Adding New Activities to GCompris Project Under KDE for Google Summer of Code 2022

Mentors: Deepak Kumar, Harsh Kumar, Emmanuel Charruau

Personal Details:

- Name: Aastha Chauhan
- University: Institute of Engineering and Technology Lucknow
- Country: India (GMT +5:30)
- Email: aastha.chauhan01@gmail.com
- Program: Bachelor of Technology
- Major: Electronics & Communications Engineering
- Primary Language: English

About Me

I’m an undergrad student majoring in Electronics and Communications Engineering from Institute of Engineering and Technology Lucknow. I have been one of the top 50 recipients of the prestigious Google Women Engineers Program Scholarship by TalentSprint for Asia Pacific Region in
2021. I am passionate to leverage technology as a tool to empower common people. I am interested in problem solving and contribution to OSS. My contributions to GCompris can be found here.

**Why Gcompris?**

I got to learn about Gcompris through Season Of Code KDE. What sparked my interest in GCompris is the fact that it is a versatile software which trains children through a variety of activities in a circumferential manner. As a developer, GCompris gives us a fantastic opportunity to contribute to a product for a better future. We never know what half-baked ideas our simple contribution can generate.

I’ve been actively interacting with the community and discussing issues with fellow contributors ever since the first day and each time I’m of some help (or try to) has started giving me confidence and a sense of satisfaction which propels me to pursue what I started.

I have been incredibly fortunate to have found such an optimistic community. With the useful guidance of my mentors, I wish to make worthy contributions with the best of my abilities.

**Abstract**

The main objective of this project is to add new activities to GCompris. For GSOC 2022, I plan to add the following activities:

_Aastha Chauhan_
1. Programmable Tux
2. Comparator Activity
3. Guess 24

These activities cater to different domains such as number logic, and computational thinking. Apart from this, I will try to write blogs not only about my activities but also on how one can start contributing to GCompris which will help the future contributors. If there are any pending wiki pages or other old documents which need to be updated, with the guidance of my mentors, I wish to update them as well.

**Programmable Tux**

This activity has been derived from the famous [MS Logo](https://en.wikipedia.org/wiki/Logo_(programming_language)) programming language developed by George Mills at the Massachusetts Institute of Technology. The sole motivation of the activity is to inculcate computational thinking in children while giving them the space to be creative.

**Screen Elements**

<table>
<thead>
<tr>
<th>S.No</th>
<th>NAME</th>
<th>TYPE</th>
<th>ATTRIBUTES</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drawing Area</td>
<td>Rectangle</td>
<td>Background with Border</td>
<td>To display the moves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Icon/Widget</td>
<td>Analogous to</td>
<td>To simulate the movements</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>-------------</td>
<td>--------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>2</td>
<td>Tux</td>
<td>Icon/Widget</td>
<td>MS Logo Turtle</td>
<td>To simulate the movements</td>
</tr>
<tr>
<td>3</td>
<td>Command Button Panel</td>
<td>Buttons</td>
<td>Push-Button</td>
<td>To perform the action moves when clicked</td>
</tr>
<tr>
<td>4</td>
<td>Input Area</td>
<td>Rectangle</td>
<td>Background</td>
<td>To get the user input value of respective commands</td>
</tr>
<tr>
<td>5</td>
<td>Commander Window</td>
<td>Rectangle</td>
<td>Border and GCText</td>
<td>To brief about most recent commands</td>
</tr>
<tr>
<td>6</td>
<td>History Button</td>
<td>Button</td>
<td>Push-Button</td>
<td>To get a comprehensive command history</td>
</tr>
<tr>
<td>7</td>
<td>Undo Button</td>
<td>Button</td>
<td>Push-Button</td>
<td>To undo the last command</td>
</tr>
<tr>
<td>8</td>
<td>Clean Button</td>
<td>Button</td>
<td>Push-Button</td>
<td>To clear the screen</td>
</tr>
<tr>
<td>9</td>
<td>Execute</td>
<td>Button</td>
<td>Push-Button</td>
<td>Pressing the execute button runs the</td>
</tr>
</tbody>
</table>
List of Commands

The activity would incorporate the basic commands of MS Logo as listed down below.

