

Programovanie virtuálneho  
objektu prístupné  
pre nevidiacich žiakov  
sekundárneho vzdelávania

Rebeka Sojka

# Diplomová práca

- Školiteľka: doc. RNDr. Ľudmila Jašková, PhD.
- Cieľ práce: programovacie prostredie pre nevidiacich; pohyb virtuálneho robota po ozvučenej štvorcovej sieti
- Cieľová skupina: 2. stupeň ZŠ

# Podobné aplikácie

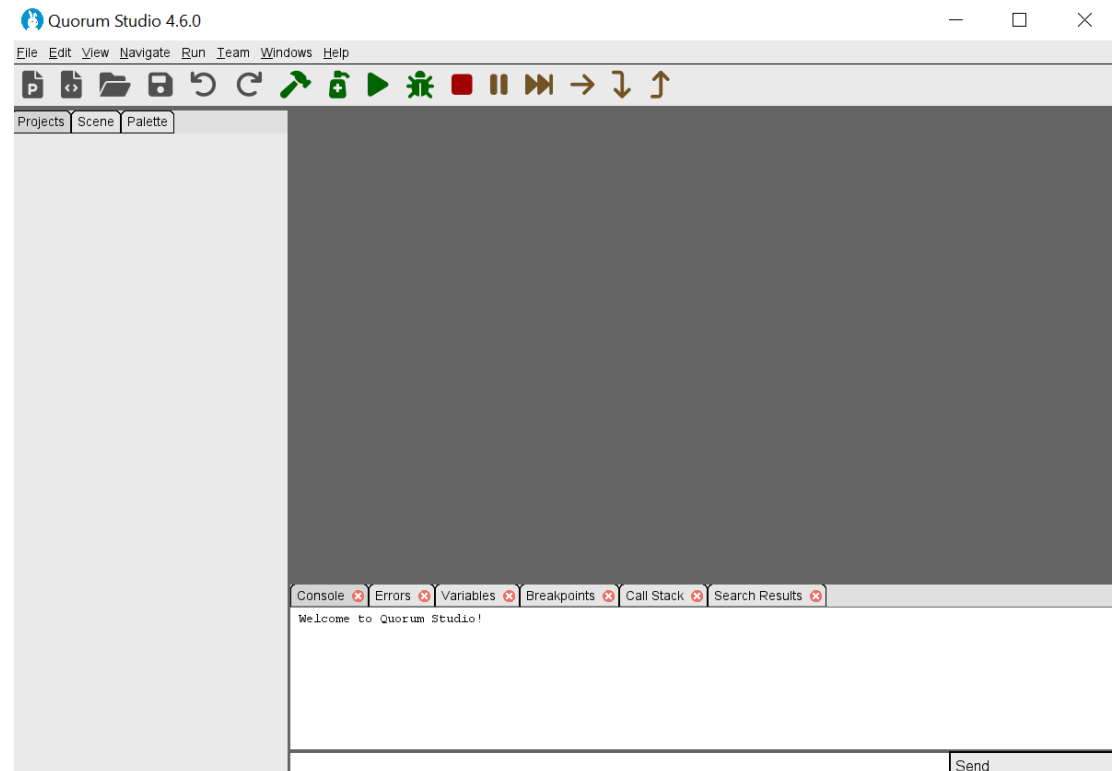
Quorum

Zvukové príbehy

Coshi 2019

# Quorum (jazyk aj IDE)

- Online tutoriál
