

GridWorld Role Play Information

Executive Summary

It has been suggested that this document has grown to sufficient length that many folks do not read the entire opus. This is a shame as it contains insights into how we designed the role play as well as specific suggestions for its use in the classroom. Collectively, it represents the wisdom of several dozen teachers. That said, the most important information is included in the paragraphs below.

Most obviously, the role play is a directed (as opposed to free-form) teaching exercise. As with any directed teaching exercise, it will be most effective if the teacher has specific goals in mind and tailors the experience to highlight those goals. Therefore, we suggest that teachers design the exercise to highlight those aspects of the case study about which they have the greatest concern.

Keep in mind that the role play as a whole is a metaphor for the case study. We have not included all of the details – indeed our early work on role plays indicated that removal of details was critical to the success of the exercise. If we removed a detail that you wish to emphasize, then put it back in.

Time spent in preparation for this exercise is time well spent. Most high school class periods have barely enough time to run through the exercise. If some of that time is spent assigning roles, setting up the room, discussing procedures, etc., then the exercise will almost certainly not run to completion and may not run sufficiently to encourage learning.

Similarly, the scripts and other devices, e.g. sheets for private data, are written in a particular style, both in terms of English and in terms of formatting on the page. There is nothing divinely inspired about this style (although it does work); rather we had to choose one and we have tried to be consistent throughout. If you teach private data in a different way; if you think we handled the commercial interruption poorly; if you feel that our use of language is confusing – THEN CHANGE IT. The hope is that this exercise is used to greatest effect, not that it is used in a liturgical fashion¹. BUT, before you change it, we encourage you to read the suggestions within, particularly the sections on the mechanics of the exercise and the advice about casting students into roles. These sections are particularly full of suggestions from other educators.

¹ We distribute the files in PDF format for a variety of reasons, but mostly to prevent multiple (and sometimes incorrect) versions from circulating; if your needs require you to edit the files, email us and we'll send you the original Word document files.

Introduction

The GridWorld role play is the most challenging that we have written to date. The challenge with any role play is to show the action of the program in such a way that concepts essential to the students are highlighted and concepts irrelevant to the students are as close to invisible as possible. The designers of the case study have done a remarkably good job of hiding code, especially the code related to the graphical user interface. Striving for a similar level of success was one of the most difficult challenges we face.

After examining the case study, we chose to create two role plays. The first is a rough approximation of a BugRunner as found in Part 2 of the case study. We imagine that this role play will be used to emphasize interobject communication, in particular how the various actors communicate with the grid. The second role play is a rough approximation of the CritterRunner found in Part 4 of the case study. We imagine that this role play will be used to emphasize how the Critter itself behaves.

The “scripts” for the roles are included in a companion document. The document you are reading contains a discussion of design decisions reached while creating the scripts and also includes some tips about how to use those scripts most effectively in the classroom. The most recent copies of both of these documents can be found at <http://www.cs.sbu.edu/dlevine/RolePlay/roleplay.html>.

Some of the advice contained in this document is self-contradictory. This is because different teachers who have used this exercise have different, and sometimes contradictory, ideas about how to make it work best. We have included the vast majority of the suggestions given to us. Read all of the advice and choose those points that seem most relevant to you.

Instructors are welcome, encouraged even, to use these scripts to help students and are requested to send feedback (positive, negative, suggestions for changes, etc.) to dlevine@cs.sbu.edu. There are no restrictions on the distribution of these documents except that authorship credit must be given where it is due.

How the Role Play Is a Compromise

GridWorld comprises many classes. Some are part of the Java runtime library and others are developed just for this case study. Some of the latter will be tested on the AP exam while others are “black box”; the functioning of the black box classes is of minimal concern to us. In particular, this role play concentrates on the various Actor subclasses – Bug, Flower, Rock and Critter, as well as the Grid interface. Although the exam will not test them, we have included scripts for ActorWorld and two “Runners”, i.e. classes with a main method.

The primary goal of these exercises is to expose students to the idea of intercommunicating objects in general, and to the classes of GridWorld in particular. The scripts were designed to be usable in a single class period (see notes below about timing.) We have attempted to ensure that interactions that occur during the role play accurately

reflect the intercommunication of objects within GridWorld, but in the interests of time, we have not included every interaction. The two most common ways that this philosophy manifests itself is that getters and setters do not appear in the role play nor do checks for error conditions. The role play strives to “tell the truth, but not the whole truth”.

