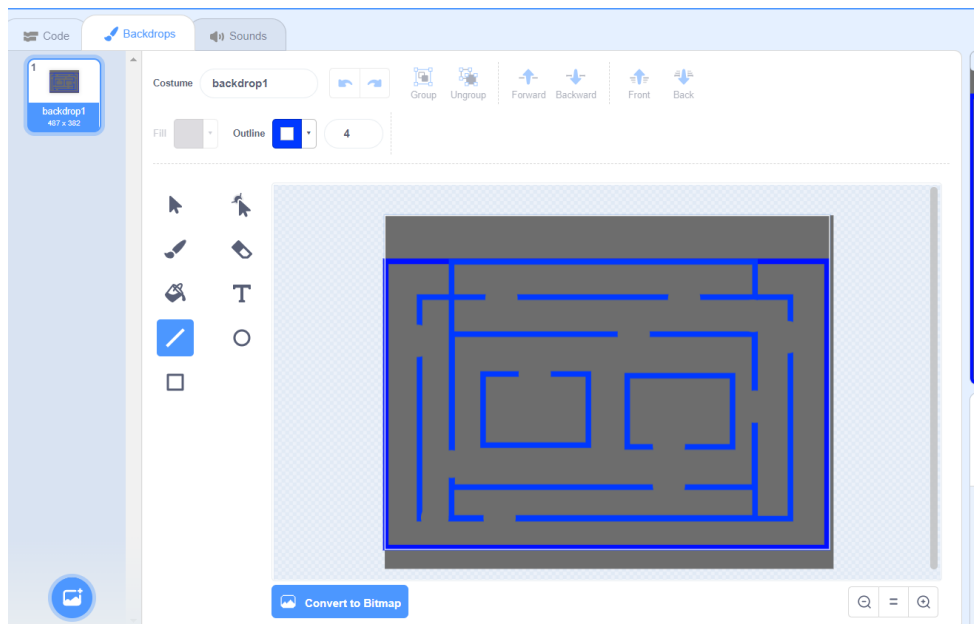


What is game?

A **video game**^[a] or **computer game** is an **electronic game** that involves interaction with a **user interface** or **input device** (such as a **joystick**, **controller**, **keyboard**, or **motion sensing** device) to generate **visual feedback** from a **display device**, most commonly shown in a **video** format on a **television set**, **computer monitor**, **flat-panel display** or **touchscreen** on **handheld devices**, or a **virtual reality headset**. Most modern video games are **audiovisual**, with **audio** complement delivered through **speakers** or **headphones**, and sometimes also with other types of sensory feedback (e.g., **haptic technology** that provides **tactile** sensations). Some video games also allow **microphone** and **webcam** inputs for **in-game chatting** and **livestreaming**.

Draw the structure of chess game.



Question No. 3.

What is the difference between 2D game and 3D game?

A 3D game is one that uses three-dimensional graphics, such as polygons, textures, or models, to create a more realistic and immersive environment. 2D games are usually simpler and more stylized, while 3D games are more complex and realistic.

Question No. 4.

Explain basic colors used in game.

that using colors can:

- shape player behavior,
- control the player's attention,
- give the player important feedback about their character and signal important cues about affordances or current state of the game world,
- convey various meanings of scenes, locations, and world elements,
- build a mood and evokes certain emotions to stimulate or to soothe the player,
- foster the automation of mental interface processing (e.g., some game HUD elements already have specific color assignments that players recognize instantly).

Section-B

(8×2=16)

Question No. 5.
PC MAN.

What are the basic principles of game design? Explain

SEVEN PRINCIPLES OF GAME DESIGN AND FIVE INNOVATION GAMES THAT WORK

The gaming industry is larger than Hollywood, by many measures. People spend tremendous amounts of time and money on games—and even pay to watch other people play games (and not just baseball...the competitive video game market is huge...and baffling!). But *gamification* is everywhere too—in education, in corporate culture, in the innovation industry. Why?

One of our favorite books, *Finite and Infinite Games* by James P. Carse, talks about how games appear in time and space and make a goal and a winner clear at the end, based on rules. When compared with most parts of our lives, that seems like a pretty good bargain – at our jobs, when does the change stop? How do we know if something “worked” or didn’t, if the game never ends? Life is, by some measures, an infinite game, where the rules constantly change and winners turn into losers, and vice versa. It’s exhausting!



[Game Design principles to fuel innovation: Play, Reward, Fidelity, Constraints and more](#)

Setting up games, with clear goals and constraints help focus our energies and efforts and can improve and clarify outcomes and motivate us to move forward to the next clearly defined challenge and reward cycle. We use games all the time in our [facilitation work](#), drawing on books

like *Gamestorming* and *The Systems Thinking Playbook*, to get teams thinking differently and to increase creative output.

SEVEN DESIGN GAME PRINCIPLES

Most of the principles below, with the exception of the first principle of Fidelity, are drawn from the Boxes and Arrows article, [“Using Design Games.”](#) Check it out for more links and discussion.