- Stiahnutelné prostredie
- Typový jazyk
- S knižnicami



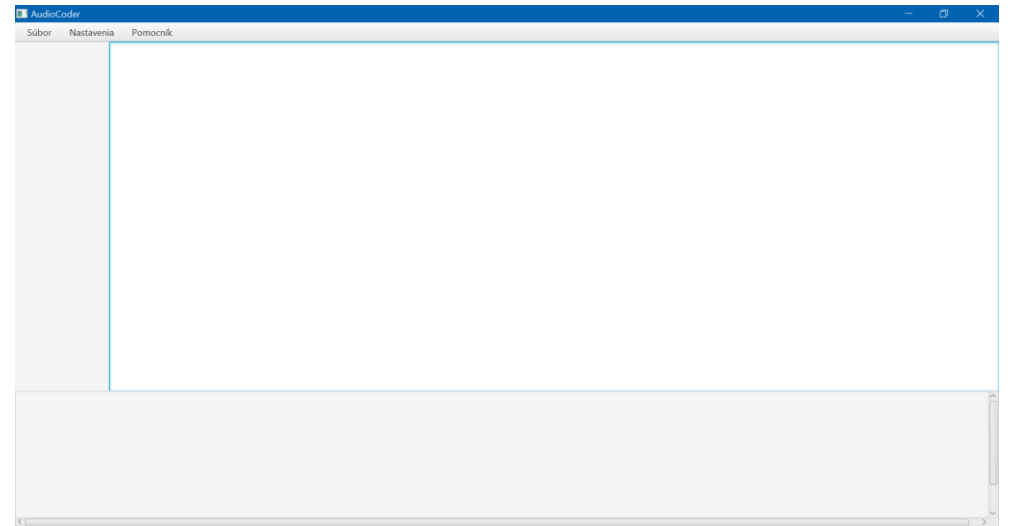
# Quorum

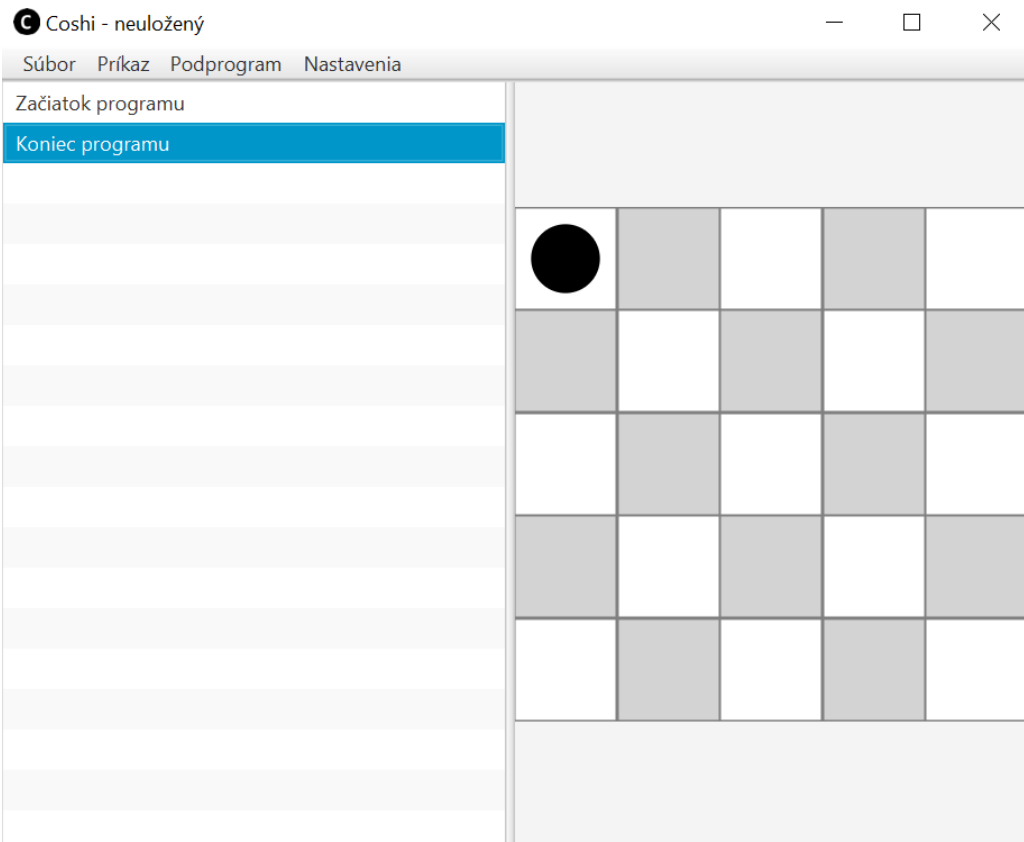
- Jednoduchá syntax
- Podpora hovoreného výstupu
- Popisy komentárov a kódu
- Funkcie pre prácu s hlasovými príkazmi
- Vysokoúrovňová abstrakcia
- Rozšíriteľnosť

```
1 class Main
2     action Main
3         integer i = 15
4         //integer j = 25
5         //integer k = 35
6         output i
7     end
8 end
```

