accelerating Conway's Doomsday Algorithm." *International Congress on Industrial and Applied Mathematics*, Vancouver Canada, July 2011.

[5] S. Hanke, R. Henry. "The Hanke-Henry Permanent Calendar."
http://hankehenryontime.com/html/calendar.html

[6] D. Richter, S. Vorthmann. "Green Quaternions, Tenacious Symmetry, and Octahedral Zome." *Bridges Conference Proceedings*, London UK, August 4–8, 2006 , pp. 429–436.

[7] R. Sarhangi, B. Martin. "The Circle: A Paradigm for Paradox." *Bridges Conference Proceedings*, Winfield Kansas USA, July 28–30, 1998 , pp. 93–111.

[8] C. Sequin, M. Galemmo. "LEGO® Knots." *Bridges Conference Proceedings*, Seoul South Korea, August 14–19, 2014 , pp. 261–270.

[9] B. Torrence, E. Torrence, C. Mulcahy. "John H. Conway – Doomsday." *Mathematics Awareness Month*, April 2014.
https://www.youtube.com/watch?v=T_nQG-Bzxsg

[10] T. Verhoeff, K. Verhoeff. "Folded Strips of Rhombuses, and a Plea for the $\sqrt{2}:1$ Rhombus." *Bridges Conference Proceedings*, Enschede The Netherlands, July 27–31, 2013 , pp. 71–78.