1. *Fidelity:*

Will the team be solving a challenge that is NOT like the current problem? That is, will the game be just for clarity and learning? Or will the team be solving a challenge like current challenges, but stripped down and simplified? There’s no hard rule on this...your mileage may vary. Our feeling is that an abstract or general challenge is best to loosen a team’s thinking up first, then to approach the challenge sideways, not directly. The “always, never” game is a great way to come at the principles to guide a solution to a challenge in a safe and fun way.

2. *Objectives:*

There needs to be some kind of goal or outcome that people can work towards. The more concrete and defined these are, the easier it is for people to participate. However, fuzzy objectives can be more rewarding, since they model real situations better. Consider more ambiguous objectives for teams that are already gelled and accept the ideas for design games.

3. *Constraints:*

There needs to be some limits on what players can or can’t do when achieving those objectives. Constraints should be relevant, related to each other, and present a coherent whole.

4. *Success Criteria:*

There needs to be some way of knowing when the objectives are met. Clear success criteria help establish expectations and buy-in for game participation. Some games are more unstructured, with less well defined criteria. Classic role-playing games like Dungeons & Dragons don’t have a clear overall objective. That ambiguity can make such games better able to model some scenarios, but harder to sell inside an organization because they don’t have a set ending.

5. *Reward:*

Incentives that reward success can be intrinsic outcomes of the game (good results, recognition), embedded in the game itself (getting more Monopoly money), or external recognition or prizes (the winner gets dinner at a nice restaurant). Balancing rewards between players can be a challenge, and needs to be considered when adopting games.

6. Play:

The most important reward needs to be a sense of fun, encouraging interaction and intrinsic value for the game. That sense of play can be elusive—playtesting a design game in your own team is important to get a sense of what is fun, and what isn't before you roll it out with a larger group. Play operates in the area of flow—balance the challenges of the game with the abilities of the players.

7. Competition (sometimes):

Sometimes, but not always, design games can involve individuals or teams competing to achieve those game objectives. While competition can be an easy game mechanic to introduce, it can also create the wrong dynamic depending on organizational culture and individual participants. Does competition make things fun, or turn people into raving lunatics bent on winning at all costs? If it's the latter, you might look for more cooperative alternatives, including setting competition against previous performances, like beating your old record for ideas generated, instead of against other teams or individuals.

FIVE INNOVATION GAMES THAT ALWAYS WORK

1. 100 uses for...

We often use “100 uses for...” as a way to get creative juices going and sometimes to prime a team to tackle a challenge. When I was in design school, one of my professors had a “100 ways to serve pizza” assignment. Everyone was given a pack of 100 paper plates and had to draw a serious or silly way of serving pizza. That was a week-long assignment, and people really saw how hard it was to come up with a 100 options... you had to think outside the box! When working with groups, we'll call the game “100 uses for...” but give them only 5 minutes! The prompt can be pizza, a log, or something more relevant. When doing a workshop with our friends at KeyMe, we used the prompt “100 uses for a key” to get people thinking about their challenge.

2. Mix and Match (Fruit Party)

Fruit party is a fun game that can be used to teach a variety of innovation themes, from the importance of generation and combinatory innovation to the idea of there being “no bad ideas.”

Team members each choose a fruit, with no duplicates. I've played this game with 15-50 people, and it works well in both cases. I then have people arrange themselves by various criterion – color, size, cost, then by flavor. I'll then select three or four people, representing different fruits and ask what fruit mix they are. Sometimes the combination sounds good, sometimes it doesn't! Each time I ask what useful purpose the combination can make.

3. Generate and Share

Groups always want to talk first...I just don't let them! The simplicity of this game is that you provide the generation template – full size paper, post-its, how many elements each concept needs to have, etc.

“100 uses for...” is a generate and share game. People don't call out ideas, they write them down! And we give rules, like one post-it per idea and ideas with words AND pictures are better. Deciding what you want the team to generate and then making a simple template for it is an easy way to get started.

4. Always/Never

I use this with teams all the time as way to clarify their thinking on an issue. A team was having a discussion about the new employee onboarding process and was getting bogged down trying to generate features, workflows and concepts. Giving them 5 minutes to generate and share what the onboarding process should “always be like” and then generating what it should “never be like” made a clear visual word and concept map that sparked features, workflows and concepts more easily.

5. Franken Ideation

When teams have generated ideas or concepts, mixing and matching is a great way to get them to go further. Fruit Party can teach the principles of this, but doing a round of Franken Ideation can help them dig deeper. Have each team member grab 2-3 post-its representing ideas or concepts from the wall, and to not think too much about which ones. What will combining these ideas give us?

Once, during a color-generating teaching exercise, a participant took “Green Sweater” and “Campfire Orange” to make “Singed Wool”. This process based mixing is creatively different from literally mixing these two colors...and far more creative! Doing this with more high level ideas is hard, but will get your team to unique ideas. Remember, even bad ideas can be good ideas if we look at them right!

Question No. 6. Explain different objects used in game?

What are Game Objects?