# Zvukové príbehy, diplomová práca

- Autorka: Ivana Nemsilajová
- ozvučené programovacie prostredie
- zvukové príbehy pozostávajúce zo sekvencie hovoreného slova a nahratých zvukov





# Coshi, 2019

---

- Skladanie programu z hotových príkazov
- Bez terminálu
- Syntax jazyka
- Navigácia v kóde
- Klávesové skratky
- Pre nižšie ročníky

# Denník práce

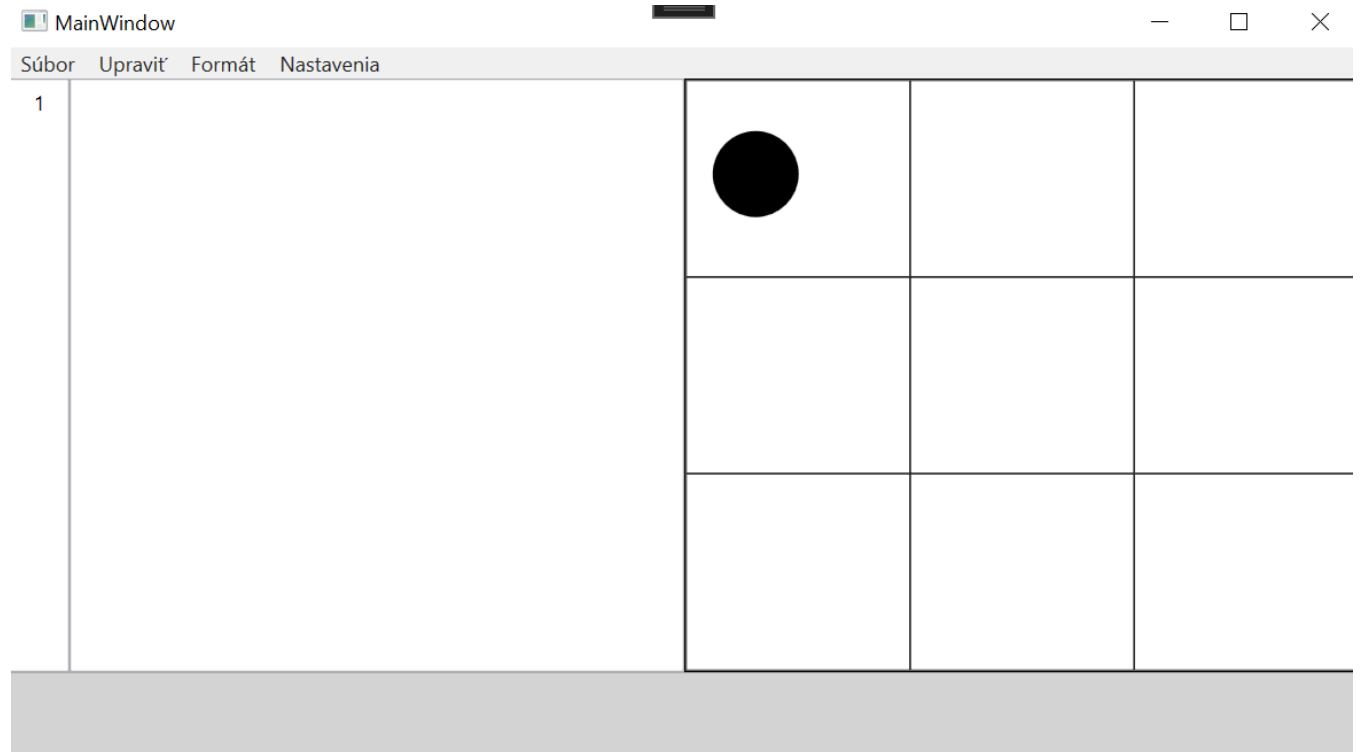
1. Malá testovacia aplikácia
2. Syntax jazyka
3. Literatúra
4. Interpreter pre základné príkazy a cykly
5. Terminál
6. wxPython -> Java -> WPF C#