- Forward: To move the tux forward
- Backward: To move the tux back
- Left: To move the tux to left
- Right: To move the tux to right
- Pen Up: To set the tux to move without drawing
- Pen Down: To reset the tux to a drawing pen when ordered to move
- Hide Tux: To hide the tux and aid viewing a clear drawing on the screen
- Show Tux: To show the tux after it is hidden from the screen

The Workflow

The working of the activity would be simple. The user presses the command button, inputs the value of the command and hits the execute button. The pattern is visible in the drawing area. Another screen would appear when the user clicks on the history button on the left top corner.

For free-to-draw mode, the user can give any commands and realize the pattern on the screen.
Supported Modes

1. Free-to-draw Mode

In this mode, the user is free to explore and be creative by entering commands of his/her choice (illustrated in fig 1.2).
2. Instructed Mode

In this mode, an instruction would be given in the beginning of each level and the difficulty would vary according to the levels (illustrated in fig 1.3).

Fig. 1.2: Free-to-draw Mode
Fig. 1.3: Instructed Mode

Implementation Plan

1. In this mode the drawing area, the button panel and the commander window are rectangles.
2. Each rectangle has its own width, length, height, border and text entity.
3. For implementing the drawing area, I am planning to use Canvas Element in qml.
4. To enable tracking of the mouse events on the canvas there is a MouseArea covering the canvas element which would incorporate the pressed and position change handlers. Let us say, we have properties
'lastPosX' and 'lastPosY', so, a mouse press on canvas would store the initial mouse position into lastPosX and lastPosY.

5. Every change on the mouse position triggers a drawing request on the canvas, which will result in calling a handler containing the instructions of the command pressed. For example, in a handler called 'Forward', we initiate a new path and go to (lastPosX, lastPosY). Then we set the new position to (lastPosX, lastPostY+'value entered by user") and draw a line to the new position. The mouse position is stored as the new last position.

6. Similarly, if we want to introduce commands for drawing circles, we can use bezierCurve under canvas element in qml.

7. The user can input the values in the top right box which would display the numPad in case of enabling the “virtual keyboard” in the options or automatically for touch based devices.

8. Depending upon the value and the command entered by the user, the tux would move.

9. When clicked on the history button, a panel, just like that of tutorial panels in other activities, would pop up which would give a comprehensive list of recent commands.

**State Variables**

1. **Declaration:** int LastPosX, int LastPosY, boolean PenStatus, int TuxAngle, int drawingArea[100][100]

Aastha Chauhan
Adding New Activities to GCompris - GSoC 2022, KDE

2. **Initialization**: LastPosX = 50 & LastPosY = 50 ; TuxAngle=8 ; penStatus = true; drawingArea[0] [0]

3. **Constraints**: 0 <= LastPosX && LastPosY <= 100 ; 1<= TuxAngle <= 8;

4. **Command Constraints**  
   - **Forward Command**: if (LastPosX<100 && LastPos>=0);  
   - **Backward Command**: if(LastPosY<100 && LastPost>=0);  
   - **Left/Right Command**%45==0;  
   - **PenUp Command**: if(PenStatus == false)  
   - **PenDown Command**: if (PenStatus == true)

5. **Constraints for shape instructions**: The size of each side of the shapes will be fixed, for example: 'Draw a square of edge length 5'. This will help us to know the starting and end position coordinates.

**Verification Of Answer**

Whenever tux moves over a coordinate, its state will change to 1.

1. At the end, the state's matrix drawn by the user will be compared with the preset matrix of the shape.
2. If there is any mismatch between the two matrices, the answer submitted will be incorrect, otherwise, correct.
3. This will also ensure that there is nothing more or less drawn on the screen other than the instructed shape as the state's of other coordinates will still be zero.

**Difficulty Levels**

Note: FD: Forward ; BK: Backward ; LT: Left ; Rt: Right

Aastha Chauhan
The sequence of commands written for each shape is not the only possible way to achieve the results. This is just one sequence of commands to draw the shape.