Game objects are the central concept in Wwise because every Event triggered in the sound engine is associated with a game object. A game object generally represents a particular object or element in your game that can emit a sound, including characters, weapons, ambient objects, such as torches, and so on. In some cases, however, you may want to assign game objects to different parts of an in-game element. For example, you can assign a different game object to

different parts of a giant character so that the footstep sounds and the character's voice emanate from different locations within the 3D sound space.

info Note

If you are familiar with the Unreal game engine, game objects in Wwise are similar to Actors in Unreal.

For every game object, Wwise stores a variety of information that it will use to determine how each sound will be played back in game. Any of the following types of information may be associated with the game object:

- Property offset values of an audio structure associated with the game object, including volume and pitch.
- 3D position and orientation.
- Game Sync information, including States, Switches, and RTPCs.
- Environmental effects.
- Obstruction and Occlusion.

Question No. 7. Think upon the idea of PC MAN and give a idea of a new game.

Design a Chess Game

Problem Statement: The problem is to design a [Chess Game](#) using Object Oriented Principles. **Asked In:** Adobe, Amazon, Microsoft, etc. **Solution:** These type of questions are asked in interviews to Judge the Object-Oriented Design skill of a candidate. So, first of all we should think about the classes. The main classes will be:

1. **Spot:** A spot represents one block of the 8×8 grid and an optional piece.
2. **Piece:** The basic building block of the system, every piece will be placed on a spot. Piece class is an abstract class. The extended classes (Pawn, King, Queen, Rook, Knight, Bishop) implements the abstracted operations.
3. **Board:** Board is an 8×8 set of boxes containing all active chess pieces.
4. **Player:** Player class represents one of the participants playing the game.
5. **Move:** Represents a game move, containing the starting and ending spot. The Move class will also keep track of the player who made the move.

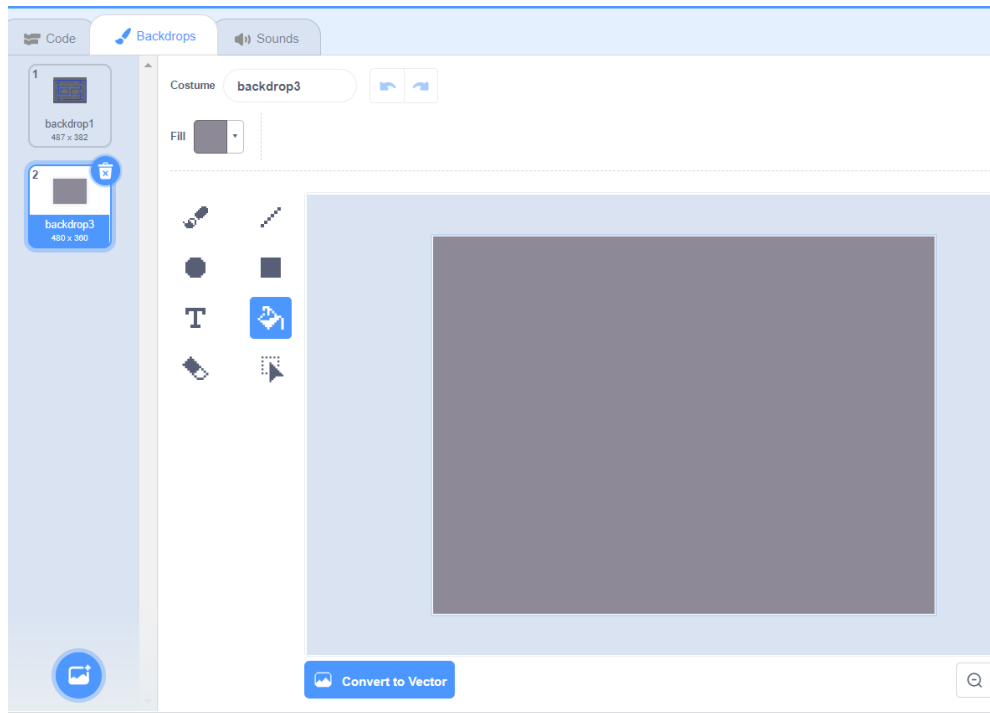
6. **Game:** This class controls the flow of a game. It keeps track of all the game moves, which player has the current turn, and the final result of the game.