# Implementácia - GUI

- WPF, XAML

```
<Window x:Class="coshi2.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:coshi2"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800"
KeyDown="Window_KeyDown"
Name="Root"
>
<Grid>
  <Grid.RowDefinitions...>
  <Menu Grid.Row="0">
    <MenuItem Name="subor_volba" Header="_Súbor"...>
    <MenuItem Header="Upraviť"...>
    <MenuItem Header="_Formát"...>
    <MenuItem Header="_Nastavenia">
      <MenuItem Header="Veľkosť plochy"...>
      <MenuItem Header="_Pomoc"/>
    </MenuItem>
  </Menu>
  <Grid Grid.Row="1">
    <Grid.ColumnDefinitions...>
    <ScrollViewer VerticalScrollBarVisibility="Auto"...>
    <Grid x:Name="graf_plocha" Grid.Column="1">
      <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
      </Grid.RowDefinitions>
      <Border Grid.Row="1" BorderBrush="Black" BorderThickness="1">
```



# Implementácia

- Thready
- Klávesové skratky
- Interpreter
- Základné príkazy
- Cyklus (for i)
- Terminál

```
11 references
public void scan()
{
    while(char.IsWhiteSpace(look))
    {
        if (look == '\n') {
            this.lineNumber += 1;
        }
        next();
    }
    this.token = "";
    this.position = this.index - 1;
    if (Char.IsNumber(this.look))
    {
        while (Char.IsNumber(this.look))
        {
            this.token += this.look;
            this.next();
        }
        this.kind = this.NUMBER;
    }
    else if (Char.IsLetter(this.look))
    {
        while (Char.IsLetter(this.look))
        {
            this.token += this.look;
            this.next();
        }
        this.kind = this.WORD;
    }
    else if (this.look != this.KONIEC)
    {
```

```
7 references
public void next()
{
    if (this.index >= this.input.Length)
    {
        this.look = this.KONIEC;
    }
    else
    {
        this.look = this.input[this.index];
        this.index++;
    }
}
```



# Ukážka



# Literatúra

# Making Programming Accessible to Learners with Visual Impairments: A Literature Review

*„When teaching with text-based programming languages, the choice of language is important. Either choose a language that is specially designed for VI learners, or a general-purpose language with simple syntax and limited use of non-alphanumeric characters, for example Ruby. “*

## Authors:



**Alex Hadwen-Bennett**  
King's College London



**Sue Sentance**  
King's College London



**Cecily Morrison**

International Journal of Computer Science Education in Schools, April 2018, Vol. 2, No. 2  
ISSN 2513-8359

## Making Programming Accessible to Learners with Visual Impairments: A Literature Review

Alex Hadwen-Bennett<sup>1</sup>  
Sue Sentance<sup>1</sup>  
Cecily Morrison<sup>2</sup>

<sup>1</sup>King's College London  
<sup>2</sup>Microsoft Research Cambridge

DOI: 10.21585/ijceses.v2i2.25

### Abstract

Programming can be challenging to learn, and for visually impaired (VI) learners, there are numerous additional barriers to the learning process. Many modern programming environments are inaccessible to VI learners, being difficult or impossible to interface with using a screen reader. A review of the literature has identified a number of strategies that have been employed in the quest to make learning to program accessible to VI learners. These can be broadly divided into the following categories; auditory and haptic feedback, making text-based languages (TBLs) accessible, making block-based languages (BBLs) accessible and physical artefacts. A common theme among the literature is the difficulty VI learners have in gaining an understanding of the overall structure of their code. Much of the research carried out in this space to date focuses on the evaluation of interventions aimed at VI high-school and undergraduate students, with limited attention given to the learning processes of VI learners. Additionally, the majority of the research deals with TBLs, this is despite the fact that most introductory programming courses for primary learners use BBLs. Therefore, further research is urgently needed to investigate potential strategies for introducing VI children in primary education to programming and the learning processes involved.

# Mini-languages: A way to learn programming principles

*„General-purpose languages provide little leverage for understanding their basic actions and control structures. The languages are not visual and their basic functions are carried out behind an opaque barrier. Professional programming environments do not usually provide any visualization capabilities. With the process of program execution hidden, the student develops an input-output orientated understanding.“*

