**Introduction**

This project is started with an aim of learning game design, coding, team-work and participating in [Georgia Educational Technology Fair](http://gatechfair.org/)

Why Game design? After reviewing all the topics for the technology fair we found game design most interesting and challenging.

**About Creators**

**Ananya Sai Tadepalli**

Ananya is a 3rd grader from Crabapple Crossing Elementary school. She loves to dance and play chess. She has been member of USCF and school chess club since 2012.

**Dhvanish Vagela**

Dhvanish is a 3rd grader from Crabapple Crossing Elementary school. He is a budding Basketball player and loves to play chess.

**Game Design**

Brainstormed on various games we could build such as educational games, basketball, but decided to build something on Chess because it was common interest for both of us.

**Why do we need a new Chess game?**

1. Other chess based games aren't fun and make children lose interest in the game.
2. Other chess based games make you loose more points when you get something wrong then if you give the correct answer.
3. Other chess based games don't give you options so that after you check all the options only one is correct.

**Game flow**

1. User is shown a splash screen with game logo. In future we will add cool animations on this page
2. User needs to press any key to begin
3. User starts at level 1
4. Each level can have any number of puzzles. In the initial version we have three puzzles per each level.
5. Each level is named after a chess piece.
   1. level 1: Pawn, in this level animation of a pawn jumping with a corresponding sound is shown
   2. level 2: Knight, in this level animation of a Knight jumping with a corresponding sound is shown
   3. level 3: Bishop, in this level animation of a Bishop jumping with a corresponding sound is shown
   4. level 4: Rook, in this level animation of a Rook jumping with a corresponding sound is shown
   5. level 5: Queen, in this level animation of a Queen jumping with a corresponding sound is shown

The animations jump to a higher step as user answers a question.

1. After correctly answering all questions in a level user advances to next level
2. A wrong answer will receive negative points (-10) and a correct answer will get you (+10)
3. There is no restriction on number of trials till getting the correct answer
4. To exit the game they can press 'ESC' at anytime

**Development Tools/ Resources**

1. Windows based game.
2. Use YOYO Game Maker studio free version
3. EXCEL for generating puzzle data
4. [Mr Data Converter](https://shancarter.github.io/mr-data-converter/) for converting excel data to json
5. Citation: <http://www.easybib.com/cite/view>
6. Code sharing : [github](https://github.com/chess4fun/chess4fun)
7. Documentation: [Github-wiki](https://github.com/chess4fun/chess4fun/wiki)

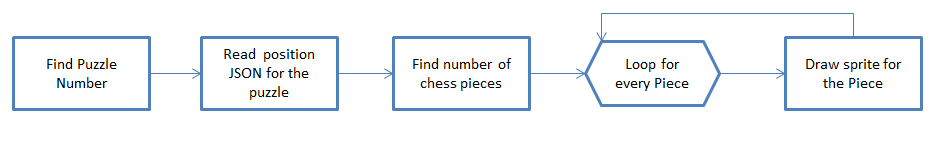
**Design Details**

**Definition of Terms**

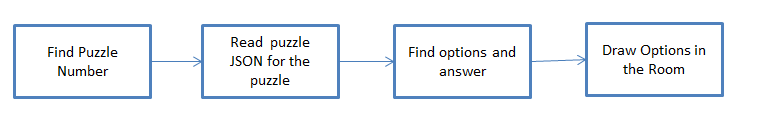
1. Room: A room is the screen space in which game is played. There can be multiple rooms. In our game we have rooms for each level and separate rooms for introduction congrats etc.
2. Object: An object is placed inside the game room. Some objects are visible like a pawn and there are other objects which are not visible. Like the input object which waits for an input and acts on it. All visible objects have a sprite attached which defines how the object looks. Objects can have events and actions configured to them. For example in our game when user clicks Escape, the exit object triggers and event and action to close the game.
3. Sprite: A sprite can have one or more images. When it has more images those help in animating the sprite.
4. Background: A background can be created using an image. Once it is defined it can be added to the room. There can be multiple backgrounds one on top of other.
5. Sound: Sound can be added from a mp3 file. Once it is defined, it can be triggered using events and actions on objects.