How to create Pac-Man game using Scratch

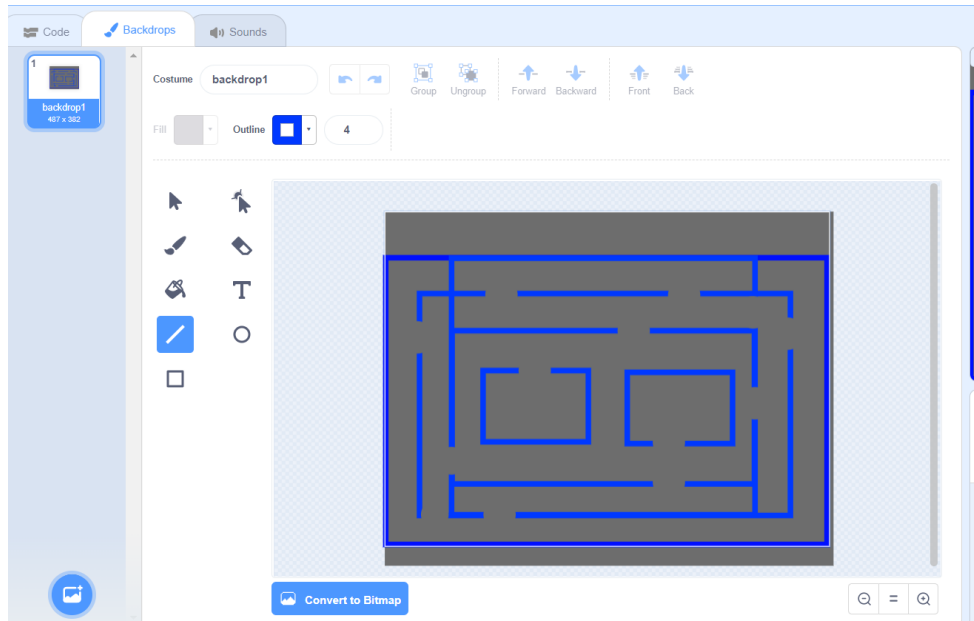
Namco developed Pac-Man in 1980 for the arcades. It is a maze game where the player can control the main character, Pac-Man, a yellow ball. It eats all the dots, also known as pills, placed inside the maze and is enclosed from all sides. In this blog , we will learn how we can create the Pacman game on the Scratch platform. We will divide this into three parts – required background and sprites addition, adding necessary variables, and coding for each.

Adding background and sprites

Scratch programming platform for kids provides functionality to add different backgrounds and sprites that are nothing but characters in the project. For the Pac-Man game, we can go for drawing our maze for the background or upload any downloaded picture for the background. Choose the paint option in the upload background popup to draw a background. Using the fill bucket option, we can easily color our background to any color. Then you can choose any fill color for the background and choose the fill option shown in the image. Click on the Convert to Bitmap option and then click on the blank space of the screen to apply color.



Now, choose the line option and color of the line in the outline option and start drawing a maze-like structure through which Pac-Man has to travel, as shown below.

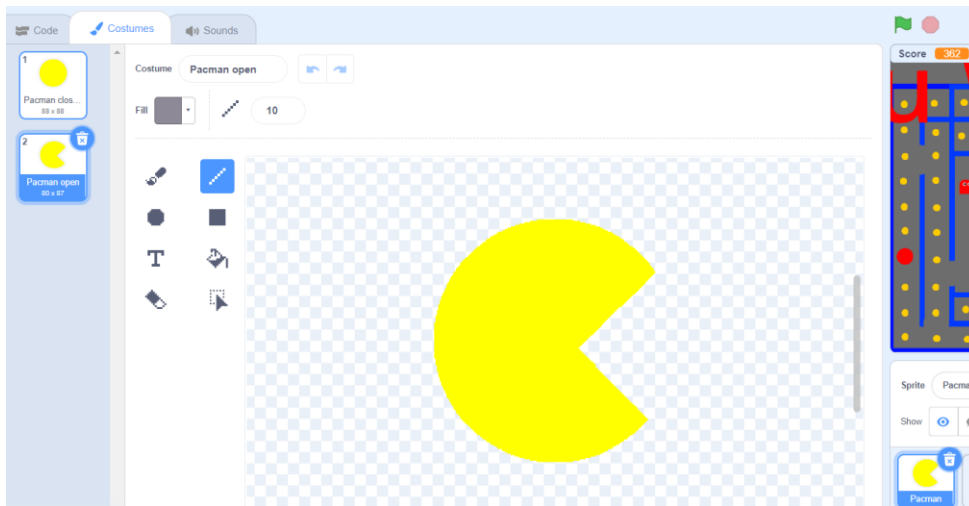


Now, we will go for adding all our sprites. First, you can download the images from google search and then remove the background of each one using the <https://www.remove.bg/upload> site. After that, you can upload all the sprites to choose

the sprite option of Scratch. You can also alternatively go for drawing your sprites using simple shape tools like circles, rectangles, lines, and erasers while painting your sprite.