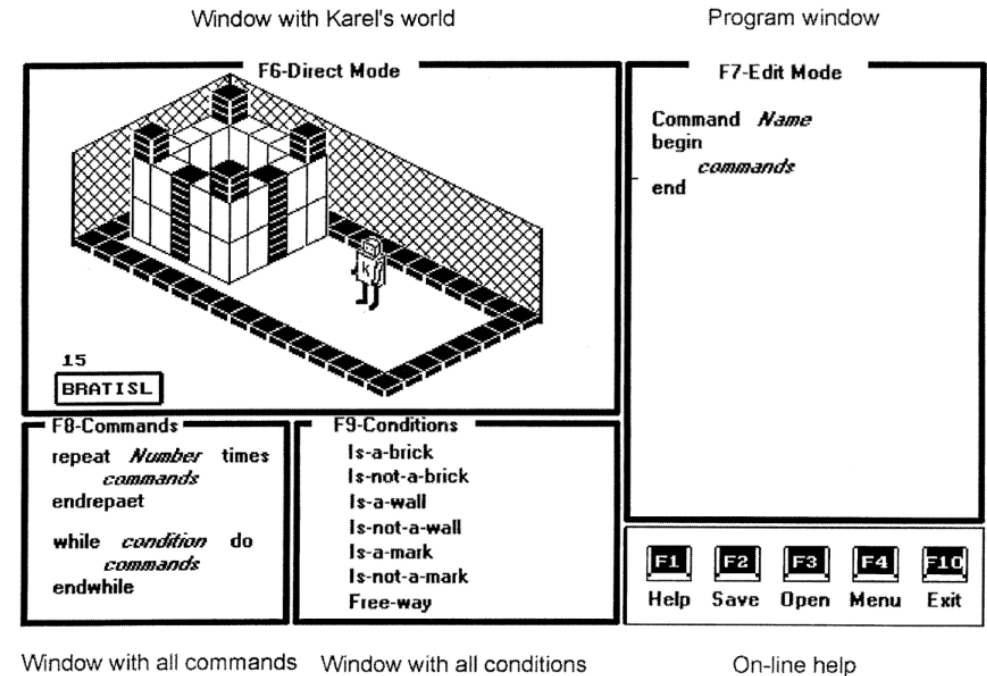


Figure 2. Karel-3D

# THREE PROGRAMMING ENVIRONMENTS FRIENDLY TO BLIND STUDENTS IN LOWER SECONDARY EDUCATION

*„Generally, we can say that the Coshi environment has been proven as appropriate for blind students. It has a good sound response and the creation of a program using commands from a menu or using the keyboard shortcuts was suitable for students. The commands were understandable, but it is important to implement a new command – conditional statement without the ELSE branch.”*

Autorka: Ľudmila Jašková

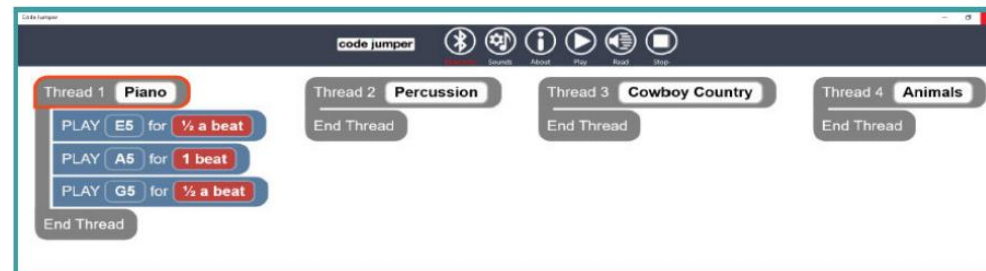


Figure 2. A software running on the computer with a text (block-based) version of program<sup>4</sup>.

While the Code Jumper was created, our team developed the Alan environment in collaboration with our student of applied informatics [3].

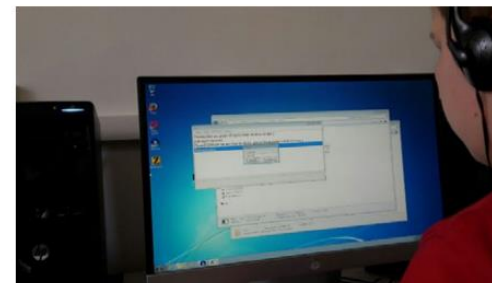


Figure 3. Blind student programming in Alan.

The Alan environment allows a user to program audio stories or simple melodies. The environment uses a simple programming language with two basic commands: **Play** and **Say**.

# Konkrétne riešenia cieľov

- Interpreter -> kompilátor
- Premenné
- Vetvenia
- Podprogramy
- Plná podpora NVDA
- Spolupráca na výskume s triedami zo Spojenej školy internátnej pre žiakov so zrakovým postihnutím



Ďakujem za pozornosť

Rebeka Sojka