There are four key components to the game

**Draw the chess board and pieces**

1. Use sprites to design the black and white squares in the chess board. Finally take a snapshot and make it into a background in order to save memory.
2. Create a sprite for each of 12 chess pieces
3. Define a json format for reading the game position
4. Read game position and add the correct sprite in the position specified 

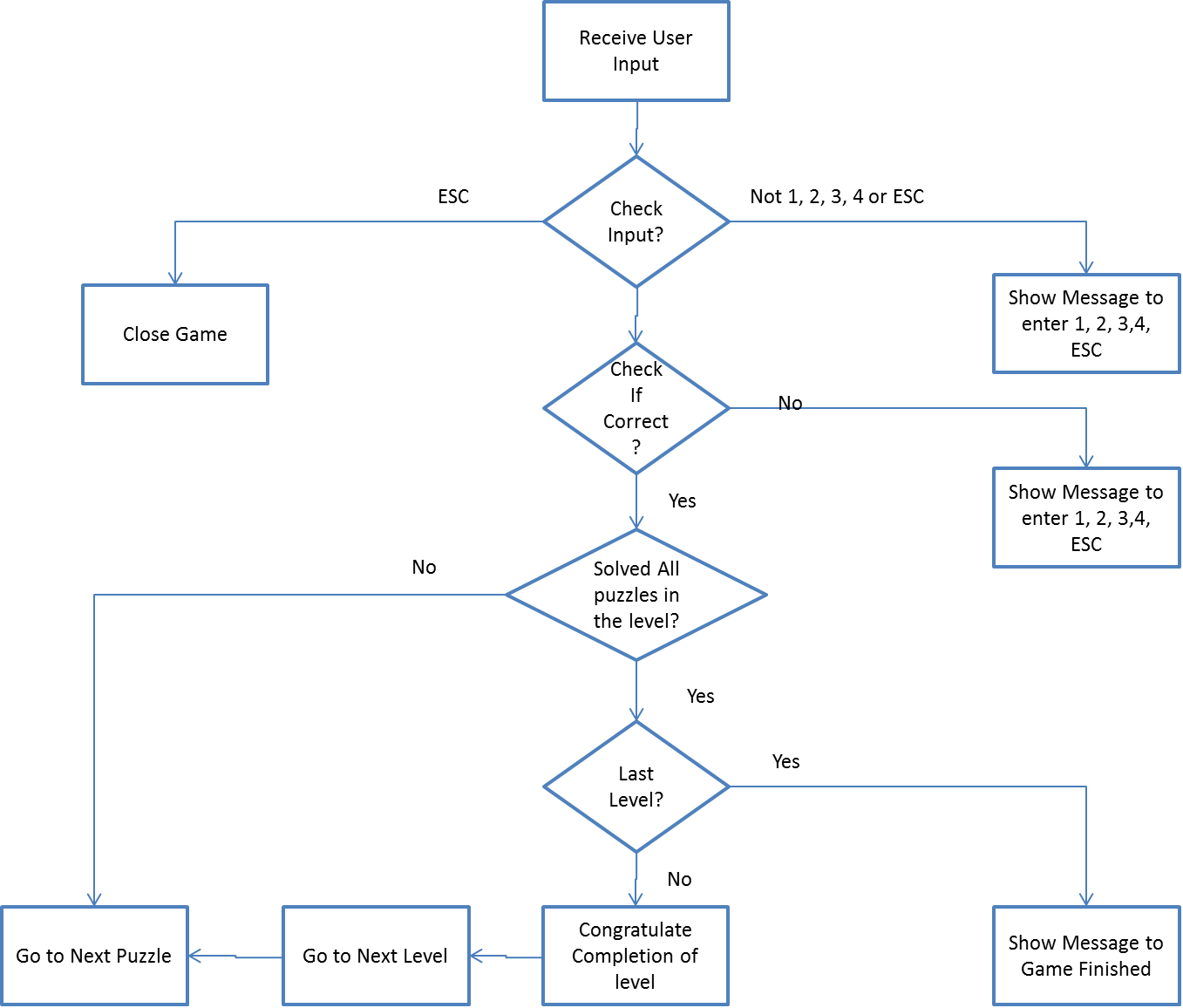
**Display puzzle options**

1. Read puzzle options and correct answer from a json file
2. Display the options in the screen
3. set global variable for correct answer 