Design Decisions

While writing this version of the role play, we made the following design decisions. We are interested in feedback regarding a) whether we decided correctly, and b) whether our decisions were implemented effectively in the scripts.

- **Getters and Setters would not be included as separate scripts; similarly issues of public vs. private would not be emphasized.** We felt that students at this level understand them and that the formality of using them in the role play distracted from other, more important points. Furthermore, Java would enforce the use of the methods whenever the students wrote code.
- **Error conditions would not be checked.** This decision was made for two reasons. The first is that an amazingly high percentage of the code is devoted to this kind of check and yet that code is not generally executed (meaning that the associated Exceptions are not thrown). Continually running this code distracts students from what we saw as the two main purposes of the role play exercises. The second reason for this decision is that it speeds up the process of the exercise.
- **Names have been changed to protect the grammatically innocent.** Some of the names of methods have been changed slight so as to make it more “natural” for English speakers to read their lines. This is particularly true with respect to pronouns, e.g. “**positionYourselfInGrid**” rather than “**putSelfInGrid**”. The order of parameters is sometimes swapped for similar reasons.
- **Inheritance as a mechanism would be de-emphasized in the role play.** It obviously exists by virtue of the fact that you have Bugs, Flowers, Rocks, and Critters, but the role play exercise does not call attention to the mechanism the way that it did in the JavaMBS exercises.
- **There is only one type of Grid.** BoundedGrid and AbstractGrid are folded into a single class. (This is consequence of the above decision.)
- **There is no role for the GUI.** The GUI is complex and event-driven. Accurately simulating it would have been nearly impossible and would certainly have detracted from the overall goals of the exercise – especially in view of the fact that students are not responsible for any of that action.

- **Some display of the data must be made for the audience support.** The result of this decision is that the Runner (main) will periodically stop for a “commercial interruption” to update the display. We have tried to make it very clear that this update is NOT part of the program.
- **The script for the Grid class greatly simplifies most of the methods.** We felt that the particular algorithms used were not relevant to the goals at hand. We have “inlined” some of the helper methods as well, again feeling that these details, while important on some levels, were not necessary to support the goals of the role play exercise.
- **Two Bug scripts are provided.** Our original Bug script emphasized the various methods (**canMove**, **move**, **turn**) within the Bug class. It has been suggested that the script for the Bug could be simplified by “inlining” these methods within the **act** method. The alternate Bug script, found at the end of the scripts, does so. The advantage of the alternate script is that it is simpler and thus may make the role play go more smoothly. The advantage of the original script is that it more accurately reflects how things happen. Normally these tradeoffs would cause us to prefer the alternate version, but the use of the internal methods is so important when students deal with Critters that we felt it should be introduced in the earlier role play. Although we have no firm data, we believe that the alternate script will result in the exercise completing in less time; this should be considered if time pressures are a concern.

Timing

“Past performance is no guarantee of future results.” Everyone who has done this has had a different group of people doing this under different circumstances. Depending upon your students, upon the mechanisms you use for sharing data as they act out roles, and (most importantly) upon the amount of external commentary during the activity itself, the exercise may take as little as forty-five minutes or as much as two hours. At one workshop with AP teachers, we were able to go through the exercise in about fifty minutes with a group of teachers who had not previously seen the scripts and were not very familiar with GridWorld.

(Side note: we are particularly interested in learning how long it takes others to go through the exercise. Send comments to dlevine@cs.sbu.edu.)

Instructions for the Cast

One of the areas where participants have the most difficulty is in deciding what to say. It is very important (see debriefing) that individual actors are addressed by their (personal) names and not their class names. We emphasize this during previous scripted role playing exercises, but anything that can be done in advance to “train” the actors will make the entire exercise go more smoothly. If the instructor has pre-cast the roles, the scripts can even be customized to help prevent this problem. (We have never done this ourselves, but have recently heard of others doing so with some success.)

In a similar vein, it is important that the actors speak all of the lines assigned to them AND that they not speak “stage instructions.” Reminding them to read their entire scripts before performing is important. On some occasions, a clarifying sentence follows an action. Participants who have not read the entire script tend to make the very mistake that the clarifying sentence is designed to prevent.

