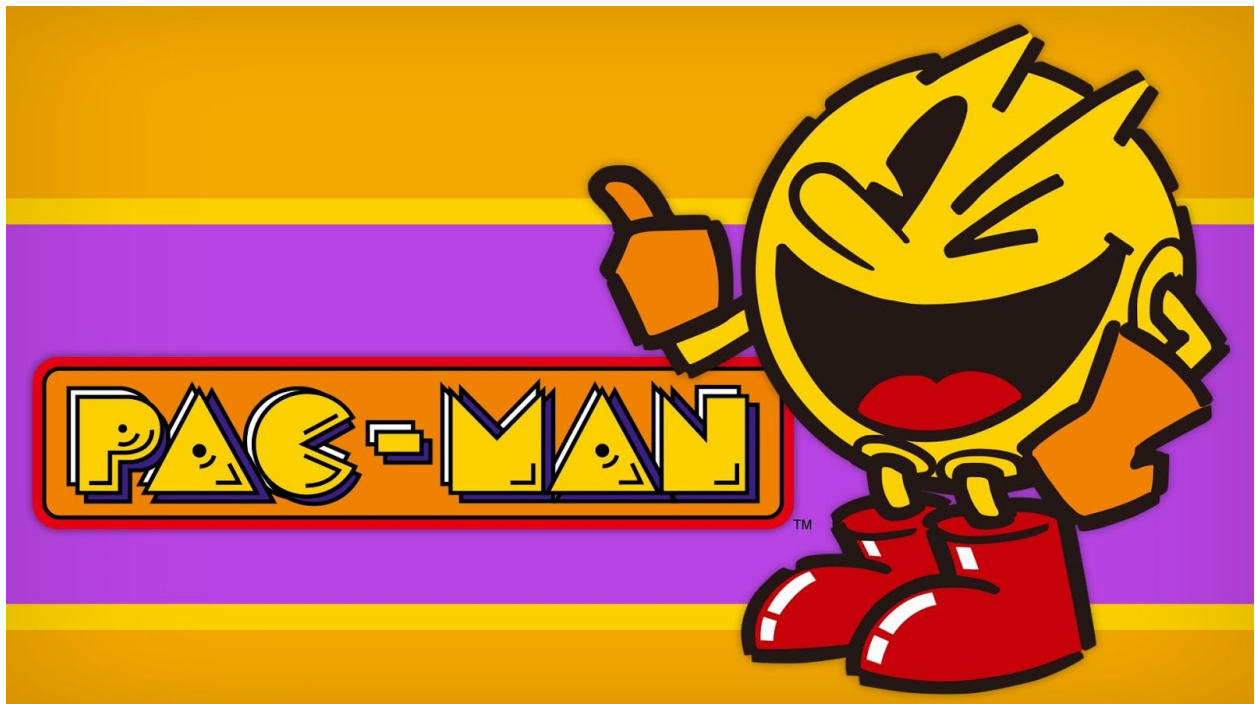


پروژه دوم



کلیت پروژه

در این پروژه شما باید بازی ای مشابه Pacman طراحی کنید. توصیه می شود که اگر Pacman را بازی نکرده اید اول بازی کنید سپس ادامه این pdf را بخوانید.

اهداف

1. یادگیری Javafx
2. آشنایی با چندریختی و ارث بری
3. کار با file
4. آشنایی با Event-Driven Programming
5. آشنایی با الگوریتم های اولیه مسیریابی (BFS)

توضیحات

مشابه بازی Pacman اصلی بازی پنج نوع روح و یک Pacman است.

انواع روح ها :

1. روحی که به صورت رندوم فقط افقی حرکت میکند. (چپ یا راست)
2. روحی که به صورت رندوم فقط عمودی حرکت میکند. (بالا یا پایین)
3. روحی که در هر چهار جهت به صورت رندوم حرکت میکند.
4. روحی که در هر چهار جهت حریصانه حرکت میکند. (نکته 1)
5. روحی که در هر چهار جهت هوشمندانه حرکت میکند. (نکته 2) (امتیازی 1)

حالت های مختلف بازی :

1. یک نفره : مشابه بازی Pacman عادی
2. دو نفره با کامپیوتر : مشابه حالت یک نفره ولی یک Pacman دیگری به شما کمک میکند برای خوردن سکه (نکته 3)
3. دو نفره : در این حالت دو نفر در نقش یار یکدیگر با استفاده از دو سری کلید کیبورد مختلف دو Pacman را هدایت میکنند.
4. (امتیازی 2) دو نفره رقابتی : در این حالت اولین کسی که به روح برخورد کند می بازد (اگر هر دو نفر همه سکه هارو خوردند آن کسی که سکه بیشتری خورده بود برنده است.)