**Level 1 : Begin with Basics**

This level would contain basic figures with the use of RT 90

**1. Square**

```
FD 100 RT 90 FD 100 RT 90 FD 100 RT 90 FD 100 RT 90
```

![Square](image)

**2. Rectangle**

```
FD 300 RT 90 FD 150 RT 90 FD 300 RT 90 FD 150 RT 90
```

![Rectangle](image)

**3. Rectangular 'U'**

```
BK 10 RT 90 FD 10 RT -90 FD 10
```

![Rectangular 'U'](image)
Level 2: Learn the Angles

This level would have different angles with use of RT and lt command

1. 'M'

   FD 10 RT 120 FD 5 RT -60 FD 5 RT 130 FD 10

2. 'N'

   FD 50 RT 135 FD 70 lt 135 FD 50
3. 'V'

RT 135 FD 70 lt 90 FD 70

Level 3: Dashed Lines

This level would make use of penUp and penDown commands

1. 'V'

RT 135 FD 2 PU FD 2 pd FD 2 PU FD 2 pd FD 2 LT 90 FD 2 PU FD 2 pd FD 2 PU FD 2 pd FD 2

2. Square

FD 30 PU FD 30 pd FD 30 RT 90 FD 30 PU FD 30 pd FD 30 RT 90 FD 30 PU FD 30 pd FD 30 RT 90 FD 30 PU FD 30 pd FD 30 RT 90
Level 4: Get Creative!

After learning to draw lines, angles and PU/PD commands, this level would include abstract figures

1. Star

   FD 10 RT 144 FD 10 RT 144 FD 10 RT 144 FD 10 RT 144 FD 10 RT 144

![Star Image]

Guess 24

This activity has been inspired from the Traditional Guess 24 game. As an extension of puzzle 24, it was organized and operated within a series of inter school tournaments. In this, 4 numbers are given randomly, and the user is expected to make the number 24 using the operations given and the 4 numbers using each number just once.
### Screen Elements

<table>
<thead>
<tr>
<th>S.No</th>
<th>NAME</th>
<th>TYPE</th>
<th>ATTRIBUTES</th>
<th>USE</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Numbers Boxes</td>
<td>Rectangle</td>
<td>Background with radius (Circle)</td>
<td>To display the numbers</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Operator Boxes</td>
<td>Buttons</td>
<td>Push-Button</td>
<td>To perform the operation when clicked, includes addition, subtraction, multiplication and division operation.</td>
<td>4 (Maxi mum)</td>
</tr>
<tr>
<td>3</td>
<td>Input Boxes</td>
<td>Rectangle</td>
<td>Text-field</td>
<td>To take the user input of numbers</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Go Button</td>
<td>Button</td>
<td>Push-Button</td>
<td>Pressing the execute button</td>
<td>1</td>
</tr>
</tbody>
</table>
analyzes the user input, only visible at the end when all numbers and operators have been used.

The WorkFlow

1. Four numbers are generated randomly and different combinations of the four operators (addition, subtraction, multiplication, division) will be provided to the user (illustrated in fig 2.1). The operators are fixed on the activity screen. Here, the number of operators will increase gradually, for example, firstly we will give a single addition operator, then we can have subtraction and addition operator and so on.
2. The user is expected to make 24 using the same set of four numbers and all operators to win the game.
3. The difficulty will gradually increase by providing a different set of four numbers with which achieving the result as 24 will be more complex. For example, making 24 with just addition of 8,9,3,4 is easier than using both addition and subtraction to make 24 using 8,1,9,8.
4. The idea behind giving random numbers is to ensure that no question is repeated.
Similarly, making 24 with a set of 5,3,4,5 and all operators is easier than with 6,9,6,2.

![Fig 2.1 HOME SCREEN](image)

5. One thing to note here is that the Go Button will only be visible when the user has used all the four numbers (illustrated in fig 2.2).

![Fig 2.2](image)
6. The result of the first row and the second row can be used again in the third row. (illustrated in fig 2.3 & 2.4).

![Fig 2.3](image1)
![Fig 2.4](image2)

**Implementation Plan**

The implementation plan of this activity will be similar to the activity called "Guess Count". Both the activities will share the same code.

1. The screen can be divided in three parts: The number sections, the operator section and the input section.
2. All the numbers, operators and input sections will be rectangles.
3. The Textview of the number and operator section will be populated using `ListModel`. For initializing the view model, `initLevel` function of `count.js` file can be used.
4. The selected numbers and operator will be visible in the input area. The result of the operation will be visible in the RightMost Box. To calculate the result, the function `calculate` from `count.js` can be used.

5. The results will be carried over to the LeftMost Box of the next row using the `ChildrenChange` function from `count.js`, and the steps 4 and 5 will be repeated until all the numbers are used.

6. The final result should be 24, and to check this manipulation is required in the `dataset.js` file of the Guess Count activity.