**Initialize game**

1. Set the starting game level, puzzle, score etc
2. Go to the first room

**Receive user input and take action**

1. Read the user input.
2. Check if it is 1, 2,3, 4 or ESC.
3. If not display message showing allowed inputs.
4. If ESC is pressed close game.
5. If 1, 2, 3 or 4 is pressed, check if it matches with the correct answer
6. If number entered does not match correct answer display message to user to try again
7. If it matches correct answer, check if user has solved all puzzles in the level
8. If user solved all puzzles in the level, check if it is the last level
9. If it is the last level congratulate user on completing the game
10. If it is not the last level congratulate user on completing the level. Increment the level. Assign points
11. Restart the room at the current or new level 

**Citations/ Credits**

1. "Game Maker Basic Chess Tutorial - Part 1/4: Setting Up the Board." *YouTube*. YouTube, n.d. Web. 06 Jan. 2016. <<https://www.youtube.com/watch?v=vEwNYf3rk7U>>. Used this video to understand how to create a chess board and lay out chess pieces
2. "Chess Icons - Download 60 Free Chess Icons Here." *Chess Icons - Download 60 Free Chess Icons Here*. N.p., n.d. Web. 06 Jan. 2016. <<http://www.iconarchive.com/tag/chess>>. Used this source to download various icon for King, Queen etc
3. "GameMaker Tutorials: Using JSON Data." *JASON LEE ELLIOTT*. N.p., 21 May 2015. Web. 06 Jan. 2016. <<http://jasonleeelliott.com/using-json-data/>>. Used this site to understand how to read puzzle data and create each puzzle.
4. "Mr. Data Converter." *Mr. Data Converter*. N.p., n.d. Web. 06 Jan. 2016. <<https://shancarter.github.io/mr-data-converter/>>. Used this tool to transform puzzle data in excel to json format.
5. "SoundBible.com." *Free Sound Clips*. N.p., n.d. Web. 06 Jan. 2016. <<http://soundbible.com/>>. Background sounds for the game were downloaded from this source
6. "Free Images - Pixabay." *Free Images - Pixabay*. N.p., n.d. Web. 06 Jan. 2016. <<https://pixabay.com/>>. Used this site for free images for backgrounds
7. "Create a Text GIF." <i>Bloggif : Create Text GIF for Free</i>. N.p., n.d. Web. 06 Jan. 2016. <span>Used this site to create animated text</span>

**Experience**

1. We had to go through several tutorials to learn the tools. We build a few sample games to understand each components.
2. We got help from our parents whenever we were stuck with a problem. They also helped us understand how to choose images and sounds that are royalty free.
3. We initially tried to use Scratch to build our animations, but we could not find an easy way to pull those animations into our game. So we decided to build simple animations using sprites.
4. We learnt a lot about working as a team and saving all work to github saved us several times. We did brainstorming, voting for name of the game etc.

**User Guide**

**Installation**

1. Download the installer from link
2. You will need a Windows 7 or higher PC
3. Run the installer in order to install the game

**Play**

1. Start the game using the game icon on your desktop
2. After you see the splash screen click any key to strt the game
3. For each puzzle 4 options are shown
4. Click the number corresponding to the correct answer in order to score points and move to next puzzle
5. At any point you can exit the game by clicking the ESC key
6. Score is displayed under the puzzle
7. A jumping animation and sound show which level you are on

**Upcoming features**

1. Adding user and save game features
2. Ability to download certificate of achievement
3. Ability to change settings such as background, music, puzzles etc
4. Ability to scan a puzzle