It may help to remind the cast that

- **words in boldface** represent requests that someone perform a task of that name
- underlined words represent stage directions
- *italicized phrases* enclosed in angle brackets represent ideas that should be applied, e.g. <your name>
- indentation matters; thus all sub-instructions are to be done provided the “if” condition is met – and none of them should be done if the condition is not met

Mechanics – Advance Prep

As part of the role play, the Runner will need to show the state of the world. There are various techniques that one can use for this, including the blackboard, an easel upon which various images are drawn before the page is flipped for the next display. One can even use a computer projector running a BugRunner to display the world for the first role play. (Randomness creeps in when you have Critters around, however.) If your set of colors is limited, you can draw a Bug, Rock, or Flower outline and write the word “red” inside to indicate a Red actor.

Before running through the role play, one needs to assign the roles to appropriate students – see below regarding casting.

Assigning the roles the day before the role-play activity and having the students read over their assigned class code helps move things along. Some teachers quiz the students on the scripts in advance; they can then assign the roles randomly (among those students who pass!)

Some people believe that the “Cast of Characters” should be publicly displayed (on the blackboard?) in an easy-to-read format, e.g.

- Grid – Happy
- Rock – Doc
- ActorWorld – Sleepy
- Runner – Grumpy
- first Flower – Snezy
- second Flower – Bashful
- third Flower – Dopey

If the students don't know each other well or if visibility is a problem, it can also be helpful to preprint the "Cast of Characters" and distribute these to all students. (Some people have found this list to be helpful even after the completion of the role playing exercise.) Nametags with LARGE labels, e.g. "Sleepy/Simulation" can help. Some people have used large sections of poster board for this purpose.

It will also be necessary to reserve a special part of the blackboard (or projection device) for the Runner.

Be sure to have appropriate numbers of copies of each script available as well as a substantial number of "List" sheets.

Many instructors assign roles and have the participants study their scripts in advance (e.g. the night before the exercise). In some situation, e.g., workshops, this may not be feasible. Regardless each student should have read his/her entire script before the exercise begins and should realize that s/he will perform only one task at a time.

In a time-tight situation, the facilitator should intervene as little as possible, generally doing so only to correct errors or to minimize duplicated work.

It will make the facilitator's life a lot easier if s/he has a copy of each script! This can also be a task assigned to a student or three!

A Note on Method Invocation

As noted above, one of the more difficult tasks for some students is figuring out how to make a request of another object. This must be clear to all participants before the role play begins. One way to accomplish this is to have run through a simpler role playing exercise before beginning this one. (See <http://www.cs.sbu.edu/dlevine/RolePlay/roleplay.html> for an example of a simple exercise that can be done as early as the first day of class that accomplishes this – and a good bit more.) If your students have no such experience, it is worthwhile to take some time to "teach" them your message passing protocol. Keep it simple. For example, consider the cast above; if the Runner wants the ActorWorld to **step**, then an appropriate statement for Grumpy to make would be, "Sleepy, please STEP." When a method is complete, the actor will either state a result (for a non-void method) or make a statement that obviously terminates the work (for a void method.) Note that the caller's name is not used as part of this statement; the caller is usually anonymous – though known to everyone in the room – and presumed to be paying attention.

A Note on Private Data

In an ideal world, only the actor playing the role of a given object would have access to that object's private data. While this would be the most faithful representation of the code, it may not be the best pedagogical use of the role play. Non-participants (or even not-so-active participants) often learn more if they can watch and monitor such bookkeeping. Some of the various options for dealing with private data include:

- Keeping the data totally private (the “faithful” solution). Each actor writes the data on a sheet of paper in front of her/himself – or even keeps it in her/his head.
- If one is using large (8½ by 11) name tags, the actor can write the data on the back of the sheet. It can be private, but the audience (or the facilitator) can ask to see it at any point in time. In this way, the privacy is emphasized, but the engagement can be maintained.
- The actor, a confederate/understudy, or the facilitator can draw the data on the blackboard, flip charts, or in some other public forum. This does not emphasize the privacy as well, but physical separation and facilitator “guidance” can reinforce this.
- Depending upon the learning outcomes desired, it may be beneficial to have the Grid show its data in a public manner. If this is done, care must be taken to ensure that the Runner does NOT ever directly view/use the Grid private data. In other situations, it may be better to have audience members maintain their own copies. See the section on the Grid's data below for a more complete discussion of mechanisms for this, along with some deeper discussion about the costs and benefits.