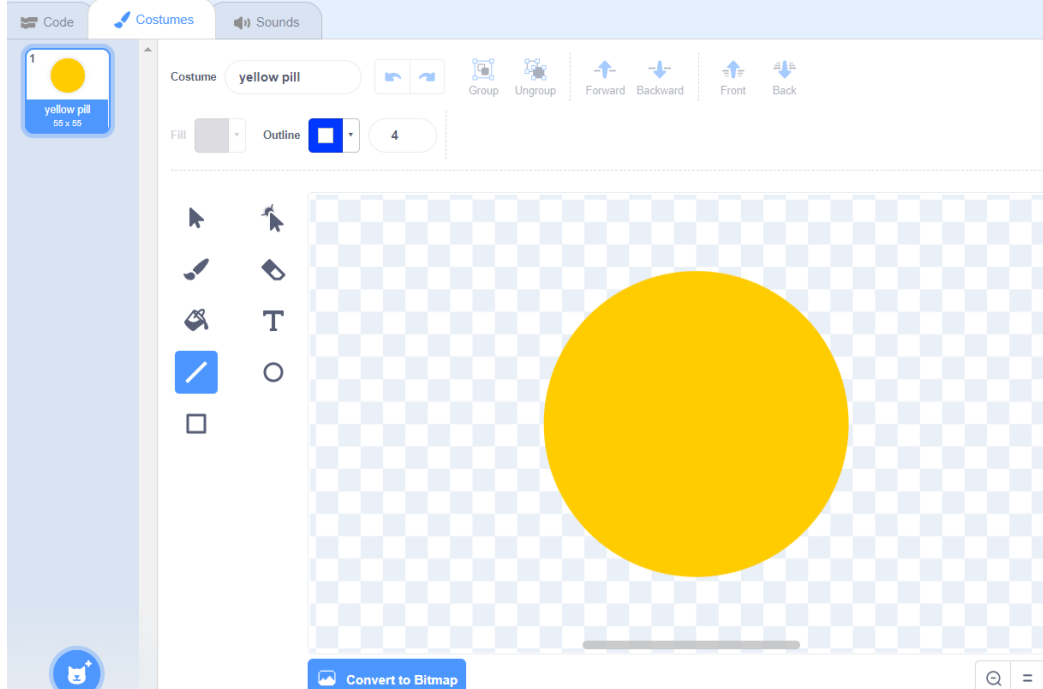
Below is the list of all the sprites and images required for the Pacman game

- Pacman sprite with two costumes of Pacman open and Pacman closed mouth to create the animation of eating **Pac-Man**

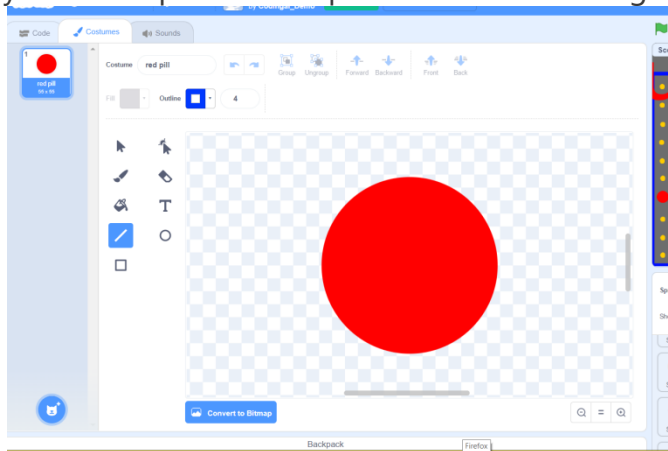


- **Yellow pill** – Pac- Man will eat this yellow pill. Every time Pac-Man eats one yellow pill, you will get 1 point and change the score by one. Later on, after completing the code of yellow pill, we will have to duplicate this sprite 109 times as we need a total of 110 yellow pills. Also, we will have to place each one of

them correctly on the stage.

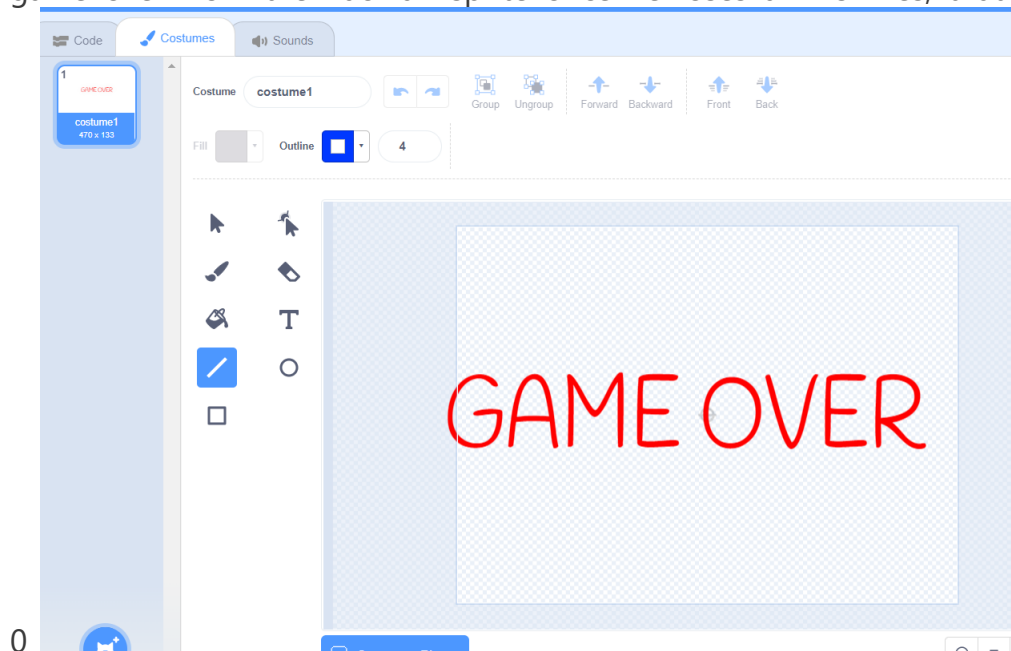


- **Red Pill** – This is the Super pill. After eating it, Pac-Man gets a superpower. This superpower will make all the aliens blue in color for 20 seconds, and Pacman can eat these aliens when they are blue-colored. We will have 3-4 red pills and can duplicate the same sprite after completing its code. After completing the code, you can duplicate the sprite instead of writing the code again.