نکات

1. روحی که حریصانه حرکت میکند تمام نقاطی که می تواند به با یک حرکت به آنجا برود را در نظر میگیرد و آن نقطه ای که به هدف نزدیک تر باشد را انتخاب میکند.
2. در این نوع روح کوتاه ترین مسیر ممکن به هدف را پیدا میکند و اولین قدم به سمت هدف را از آن مسیر انتخاب میکند. (میتواند از BFS یا Dijkstra استفاده کنید)
3. حرکت این Pacman به صورت رندوم است و با خوردن به روح از بین میرود. (در صورتی که هوشمند حرکت کند (امتیازی 3))
4. در چهار گوشه نقشه چهار سکه بزرگ وجود دارد که با خوردن آنها روح ها به مدت 5 سیکل بازی متوقف میشود. (هر سیکل بازی یک بار حرکت کردن تمام روح هاست)
5. بازی دارای یک نقشه پیش فرض است که از روی فایل خوانده می شود. (امتیازی 4 داشتن نقشه اضافی)
6. هر Pacman سه جان دارد غیر از Pacman ای که در نقش یار در حالت دوم است.
7. محل قرار گیری اولیه Pacman و روح ها بسته به انتخاب شما میتواند ثابت یا رندوم باشد.
8. در پیاده سازی باید از شی گرایی بهره ببرید (عدم استفاده از اصول برنامه نویسی شی گرا از جمله ارث بری و چند ریختی و ترکیب موجب کسر نمره می شود)
9. سعی کنید مساله را به بخش های کوچکتر تقسیم کنید و روی کاغذ یادداشت کنید سپس کد آن را پیاده سازی کنید. (اگر پروژه برای شما سخت است از امکانات سخت آن کم کنید.)

10. طراحی شی گرا پروژه نسبت به اجرای آن مهم تر است اما عدم اجرای موجب کسر حداقل 10 درصد نمره میشود.
11. پروژه را زودتر شروع کنید تا وقت کم نیاورید، در صورتی که بخش قابل توجهی از پروژه را زودتر (حداقل یک هفته) پیاده سازی کنید می توانید پیش تحویل دهید و از مشکلات احتمالی زودتر با خبر شده و تا موعد تحویل مشکلات برنامه خود را رفع کنید.
12. قابلیت های امتیازی بسیاری در سطوح مختلفی در نظر گرفته شده ، بخش های ساده را تشخیص داده سعی کنید پیاده سازی کنید تا نمره بیشتری بگیرید.
13. سعی کنید کار کردن با گیت را یاد بگیرید چرا که جز امتیازی های ساده محسوب می شود برای پروژه نهایی احتمالا اجباری بوده و حدود 10 درصد نمره اصلی را در بر خواهد گرفت.
14. در صورتی که پروژه قبلی را انجام نداده اید و یا نمره کامل نگرفته اید حتما وقت کافی روی این پروژه بگذارید تا علاوه بر یادگیری نمره نگرفته پروژه قبلی جبران شود.

پیشنهادهات

1. در این پروژه می توانید از کتابخانه JGraphT استفاده کنید که دارای انواع مختلف گراف و درخت و نیز الگوریتم های مسیریابی است.
2. در صورت برخورد با ارور not on javafx thread از platform.runLater استفاده کنید.
3. برای پیاده سازی هر سیکل بازی میتوانید از کلاس های Timer و TimerTask استفاده کنید.
4. استفاده از AnimationTimer مجاز و استفاده از fxmml غیر مجاز است.
5. استفاده از الگوهای طراحی مختلف پیاده سازی را برای شما آسان تر میکند از جمله observer
6. استفاده از الگو های معماری مثل mvc پیاده سازی را برای شما روشن تر و ساده تر میکند.
7. سعی کنید قبل از شروع پیاده سازی پروژه را با استفاده از class diagram مدل کنید تا سرعت پیاده سازی شما افزایش یابد.

توضیحات گیم پلی

روند حرکت روح ها ترتیبی است به این معنی که نیاز نیست همه روح ها با هم حرکت کنند و نوبتی حرکت میکنند.

قابلیت های امتیازی

1. امتیازی 1 (در بالا اشاره شده است) 80 امتیاز
2. امتیاز 2 80 امتیاز
3. امتیازی 3 60 امتیاز
4. امتیازی 4 40 امتیاز
5. قابلیت ذخیره بازی 50 امتیاز
6. نوشتن تست واحد 50 امتیاز (با استفاده از junit)
7. استفاده از گیت 60 امتیاز
8. قابلیت replay بازی 100 امتیاز
9. استفاده از تصاویر مناسب برای نمایش اجزای برنامه 50 امتیاز
10. ذخیره امتیازات در فایل 40 امتیاز (اطلاعات این فایل در داخل بازی نمایش داده نمی شود)

11. به وجود آمدن جان اضافی در نقشه برای مدت محدود 50 امتیاز
12. داشتن صدا در بازی 50 امتیاز
13. امکان بازی دو نفره تحت شبکه با استفاده از سوکت 90 امتیاز
14. قابلیت عوض کردن رنگ Pacman و پس زمینه 40 امتیاز
15. استفاده از serialization در فرآیند ذخیره سازی 40 امتیاز
16. راحت بودن کار کردن با برنامه 20 امتیاز
17. استفاده از الگوریتم های heuristic دار جهت هوشمند سازی برنامه 30 امتیاز (A-star)
18. داشتن روح با قابلیت جدید که در بالا ذکر نشده 15 امتیاز (مثلا میتواند از روی دیوار ها بپرد)
19. داشتن سطح بندی در بازی 45 امتیاز (مثلا در سطح سخت روح با حرکت رندوم نداریم)
20. داشتن صفحه game over و امکان از نو بازی کردن 10 امتیاز

لینک های مفید:

پیاده سازی الگوریتم bfs به همراه اسلاید های آموزش :

<https://github.com/shayandaneshvar/Path-Finder-using-BFS>

کتابخانه JGraphT:

<https://jgrapht.org/>

نمونه ساده استفاده از MVC و Observer:

<https://github.com/shayandaneshvar/2048>

نمونه استفاده از JGraphT و Observer و MVC و ...:

<https://github.com/shayandaneshvar/Quoridors>

لینک ساختار اولیه پروژه:

.....