Mechanics – As the Action Unfolds

There are several points to consider/emphasize regarding the “acting on stage”:

- Objects must be addressed by their names, not by their roles, e.g. “Chris, please getNeighbors using (3,4)”, not “Grid, please getNeighbors using (3,4)” or “Grid Chris, please getNeighbors using (3,4)”. [Note that Chris is the actor's name, i.e., the name of the object. Grid is the role, i.e., the name of the class.] Although the protocol for this should have been worked out in advance, this is one of the most common errors made by participants and the facilitator needs to show constant vigilance.
- The Grid should be storing the names of the actors who are in each cell. It should not store any of their private data.
- Some people like to indicate flow of control through the use of an artifact, e.g., a nerf ball held by the object currently executing, or a jester's cap, etc. If this is done, then one actor should pass the artifact to another as “control” is passed via a message. Note that furniture configurations can play a role in the

success of this idea! [On the negative side, some people don't care for this as it isn't always clear where the artifact should go at the completion of a task.]

- Some people like to maintain a call stack of some sort. One way to do this is to have the actor playing the main method start holding the end of a strand of a ball of yarn. Whenever a message is passed, the source object grabs the near end of the ball, and then passes the ball to the object being called, unrolling the ball along the way. When a task completes, the ball is returned (and rewound) as it is passed back to the caller. Note that if an object appears several times on the call stack, then its actor will need to hold several points along the string of yarn.
- Since the role is so “active”, some people like to have the Grid in front of the group
- It is helpful to distinguish between constructed and not-yet-constructed objects. Depending upon the physical environment this can be accomplished by requiring not-yet-constructed objects to stay “off-stage”, either by staying against a wall or in their seats.
- More students can be engaged if the idea of understudies is used. Understudies share private data with their main actor counterparts. To increase student engagement, understudies can be told that they are to take over the roles during a commercial interruption.
- (Unauthorized) shortcutting is NOT allowed. After a bit, some actors/objects will try to bypass intermediaries “for efficiency.” This should not be allowed to happen. There is a great “teaching opportunity” here regarding efficiency of the code.
- Student actors will make errors. It is important to correct them politely, but firmly when this does happen.

Mechanics – The Runner and the Commercial Interruption

Writing a role play for an event driven program is difficult, although probably not impossible. At the very least, it would require actors whose sophistication level is beyond the scope of APCS. Given these constraints, as well as a desire to focus on the classes that are formally part of the curriculum, we have created the concept of a Runner. Within the role play, the Runner is meant to serve a similar role to that of a main class, but we have attempted to have it perform in a slightly unusual manner so that it will be more obvious that this is not a primary part of the case study.

Thus was born the “Commercial Interruption”. The explicit announcement of this fact, along with the bad pun at the end, are meant to highlight the fact that what is occurring between those two statements is NOT part of the program, but is instead a

compromise that has been explicitly included (for the benefit of the audience). The instructions for the commercial interruption are thus more informal and may be adjusted as the circumstances merit. What is important, however, is that the students understand that the Runner is choosing to show them a view of the data and that while this view reflects the status of the program, it is not being produced explicitly by any of the classes.

Another reason for the informality of the commercial interruption is that the existing classes do not support the methods needed to draw the display. To do so effectively, the Runner would need to get a list of Actors from the ActorWorld, but the ActorWorld supports no such method. We decided that adding such a method to the ActorWorld would create much more confusion and would still not bring the role play any closer to the actual event-driven process that takes place when the GUI is used.

Mechanics – Ending

Currently, the Runners move through three steps. This is arbitrary. It seems like the right number to us, but you can easily do more (or fewer) steps.