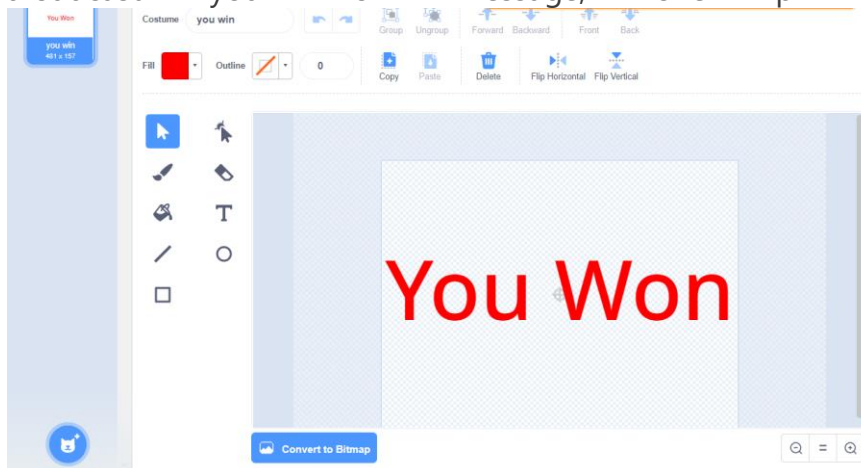


- **Game Over** – We will need a sprite which we will show once Pacman loses all his three lives and gets caught up by an alien thrice. For this, we will add one sprite of the game over, which we will show once we get the broadcast message of the

game over from the Pacman sprite once he loses all his lives, that is, lives =

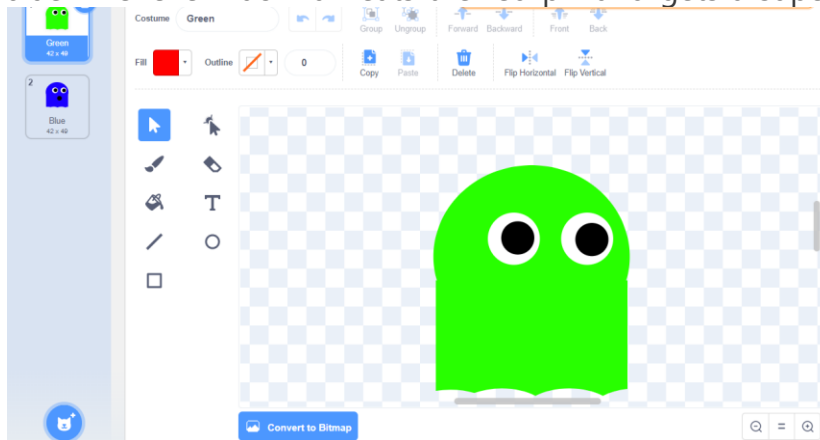


- **You Win** – You will win the game when Pac-Man eats all the 110 yellow pills. To display the same, we have added one sprite of You Won, shown when we broadcast you won message, one pill left is zero.

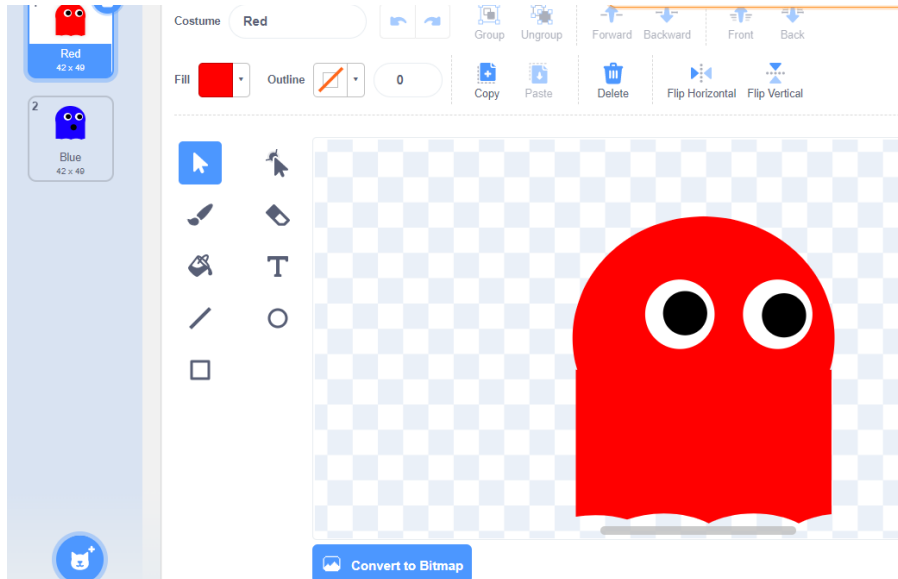


- **Alien 1** – There are four aliens patrolling the maze. If Pac -Man touches any of the aliens, he loses one of his lives. Each alien sprite will have two costumes—one of his original color and another blue color. The alien changes his costume to