7. Finally, the `checkAnswer` function will be used to evaluate the answers.

**The Comparator Activity**

This activity aims to train children to understand the use of "<" (less than), ">")" (greater than) and "=" (equal to) symbols. I started working on this activity in the mid of March 2022. I wish to complete the remaining features in GSOC 2022.

I plan to implement the following features in the Comparator Activity:

1. Implement one-question-per-window mode
2. Implement Multiple Datasets
3. Improve the way questions are selected to make the activity more intuitive.
Implementation Plan

1. Multiple datasets can be implemented using `DialogChooseLevel` function from ordering.qml in ordering numbers activity, then by adding the datasets limits in resource folder, data.qml file.

2. For more improvements, I will highlight the questions selected by using a rectangle, and make the question to be answered bigger than the ones in the list.

3. The okay button will only be visible when all the questions have been answered.

4. The next question will appear automatically, without having to push the down button.

5. When the size of the screen allows it, a sum up of all the questions with answers will be presented on the left top part of the screen.
6. When there are more questions in the list than place to display them, we can have a scroll bar at the side of the questions in the list, so the user knows that more questions are there.

7. A flow mechanism will be used to allow users to scroll through the questions.

**Timeline:**

**Pre GSoC**

- Work on existing issues
- Brainstorm features for proposed activities

**Community Bonding (May 20 to June 12)**

- Continue working on existing issues
- Plan milestones and discuss possible challenges
- Gather additional requirements such as assets and designs for all activities

**Coding Period (June 13 to September 10)**

I will publish blogs biweekly about my progress. I am expecting the tasks to be completed by

_Aastha Chauhan_
Week 1 - 2: 13 June - 27 June (ComparatorActivity)

- Implement additional features
- Test the activity for different modes and devices
- Take more reviews for my activity from the mentors.
- Make required changes on the basis of reviews.
- Adding at least 1-2 blogs for this activity
- Android compilation and testing.

Week 3 - 4 : 30 July - 11 July (Guess 24)

- Finalize the working algorithm
- Make the initial layout
- Implement the activity
- Test the activity for different modes and devices
- Take more reviews for my activity from the mentors.
- Make required changes on the basis of reviews.
- Add at least 2-3 blogs for this activity
- Android compilation and testing.
Evaluations (25 July to 29 July)

I plan to complete Guess 24 and Comparator activity before phase 1 Evaluations end.

Week 5 - 9: 11 July - 8 August (Programming Tux)

- Finalize the working algorithm
- Make the initial layout
- Implement the activity
- Test the activity for different modes and devices
- Take more reviews for my activity from the mentors.
- Make required changes on the basis of reviews.
- Adding at least 2-3 blogs for this activity
- Android compilation and testing.

Week 9 - 10: 8 August - 20 August (Finalize and Submit)

- I would utilize this time in polishing the features in the activities
- Extensive testing for various devices
- Finish working with all the activities
- Review all the work done
- Make sure the documentation is proper and complete

Aastha Chauhan
Final Evaluations (20 August to 10 September)

Post GSoC

I plan to continue contributing to GCompris, and would love to mentor students in their projects in the upcoming summers. Moreover, if for some reasons I am not able to complete all the proposed activities in the stipulated time, I will continue working on it in a timely manner to bring it to a successful close.

Other Commitments:

I will be able to put in about 17-20 hours per week. I only have some university commitments. I'll totally commit to working on my Gcompris project and will ensure that I stick to the deadlines as discussed. I have no other employment. However, if some unforeseen events involve me, I'll make sure to put in extra hours prior to them and ensure that I don’t miss any set targets. I’m largely flexible when it comes to communication times.

Other GSoC Proposals

This is the only proposal that I am submitting for Google Summer of Code 2021.

Final Thoughts

I’m really glad to have got this opportunity of working with such a huge, active and helpful community of developers from around the globe at

Aastha Chauhan
GCompris. I am fully aware of what I’m committing myself to and understand that it is not going to be a walk in the park but something that requires a lot of sincerity and dedication. However, I’m ready to give whatever it takes and assure you that I’ll make sure to bring this project to a close as expected by my mentors. I’m appreciative of the challenges that I am going to encounter in delivering what I’ve just committed and look forward to a great learning experience which makes a difference in the world.

- Aastha Chauhan