Debriefing

During the debriefing, it should be emphasized that the role play is a simplification. Some of the simplifications of the role play can lead to student misunderstandings later. These can be avoided if you are alert to them. Things to discuss during the debriefing:

- Importance of name of object vs. name of class – particularly for the various Actors (i.e. we have only one Flower class, but many Flower objects.)
- Getters and setters were removed to simplify the exercise.
- Behaviors of some methods have been considerably simplified in certain cases, notably to remove various kinds of error checking.
- The script for the Grid class greatly simplifies all of the relevant methods.
- The role play Grids are 10x10 and bounded; this is not generally true in GridWorld although it is the default.
- The “commercial interruptions” are part of the role play, but not part of the program. They are a compromise because the GUI is not to be studied.
- The order in which the various actors act is arbitrary. It is determined by the order of the actors names in the list given to the ActorWorld by the Grid, but neither the specification nor the role play scripts specifies a particular order. (The code, of course, does induce a particular order, but that is unavoidable.)

- Some people are concerned about the statement “Null has been removed from the grid.” While grammatically dubious, the statement reflects the fact that the remove method of the grid class does return a value; that value is null if there was no Actor to be removed.

Note: If each student has a copy of the “Cast of Characters”, the discussion often goes more smoothly during the debriefing.

Casting

Who is chosen to play which role can have a major effect on the time needed to complete the exercise. Some tips are below:

- Runner – This role is fairly easy, but is the weirdest of all of the roles due to the nature of the “GUI”.
- ActorWorld – A fairly easy role.
- Rock – Ditto.
- Flower – Only a bit tougher than Rock
- Bug – The hardest role in the first role play; it requires a sharp student who pays attention to detail.
- Critter – Moving “up the evolutionary ladder”, a bit tougher still. Requires a student to have patience and an ability to follow the script slavishly.
- Grid – The “starring” role; it is best played by a student who can think on his/her feet (either with or without preparation as circumstances dictate).

It may be best to cast by trying to find bright students to play the roles of Grid, the Bug and any Critters. Don’t worry too much about the rest of the casting. If the class size is small, the facilitator can assume the roles of the random number generator and the main program (in that order).

Kinesthetic Learning

The role play exercise is (or at least should be) a kinesthetic learning experience. Having the students get up, move around, and interact with each other instead of sitting listening to a lecture presents a different learning modality. Experience has shown that the more the students are “out of their chairs” the more memorable the experience is for the students. In particular, it is most effective to have the participants in the front of the room, like actors on a stage.

There are several things that one can do to make the exercise more kinesthetic. Some of these relate to using artifacts to indicate the flow of control and are described above under the heading of “Mechanics”.

One nice thing that one can do is to make use of costumes. While “full body costumes” are most amusing, one can get significant “bang for the buck” using much

simpler devices including hats or simply large (8.5 x 11 inch) signs hung around participants' necks. [Simply print the sign on a sheet of paper, drop it in a clear plastic protector that would be used with a 3-ring binder, and tie string to the outer holes, leaving enough to slide easily over a person's head. Such signs might (or might not) illustrate the private data for an object.] Simple, yet comical, costumes are often the best.

One great way to engage more students is to assign an understudy to each actor – or at least to the more complex actors. The understudy is responsible for making sure that the actor behaves properly and must be ready to take over at any time. Declaring a “major cast illness” that requires all understudies to take over can be very effective. Requiring the actor and the understudy to share a copy of the script has been recommended by some folks, but has been thought to be cumbersome by others.

Lastly, one can create the role of critic. Critics watch the role play unfold and then tell everyone what is wrong with it. This includes both obvious flubs and areas in which the role play is not behaving according to the code. Critics should give their reports before the debriefing. Note that the inclusion of critics implies that students are already familiar with the code; many people use the role play as an introduction to the case study and thus cannot make good use of critics.

Variations

Depending upon when and how the role play is used, it may be useful to change the initial world. It might be desired to have a Bug run into a Rock or a Flower or to attempt move off the edge of a BoundedGrid. Alternatively, various Critter-Critter interactions might be demonstrated. In either case the Runner scripts can be easily modified to ensure that the desired actions take place.

Similarly, new scripts could be written for creatures such as BoxBugs with the Runners again modified. It is not clear to us that the role play is the best vehicle for studying new classes such as BoxBug, but if that is the goal, then the modifications are easy enough to make.