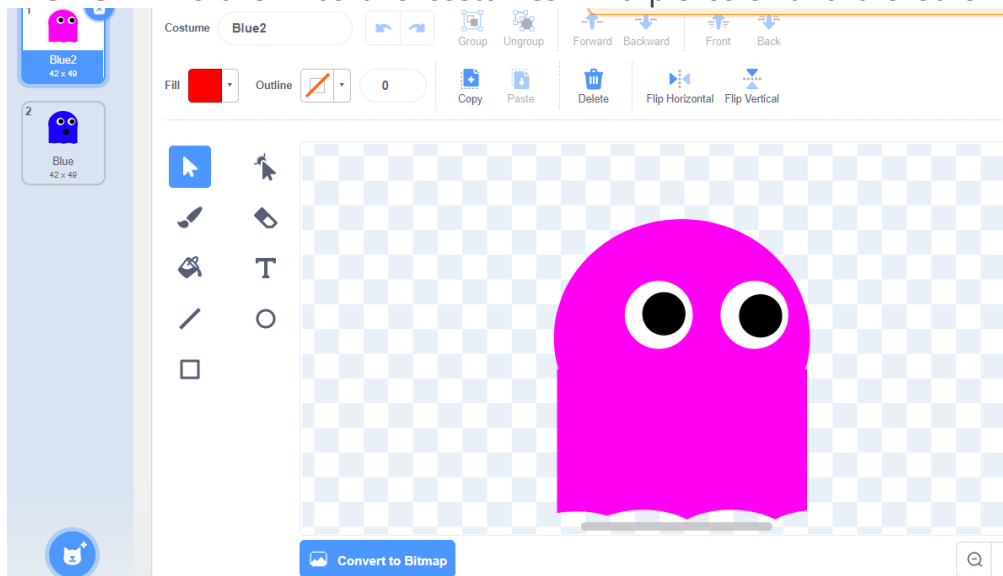
blue whenever Pac-Man eats the red pill and gets a superpower for 20 seconds.



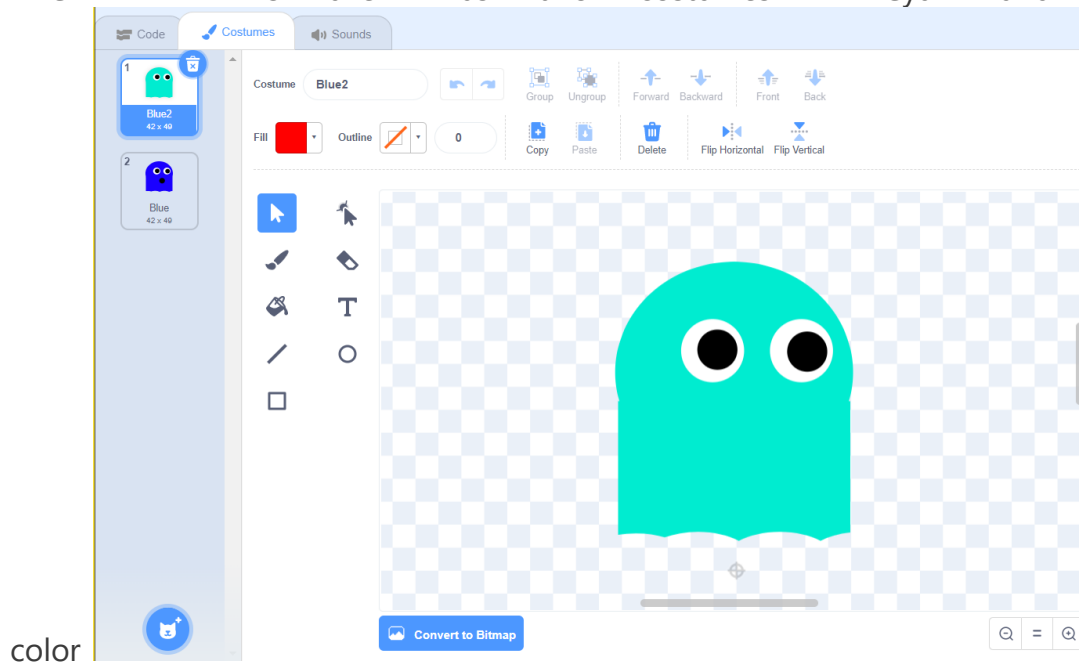
- **Alien 2** – Even this alien has two costumes – one red and the other blue color.