The Grid, the Display, and Some Debate

While the Actors store small amounts of private, the Grid stores an entire array of references to Actors. This makes it very tempting to “build” this composite structure out of artifacts in the classroom. It has been suggested that the grid could be marked out on the floor in tape and that the various Actors could be “placed” in the appropriate cells by the student playing the role of the Grid. Alternatively, the desks (if arranged in conventional row-column manner) could be used as Grid cells and Actors could be seated at particular desks. In these cases, it can be advantageous to put signs on the various front, side, and back walls reading, “North”, “South”, “East”, and “West”. Various other physical manifestations of the Grid are possible and limited only by the imagination of the producer (i.e. the teacher) and the room itself. The question is, “Is this wise?” Arguments in both directions are given below.

Pro:

Argument: As noted in the previous section, the role play is a kinesthetic experience. Creating a physical grid only enhances this for the participants. Also, by creating a physical grid, one enables non- or minimally-active participants to become more engaged. This also helps minimize the chance for any placement errors. Overall, everything is more visual and thus the exercise supports another type of learning. Besides, it's fun.

Rebuttal: The role play exercise is already fun and there are other equally good mechanisms for achieving these benefits without incurring the costs (described below). Understudies can be used to help minimize errors and the physical manifestation aspect can be used during the display phase.

Con:

Argument: The Grid (as it exists in the case study) is NOT visual nor does it store Actors. Rather it store references to Actors. The physical manifestations of the Grid described above can create confusion in students' minds about the distinction between an object and the reference to that object – a distinction that is already shaky in some cases. In addition, showing this off violates the idea of private data AND gives the impression that the Grid is intrinsically a matrix whereas that is not true the UnboundedGrid implementations discussed in the case study. Finally, these implementations take away the “power” of the display component and blur the Model-View-Controller aspect of the case study.

Rebuttal: Students have no problem with the reference/object distinction by the time they hit this portion of the course. On the other hand, by sharing the Grid's data, we can reinforce concepts such as array indexing. Besides, public/private isn't part of the role play. Furthermore, the Grid IS a matrix, in an abstract sense even if some implementations don't store it this way. Having a physical manifestation opens up this entire aspect of the case study for discussion during the debriefing. And, MVC may be part of the program we run, but it is not part of the case study, so blurring distinctions there is cost-free.

Conclusion: There are good arguments on both sides of this issue. Personally, we feel that the “Cons” outweigh the “Pros” and would not take this approach. We are certain, however, that anyone who considers the issue carefully before choosing is probably going to do a better job with the exercise.

As for the display, it is NOT part of the case study. On the other hand, if one can't see the data, much of the value is lost. Therefore it is important to give careful consideration to how the data is to be presented. Blackboards, whiteboards, easels with separate sheets, poster boards, “sticky sheets” made of plastic and/or felt are all obvious choices for media on which to create a display.

Alternatively one can use any of the physical manifestations described above. In this case, it should be made clear that the runner (an outside agent) is directing the action, not the Grid. In fact, having students (not the objects) stand in places can be an excellent way of getting extra students involved. The difference between objects that “appear” and

don't appear could be made more dramatic by using costumes ONLY for the former. (In other words, you get a script OR a costume, but not both.)

Finally, one might use the computer and the room's projector to show the display. It is trivial to set up a GridWorld that mimics the role play and then run it one step at a time. The only real drawback to this is that the connection between what is being done in the role play and what is being done inside the Java virtual machine must be taken on faith. This may have a "watching magic be done on non-live television" feel to it.

Acknowledgements

All of our previous role playing exercises were originally developed by us and then modified significantly based upon the feedback from many other folks. Many of the best ideas have come from folks who took the time to share their insights with us and all of the other people who download these documents. We look forward to more of the same with this project.

More specifically, thanks are due to Michael Lew (Loyola High School), Robert Duvall (Duke University), Maria Litvin (Phillips Academy), and Kathy Larson (various affiliations) for their contributions to the scripts and this document.

A Final Note

We hope that you find these exercises useful. All we ask is that if you use them, you give us feedback.

These materials represent a best-faith effort of the authors – both named and unnamed. When, not if, you find errors of omission (or of commission) within either of them, please send corrections, comments, etc. to dlevine@cs.sbu.edu.

We have had great success with this exercise as have more than a few high school teachers. We hope that you do, too.

--david levine and steve andrianoff, st. bonaventure university, Fall, 2007

***P.S. Links to all of the role play documents can be found
at***

<http://www.cs.sbu.edu/dlevine/RolePlay/roleplay.html>