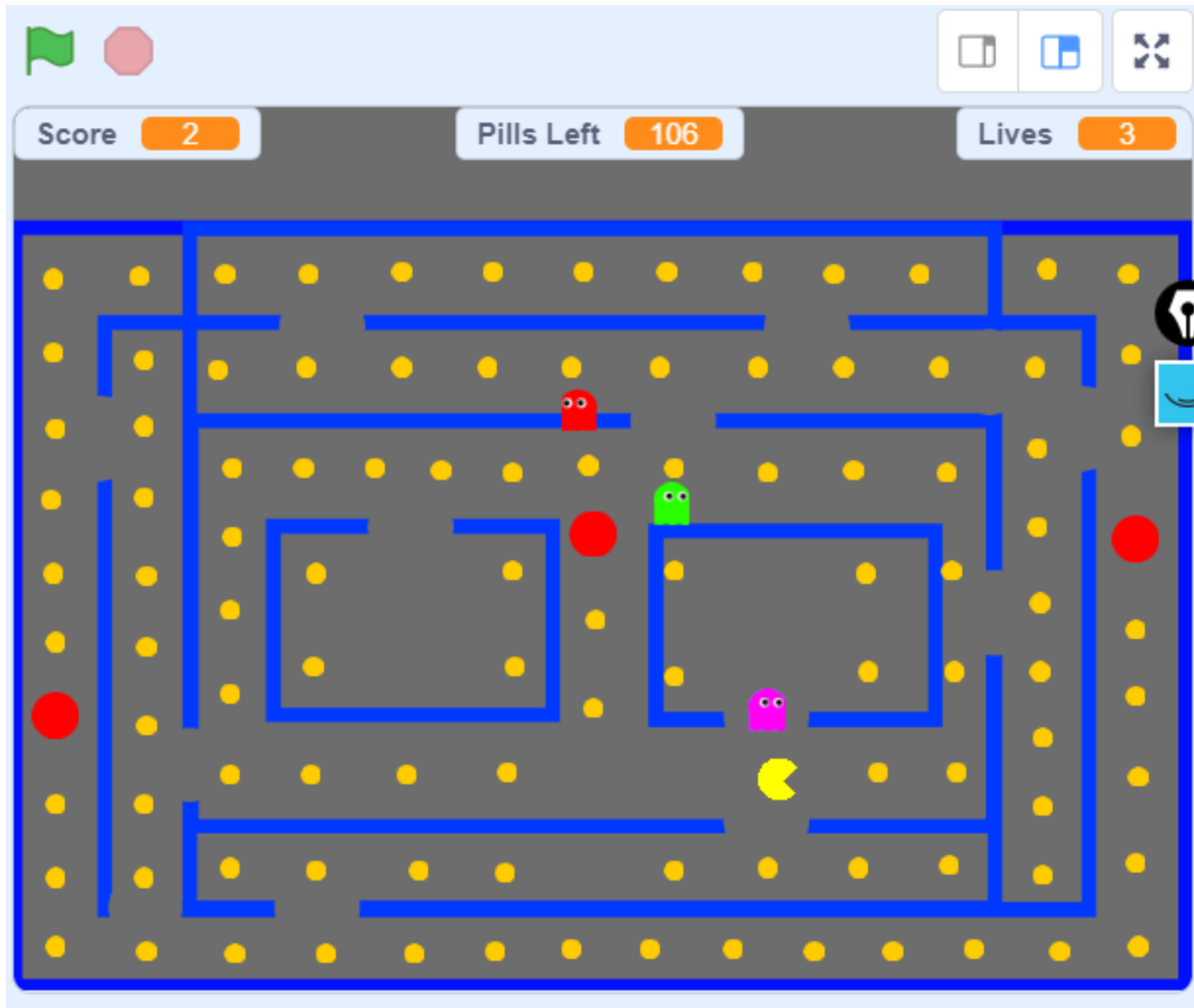
- **Alien 3** – This alien has two costumes – Purple color and the other blue color.



- **Alien 4** – This alien has two costumes – Cyan and blue



Note – You can change the position and size of the sprites as per your maze space. Pacman and aliens should be able to travel through the space available in the maze. Also, you need to hide the the game over, and you win, sprites in the beginning.



The final outcome of all the sprites and the background should look like the above image. Also, ensure that every sprite's initial position is fixed. Place all of them in the proper position as per your requirement.

Adding required variables

You will need to create 4 variables by going to the Variables block section of Scratch and clicking on the "Make a Variable" option. These variables are as listed below, along with the purpose for creating them.

- Blue – This variable is hidden and is used to control whether the aliens will be in the standard or blue color. This variable will initially have a false value as aliens will have their standard colors. When Pacman eats the red pill, the variable is set as true, and depending on the value of this variable, all the aliens turn their costumes to blue for 20 seconds. During this time, Pacman can eat blue aliens

and can get extra 100 points added to the score. After which, again blue variable is set to false, and aliens are back in their original colors where they can eat Pac-Man.

- **Lives** – This is a visible variable and is shown on the top of the stage. This variable is responsible for maintaining how many lives are available with the PacMan. In the beginning, its value is 3. As and when Pacman gets eaten by the alien, its life decreases by one each time. The game gets over when all of the lives finish, i.e., Lives = 0.
- **Pills Left** – It is a visible variable on the screen. This will be the total yellow pills remaining on the screen to be eaten by Pacman. Initially, it is set to 110 pills, as we have created 110 yellow pills. However, you can change it to as many yellow pills as you want to create. When Pac-Man eats any yellow pill, the pill left variable decreases by one. When the pill left variable reaches zero, Pac-Man wins the game.
- **Score** – This is also a visible variable. This is the number of points gained by us. For eating every yellow pill, you get 1 point. For eating a red pill, you get 25 points. While eating an alien when it's blue-colored gives you 100 points.